

Annales corrigées et commentées

Concours

2021 / 2022 / 2023

MP MP*

Maths et informatique

3^e édition

**Concours commun Mines-Ponts
Centrale-Supélec
CCINP**



Abdellah Bechata
Christophe Devulder

Maths et informatique

Annales

corrigées et commentées

Concours

2021 / 2022 / 2023

MP-MP*

3^e édition

Maths et informatique

Concours commun Mines-Ponts
Centrale-Supélec
CCINP

Abdellah Bechata
Professeur de chaire supérieure

Christophe Devulder
Professeur de chaire supérieure



Collection
Annales
corrigées et commentées

Retrouvez tous les livres de la collection sur www.editions-ellipses.fr



ISBN 9782340-085589
©Ellipses Édition Marketing S.A., 2024
8/10 rue la Quintinie 75015 Paris



Le Code de la propriété intellectuelle n'autorisant, aux termes de l'article L. 122-5.2° et 3°a), d'une part, que les « copies ou reproductions strictement réservées à l'usage privé du copiste et non destinées à une utilisation collective », et d'autre part, que les analyses et les courtes citations dans un but d'exemple et d'illustration, « toute représentation ou reproduction intégrale ou partielle faite sans le consentement de l'auteur ou de ses ayants droit ou ayants cause est illicite » (art. L. 122-4).

Cette représentation ou reproduction, par quelque procédé que ce soit constituerait une contrefaçon sanctionnée par les articles L. 335-2 et suivants du Code de la propriété intellectuelle.

www.editions-ellipses.fr

À Clément-Raphaël auquel je pense beaucoup.

Avant-propos

Cet ouvrage propose l'intégralité des annales en mathématiques, informatique commune (IPT) et option informatique des concours passés par le plus grand nombre d'étudiants de CPGE MP (Mines, Centrale, CCINP) entre 2021 et 2023. Afin que la préparation aux concours soit optimale, ce livre propose :

- le sujet original proposé aux candidats du concours pour chaque épreuve, chaque concours et chaque année ;
- les diverses thématiques abordées par ce sujet ;
- un corrigé détaillé ;
- des commentaires associés aux différentes questions : nature de la question (ouverte, fermée, cours, etc), niveau de difficulté (élémentaire, difficile, etc), compétences attendues (autonomie, initiative, technique, etc), réussite des candidats ayant réellement passé cette épreuve en se basant sur les rapports de jury.

Conseils pour rationaliser et optimiser la préparation de l'écrit

La préparation de l'écrit commence ... dès le début de l'année et se prépare régulièrement (d'où le terme CPGE : classe PRÉPRATOIRE aux grandes écoles).

PREMIÈRE ÉTAPE

Il faut construire un système de fiches pour chaque chapitre (de première et seconde année). Chaque fiche doit être courte (2 à 4 pages maximum, si possible sous forme de fiches bristol ou d'un cahier de primaire i.e. de petit format) et contenir les éléments les plus importants du chapitre (définitions et théorèmes importants avec leurs hypothèses et conclusions bien identifiées). Il est important de comprendre que :

- les définitions sont importantes car il s'agit des outils avec lesquels on veut travailler (par exemple, injectivité, série convergente, algorithme de tri ou notion de pile (informatique commune), arbre binaire de recherche (option informatique) etc.) ;
- les théorèmes (ou propositions ou lemmes) sont importants car ils sont les méthodes qui permettent d'utiliser les objets primitifs formant les définitions et de les manipuler effectivement. Par exemple, si f est linéaire alors f est injective si et seulement si $\ker(f) = \{0\}$ ou bien un théorème de comparaison : si $\forall n \in \mathbb{N}, u_n \geq 0$ et $u_n \underset{n \rightarrow +\infty}{\sim} v_n$ et la série $\sum_n v_n$

converge alors la série $\sum_n u_n$ converge ou la complexité des différents tris (informatique commune) ou l'automate reconnaissant l'intersection de deux langages reconnus par des automates (option informatique) ;

- dans les raisonnements complexes (typiquement ceux des concours), il y a une inter-action permanente entre les définitions et les théorèmes. Par exemple, pour montrer qu'une suite $(u_n)_n$ est convergente, on utilise le lien suite-série : $(u_n)_n$ converge si et seulement si la série télescopique $\sum_n (u_{n+1} - u_n)$ converge. Pour étudier la convergence de la série

$\sum_n (u_{n+1} - u_n)$, on étudie par exemple son absolue convergence (théorème : convergence absolue entraîne convergence), on utilise les théorèmes de comparaison (cf. item précédent), on conclut à la convergence absolue donc à la convergence de la série $\sum_n (u_{n+1} - u_n)$

donc à la convergence de la suite $(u_n)_n$. Pour écrire un algorithme de tri (informatique commune ou option informatique) sur des données qui ne sont plus des nombres (par exemple, des chaînes de caractères ou des couples (pays, nombre d'infectés)) mais des entités possédant un ordre, on modifie un algorithme de tri du cours afin de tenir compte du nouvel ordre. L'analyse de sa complexité découle également d'une modification de la complexité du tri analogue sur les nombres ;

Ce travail est important : outre qu'il permet d'accéder rapidement et simplement aux éléments clés de chaque chapitre (ce qui sera un élément déterminant lors de la préparation des écrits), il vous force à effectuer une synthèse régulière des connaissances qui vous ont été enseignées et donc à cristalliser vos connaissances sur un thème précis. Si vous avez du mal à en produire, discutez en avec vos camarades et/ou vos professeurs.

DEUXIÈME ÉTAPE

Il faut apprendre régulièrement son cours, précisément ses définitions et théorèmes fondamentaux en premier lieu (avec précision en distinguant les hypothèses des conclusions et en n'oubliant pas les quantificateurs). Puis, si possible apprendre les preuves et/ou les programmes associés. Dans le doute, il s'agit probablement des questions de cours pour les colles de maths et pour l'informatique, il s'agit des preuves et programmes inclus dans le cours (pas ceux des séances de travaux dirigés ou de travaux pratiques ou de devoirs) et inscrits au programme officiel (par exemple, les algorithmes de tri, leurs complexités, leurs corrections ainsi que leurs programmations).

L'intégralité des rapports de concours (CCINP, Centrale-SupElec, Mines-Ponts, X-ENS) indique depuis de nombreuses années l'importance de la connaissance du cours (que cela soit en mathématique, informatique commune (IPT) ou option informatique) et de la maîtrise de ces points fondamentaux. Il s'avère dans la pratique qu'il s'agit du premier critère de discrimination des candidats aux concours (tant aux écrits qu'aux oraux). Si vous ne savez pas ce qu'est un point fondamental, soyez pragmatique : si ce point est utilisé régulièrement au cours de l'année (cours, exercice, devoir libre, devoir surveillé), c'est qu'il est important. Sinon, discutez en avec des camarades ou vos professeurs.

Cette connaissance précise du cours vous octroiera une agilité certaine lors des écrits et vous permettra de vous adapter bien plus facilement à des situations variées et nouvelles.

TROISIÈME ÉTAPE

Lorsque l'on travaille un sujet d'écrit, on commence par disposer de l'intégralité de ces fiches de révisions ou bien de son cours ou bien d'une référence (un manuel par exemple). Au début, je vous déconseille de travailler 3 ou 4h sur un sujet. Il est préférable de fractionner le travail en paquet d'une ou deux heures (pas plus) correspondant à une ou deux parties (pas plus) ou à une sélection pré-déterminée de questions (10 au plus). Pensez à lire toutes les questions, à indiquer les questions qui vous semblent accessibles afin d'y réfléchir en priorité (mettre son énergie en priorité sur ce qui est accessible, au bout de 2h, vous aurez moins d'énergie et vous serez moins lucide).

Pour chaque question que vous traitez, commencez par vous demander à quel(s) chapitre(s) ou notion(s) se rattache(nt) la question. Si vous connaissez la démarche à suivre pour y répondre (définition, théorème, méthode, algorithme ou programme), suivez là. Si vous avez des doutes, n'hésitez pas un instant à vous référer à vos fiches (ou cours ou manuel) pour récupérer les informations adéquates puis répondre à la question. En tout cas, si à l'issue d'au plus 10 mn, vous n'avez pas démarré la résolution de la question ou si vous êtes sur la même question depuis 20 mn, changer impérativement de question (le temps fait partie de l'équation aux concours). Vous y reviendrez ultérieurement s'il vous reste du temps.

Pensez à rédiger comme aux concours avec précision, clarté, rigueur et en encadrant ou soulignant la réponse (vous vous entraînez pour cela et ceci ne s'apprend pas le jour du concours). Si vous êtes trompé, rayez par un seul trait oblique l'intégralité de votre erreur et écrivez en dessous (surtout ne pas effacer et écrire dessus, cela fait toujours des copies très sales). L'objectif est de produire une réponse de qualité (clarté, propreté, précision) qui sera un élément clé de l'évaluation par les correcteurs.

À l'issue d'une ou de quelques questions traitées (ou partiellement), n'hésitez pas à lire le corrigé et surtout à vérifier que vous avez convenablement utilisé les définitions, théorèmes, méthodes ou programmes de votre cours (des points clés oubliés ou mal justifiés, des erreurs grossières de programmation, etc.). La priorité n'est pas de faire beaucoup de questions mais que chaque réponse soit juste avec un argumentaire convenable ou, si ce n'est pas le cas, que vous réfléchissiez sur vos erreurs et/ou imprécisions et/ou oublis. Pour chacune d'elle, pointez le (ou les) chapitre(s) concerné(s) et pensez à revoir en profondeur les définitions, théorèmes, méthodes, programmes correspondants non pas le jour même mais le lendemain ou dans les jours suivants afin de permettre à votre cerveau de se reposer un peu, d'y réfléchir et de revoir cela avec un esprit frais (indépendamment de la résolution de votre problème).

L'objectif de réviser de façon fractionnée n'est pas de répondre à des questions particulières d'un sujet particulier mais de remettre à flot les connaissances, méthodes et compétences accumulées au cours des deux années de CPGE, de les consolider. Cela permet également de les faire inter-agir afin de gagner en agilité, autonomie et initiative, condition indispensable pour être en mesure de répondre convenablement à tout nouveau questionnement issu de n'importe quel sujet de concours. Ceci vous sera infiniment plus utile que de répondre à l'intégralité d'un sujet (ce dont sont capables bien peu de candidats, au mieux une dizaine sur des milliers de candidats).

QUATRIÈME ÉTAPE

La priorité pour préparer l'écrit n'est pas de faire une quantité impressionnante de sujets, mais de disposer d'une large couverture du programme de première et seconde année. La qualité et la variété sont à privilégier incontestablement sur la quantité afin de gagner en agilité, autonomie, initiative et hauteur de vue qui sont les clés du succès aux concours d'ingénieurs. Ne passez pas un temps inutilement long sur des parties ou des questions dont les thématiques sont très étroites (par exemple, une partie testant uniquement l'intégration ou uniquement la gestion des listes etc. à travers 10 questions !). Lorsque vous avez validé une thématique (par exemple les intégrales à paramètre ou la compacité ou les bases de données (informatique commune) ou les automates (option informatique)) c'est-à-dire que vous maîtrisez ou avez consolidé les points fondamentaux associés, travaillez des parties ou des sujets ne l'abordant pas. Une fois que vous avez couvert les points fondamentaux de votre programme et s'il vous reste du temps, n'hésitez pas à faire des sujets intégraux en temps limité (3 ou 4 heures) si possible sur des problématiques que vous n'avez jamais abordé (pour apprendre à gérer la nouveauté). Si vous n'avez pas le temps, ce n'est pas grave, l'important est d'avoir la couverture maximale du programme. Ne négligez pas le programme de première année qui est évalué régulièrement aux concours dont voici une liste non exhaustive : nombres complexes ou polynômes/fractions rationnels en mathématiques, représentation des réels, bases de données ou schéma d'Euler en informatique commune, liste en option informatique).

Table des matières

ÉPREUVE	SUJET	ÉNONCÉ	SOLUTION
CCINP 2021			
Maths 1	Calcul d'une intégrale. Extrémums d'une fonction. Un peu d'arithmétique avec la fonction zêta de Riemann. <i>Python. Intégration. Calcul différentiel. Fonctions d'une variable réelle. Séries numériques. Séries de fonctions. Familles sommables. Probabilités. Variables aléatoires.</i>	13	18
Maths 2	Théorème de décomposition de Dunford. <i>Espaces euclidiens. Réduction. Algèbre linéaire. Polynômes et fractions rationnelles. Espaces vectoriels normés.</i>	29	33
Option Info	Étude des partitions non croisées. Logique et étude du problème Horn-Sat. Étude des classes sylvestres. <i>programmation en Caml et en Python, programmation récursive et impérative, analyse de code informatique, dénombrement, logique, arbres, mots, langage.</i>	45	56
Centrale-SupElec 2021			
Maths 1	La loi du demi-cercle <i>calcul matriciel, équations différentielles, espaces euclidiens, fonction d'une variable réelle à valeurs vectorielles, réduction, réduction des matrices symétriques.</i>	69	75
Maths 2	Inégalités de Bernstein <i>arithmétique, polynômes, probabilités, séries entières, séries numériques, variables aléatoires.</i>	95	99

Option Info	Arbres couvrants et pavages	115	124

automates, complexité, files, graphes, langages, logique, preuve de correction, preuve de terminaison, programmation.

Mines-Ponts 2021

ITC	Marchons, marchons, marchons...	139	150

SQL, base de données, gestion de fichiers, programmation, méthodes numériques, schéma d'Euler, récursivité, aléatoire, complexité, modélisation.

Maths 1	Théorème de De Moivre-Laplace	157	162

algèbre linéaire, analyse asymptotique, espaces vectoriels normés, intégration, séries entières, suites numériques, topologie, valeurs propres (en dimension infinie).

Maths 2	Fonctions de matrices symétriques, continuité et convexité	185	191

convexité, endomorphismes des espaces euclidiens, espaces euclidiens, espaces vectoriels normés, topologie.

Option Info	Le jeu du solitaire	201	213

automates, complexité, décomposition binaire, exponentiation, programmation, langages, listes, tableaux.

CCINP 2022

Maths 1	Exercice Python. Intégrales de Fresnel	223	232

algèbre linéaire, complexité algorithmique, espaces préhilbertiens, fonction d'une variable réelle, intégrales à paramètre, intégration, polynômes, programmation, récursivité, série entière, séries numériques, suites de fonctions.

Maths 2	Application des déterminants de Vandermonde. Exponentielle de matrices. Produit de Hadamard de deux matrices	243	248

algèbre linéaire, équations différentielles, espaces préhilbertiens, exponentielles de matrices, probabilités, matrices, réduction des endomorphismes et des matrices, suites numériques, variables aléatoires.

Option Info	Automates : Calcul réussi au seuil logique et énigmes. Autour des tas. Autour de l'énumération des fractions positives.	261	269

Automates, mots, langages, programmation en Caml et en Python, programmation récursive et impérative, analyse de code informatique, invariant de boucle, complexité.

Centrale-SupElec 2022

Maths 1	Exemples de contraintes symplectiques linéaires	281	286

convexité, espaces euclidiens, espaces vectoriels normés, fonctions d'une variable réelle, intégration, matrices, probabilités, topologie, variables aléatoires.

Maths 2	Inégalités d'interpolation des dérivées. Convergence d'une série aléatoire de Rademacher. Dérivation pour des séries aléatoires de fonctions	301	305

calcul algébrique, calcul différentiel, équations différentielles, fonctions d'une variable réelle, intégrale à paramètre, intégration, séries de fonctions, séries entières, suites de fonctions, topologie.

Option Info	Tour d'horizon (assez complet) des automates et langages	319	328

complexité, correction, enregistrements, file, liste, graphes, programmation, tableaux, terminaison.

Mines-Ponts 2022

ITC	Modélisation numérique d'un matériau magnétique	349	362

programmation, méthodes numériques, dichotomie, base de données, SQL, aléatoire, complexité, modélisation, pile, récursivité, modélisation.

Maths 1	Formule asymptotique de Hardy et Ramanujan	371	379

algorithmes, intégration, nombres entiers, nombres réels, probabilités, suites numériques, variables aléatoires.

Maths 2	Autour des exponentielles de matrices.	397	404
	<i>calcul algébrique, calcul différentiel, espaces euclidiens, espaces vectoriels normés, matrices, réduction des matrices et des endomorphismes, suites numériques.</i>		
Option Info	Une mesure des erreurs de saisie. Fouille dans un trie. Système de transitions des voisins d'un mot brouillé.	417	427
	<i>algorithmes, automates, chaînes de caractères, complexité, enregistrements, langages, listes, programmation.</i>		
CCINP 2023			
Maths 1	Exercice Python. Calcul différentiel. La formule des compléments de la fonction Gamma.	437	442
	<i>familles sommables, intégrabilité, séries de fonctions, séries entières, variables aléatoires.</i>		
Maths 2	Exercice sur les espaces euclidiens. Exercice variable aléatoire. Diagonalisabilité via les projecteurs.	453	458
	<i>algèbre linéaire, espaces euclidiens, fonctions d'une variable réelle, matrices, polynômes, programmation, récursivité, réduction des matrices et des endomorphismes.</i>		
Option Info	sélection du $(k + 1)$ e plus petit élément. Recherche d'une clique de célébrités. Étude d'une famille d'automates	471	479
	<i>Programmation en Caml et en Python, programmation récursive et impérative, analyse de code informatique, complexité, graphes, automates, mots, langages.</i>		
Centrale-SupElec 2023			
Maths 1	Sur le calcul ombral.	487	492
	<i>algèbre linéaire, isométries, matrices, polynômes, réduction des endomorphismes.</i>		

Maths 2	Inégalité maximale de Komolgorov. Théorème central limite. <i>algèbre générale, dénombrabilité, fonctions d'une variable réelle, intégrales à paramètre, intégration, nombres réels, probabilités, suites de fonctions, trigonométrie, variables aléatoires.</i>	515	520
Option Info	Différentes représentations classiques d'ensembles. Arbres de van Emde Boas <i>bijections, combinaisons, circuits logiques (hors-programme), décomposition combinatoire d'un nombre, listes, modélisation, programmation, représentation binaire, tableaux.</i>	539	544

Mines-Ponts 2023

ITC	La typographie informatisée. <i>analyse de codes informatiques, bases de données, SQL, programmation, programmation dynamique, récursivité, dictionnaires, complexité, chaînes de caractères.</i>	561	570
Maths 1	Théorème de stabilité de Liapounov <i>calcul algébrique, équations différentielles, fonctions d'une variable réelle, probabilités, séries entières, séries numériques, suites numériques, variables aléatoires.</i>	581	587
Maths 2	Fonction de Wallis <i>algèbre linéaire, calcul algébrique, équations différentielles, espaces euclidiens, espaces vectoriels normés, intégrales à paramètre, intégration, matrices, nombres réels, réduction des matrices et des endomorphismes, suites numériques.</i>	597	603
Option Info	jeu du hanjie <i>automates, complexité, langages, listes, programmation, tableaux, tuple.</i>	619	631

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES 1**Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de deux exercices et d'un problème.

EXERCICE I

On note f la fonction définie sur $]0,1[$ par :

$$f(t) = \frac{\ln t}{t^2 - 1}.$$

Q1. Soit $k \in \mathbb{N}$. Justifier l'existence puis calculer l'intégrale

$$I_k = \int_0^1 t^{2k} \ln t \, dt.$$

Q2. Justifier que la fonction f est intégrable sur $]0,1[$, puis démontrer que :

$$\int_0^1 f(t) dt = \frac{\pi^2}{8}.$$

On pourra utiliser librement que :

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6}.$$

EXERCICE II

Q3. Justifier que la fonction $\ln$ est concave sur $]0, +\infty[$ et en déduire que :

$$\forall (a, b, c) \in]0, +\infty[^3, \quad \sqrt[3]{abc} \leq \frac{a + b + c}{3}.$$

On note f la fonction définie sur $]0, +\infty[^2$ par :

$$f(x; y) = x + y + \frac{1}{xy}.$$

Q4. Démontrer que f admet un unique point critique sur l'ouvert $]0, +\infty[^2$, puis démontrer que f admet un extremum global que l'on déterminera.

PROBLÈME

Un peu d'arithmétique avec la fonction zêta de Riemann

On note ζ la fonction zêta de Riemann définie sur $]1, +\infty[$ par :

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

Le problème est constitué de trois parties indépendantes dans une large mesure.

Partie I - Algorithmique : calcul de zêta aux entiers pairs

La suite des nombres de Bernoulli notée $(b_n)_{n \in \mathbb{N}}$ est définie par :

$$b_0 = 1, \quad \forall n \geq 1, \quad b_n = \frac{-1}{n+1} \sum_{k=0}^{n-1} \binom{n+1}{k} b_k.$$

Leonhard Euler (1707-1783) a démontré la formule suivante qui exprime les nombres $\zeta(2k)$ à l'aide des nombres de Bernoulli :

$$\forall k \in \mathbb{N}^*, \zeta(2k) = \frac{(-1)^{k-1} 2^{2k-1} \pi^{2k} b_{2k}}{(2k)!}.$$

Dans cette partie (informatique pour tous), on se propose de programmer le calcul des nombres de Bernoulli b_n afin d'obtenir des valeurs exactes de $\zeta(2k)$.

Les algorithmes demandés doivent être écrits en langage Python. On sera très attentif à la rédaction du code notamment à l'indentation.

Q5. Écrire une fonction `factorielle(n)` qui renvoie la factorielle d'un entier $n \in \mathbb{N}$.

Q6. On considère la fonction Python suivante `binom(n,p)` qui renvoie le coefficient binomial $\binom{n}{p}$:

```
def binom(n, p):
    if not (0 <= p <= n):
        return 0
    return factorielle(n) // (factorielle(p) * factorielle(n-p))
```

Combien de multiplications sont effectuées lorsque l'on exécute `binom(30, 10)` ?

Expliquer pourquoi il est possible de réduire ce nombre de multiplications à 20 ? Quel serait le type du résultat renvoyé si l'on remplaçait la dernière ligne de la fonction `binom` par

```
return factorielle(n) / (factorielle(p) * factorielle(n-p)) ?
```

Q7. Démontrer que, pour $n \geq p \geq 1$, on a

$$\binom{n}{p} = \frac{n}{p} \binom{n-1}{p-1}.$$

En déduire une fonction récursive `binom_rec(n,p)` qui renvoie le coefficient binomial $\binom{n}{p}$.

Q8. Écrire une fonction non récursive `bernoulli(n)` qui renvoie une valeur approchée du nombre rationnel b_n . On pourra utiliser librement une fonction `binomial(n,p)` qui renvoie le coefficient binomial $\binom{n}{p}$.

Par exemple `bernoulli(10)` renvoie 0,075 757 575 757 575 76 qui est une valeur approchée de $b_{10} = \frac{5}{66}$.

Partie II - Généralités sur la fonction zêta

Pour tout $n \in \mathbb{N}^*$, on note f_n la fonction définie sur $]1, +\infty[$ par :

$$f_n(x) = \frac{1}{n^x}.$$

Q9. Pour tout $a > 1$ réel, démontrer que la série $\sum \frac{\ln n}{n^a}$ converge.

Q10. Démontrer que la fonction ζ est de classe C^1 sur $]1, +\infty[$, puis qu'elle est décroissante.

Q11. La série de fonctions $\sum f_n$ converge-t-elle uniformément sur $]1, +\infty[$?

Q12. Déterminer la limite de ζ en $+\infty$.

Q13. Soit $x > 1$. On pose :

$$I(x) = \int_1^{+\infty} \frac{dt}{t^x}.$$

Démontrer que :

$$I(x) \leq \zeta(x) \leq I(x) + 1.$$

En déduire un équivalent de ζ au voisinage de 1.

Q14. Un premier lien avec l'arithmétique : pour tout $n \in \mathbb{N}^*$, on note d_n le nombre de diviseurs de l'entier n . On pose $A = \mathbb{N}^* \times \mathbb{N}^*$ et on prend $x > 1$. Justifier que la famille $\left(\frac{1}{(ab)^x}\right)_{(a,b) \in A}$ est sommable et que sa somme vaut $\zeta(x)^2$. En déduire que :

$$\zeta^2(x) = \sum_{n=1}^{+\infty} \frac{d_n}{n^x}.$$

On pourra considérer la réunion $\bigcup_{n \in \mathbb{N}^*} A_n$ où $A_n = \{(a, b) \in A, ab = n\}$.

Partie III - Produit eulérien

Soit $s > 1$ un réel fixé. On définit une variable aléatoire X à valeurs dans $\mathbb{N}^*$ sur un espace probabilisé $(\Omega, \mathcal{A}, P)$ par :

$$\forall k \in \mathbb{N}^*, P(X = k) = \frac{1}{\zeta(s)k^s}.$$

On rappelle qu'un entier a divise un entier b s'il existe un entier c tel que $b = ac$. On note alors $a|b$.

Q15. Soit $a \in \mathbb{N}^*$. Démontrer que $P(X \in a\mathbb{N}^*) = \frac{1}{a^s}$.

Q16. Soient $a_1, a_2, \dots, a_n$ dans $\mathbb{N}^*$ des entiers premiers entre eux deux à deux et $N \in \mathbb{N}^*$.
Démontrer par récurrence sur n que :

$$(a_1|N, a_2|N, \dots, a_n|N) \Leftrightarrow a_1 \times a_2 \times \dots \times a_n|N.$$

Le résultat persiste-t-il si les entiers $a_1, a_2, \dots, a_n$ sont seulement supposés premiers dans leur ensemble, c'est-à-dire lorsque leur PGCD vaut 1 ?

Q17. En déduire que si $a_1, a_2, \dots, a_n$ sont des entiers de $\mathbb{N}^*$ premiers entre eux deux à deux, alors les événements $[X \in a_1\mathbb{N}^*], \dots, [X \in a_n\mathbb{N}^*]$ sont mutuellement indépendants.
On pourra noter $(b_1, \dots, b_r)$ une sous-famille de la famille $(a_1, \dots, a_n)$.

On note $(p_n)_{n \in \mathbb{N}^*} = (2, 3, 5, 7, 11, \dots)$ la suite croissante des nombres premiers.
Pour tout entier $n \in \mathbb{N}^*$, on note B_n l'ensemble des $\omega \in \Omega$ tels que $X(\omega)$ n'est divisible par aucun des nombres premiers $p_1, p_2, \dots, p_n$.

Q18. Soit $n \in \mathbb{N}^*$. Déduire des questions précédentes que :

$$P(B_n) = \prod_{k=1}^n \left(1 - \frac{1}{p_k^s}\right).$$

Q19. Soit ω dans $\bigcap_{n \in \mathbb{N}^*} B_n$. Que vaut $X(\omega)$? En déduire que :

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

On se propose, en application, de prouver que la série $\sum \frac{1}{p_n}$ des inverses des nombres premiers diverge. On raisonne pour cela par l'absurde en supposant que la série $\sum \frac{1}{p_n}$ converge.
On pose pour tout $n \in \mathbb{N}^*$,

$$u_n = \prod_{k=1}^n \frac{1}{1 - \frac{1}{p_k}}.$$

Q20. Justifier que la suite (u_n) converge vers un réel l et que l'on a pour tout réel $s > 1$, $l \geq \zeta(s)$.
Conclure.

FIN
5/5

SOLUTION

Q1. La fonction $f_k(t) = t^{2k} \ln(t)$ est continue sur $]0, 1]$ et $f_k(t) \underset{t \rightarrow 0}{=} o\left(\frac{1}{t^{1/2}}\right)$ car

$$t^{1/2} f_k(t) = t^{2k+1/2} \ln(t) \xrightarrow{t \rightarrow 0} 0$$

d'après les croissances comparées (puisque $2k + \frac{1}{2} > 0$). La fonction $t \mapsto \frac{1}{t^{1/2}}$ étant intégrable sur $]0, 1]$ (intégrale de Riemann de paramètre $\frac{1}{2} < 1$), on en déduit que la fonction f l'est aussi. En outre, par intégration par parties (en intégrant t^{2k} en $\frac{t^{2k+1}}{2k+1}$ et en dérivant $\ln(t)$ en $\frac{1}{t}$), on obtient

$$\begin{aligned} \int_0^1 f_k(t) dt &= \left[\frac{t^{2k+1}}{2k+1} \ln(t) \right]_{t=0}^{t=1} - \int_0^1 \frac{t^{2k+1}}{2k+1} \times \frac{1}{t} dt = -\frac{1}{2k+1} \int_0^1 t^{2k} dt \\ &= -\frac{1}{2k+1} \left[\frac{t^{2k+1}}{2k+1} \right]_{t=0}^{t=1} = -\frac{1}{(2k+1)^2}. \end{aligned}$$

Commentaires : Le calcul est standard et ne doit pas poser de problème. Par contre, l'existence de l'intégrale est trop souvent mal rédigé par les candidats : contrinuité, étude en chaque point, mention et justification des comparaisons. La question étant standard et sans difficulté particulière, les correcteurs de tous les concours sont absolument intransigeants sur la rédaction.

Q2. Intégrabilité. La fonction $f(t) = \frac{\ln(t)}{t^2 - 1}$ est continue sur $]0, +\infty[$. Comme $f(t) \underset{t \rightarrow 0}{\sim} \frac{\ln(t)}{-1} = o\left(\frac{1}{t^{1/2}}\right)$ (car $t^{1/2} \ln(t) \xrightarrow{t \rightarrow 0} 0$ d'après les croissances comparées puisque $\frac{1}{2} > 0$) et que la fonction $t \mapsto \frac{1}{t^{1/2}}$ est intégrable sur $]0, 1]$ (intégrale de Riemann de paramètre $\frac{1}{2} < 1$), on en déduit que la fonction f l'est aussi.

Preuve de l'égalité. D'après la série géométrique, comme $|t^2| = |t|^2 < 1$ si $t \in [0, 1[$, on peut affirmer que

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

Nous allons utiliser un théorème de permutation série-intégrale. Pour chaque entier k , la fonction f_k est continue et intégrable sur $]0, 1[$. La série $\sum_{k \geq 0} (-f_k)$ converge simplement

sur $]0, 1[$ et sa somme $\sum_{k=0}^{+\infty} (-f_k) = f$ est continue sur cet intervalle. En outre, la série

$\sum_{k \geq 0} \int_0^1 |f_k| = \sum_{k \geq 0} -\int_0^1 f_k$ (car f_k est positive sur cet intervalle) $= \sum_{k \geq 0} \frac{1}{(2k+1)^2}$ converge (son terme général est équivalent à $\frac{1}{(2k)^2} = \frac{1}{4k^2}$ qui est le terme général d'une série positive

convergente, série de Riemann de paramètre $\alpha = 2 > 1$). Par conséquent, on peut affirmer que

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

En scindant la somme $\sum_{n=1}^{+\infty} \frac{1}{n^2}$ selon les indices pairs et impairs, on obtient

$$\begin{aligned} \frac{\pi^2}{6} &= \sum_{n=1}^{+\infty} \frac{1}{n^2} = \sum_{\substack{n \in \mathbb{N}^* \\ n \text{ pair} \\ \text{i.e. } n=2k \text{ avec } k \in \mathbb{N}^*}} \frac{1}{n^2} + \sum_{\substack{n \in \mathbb{N}^* \\ n \text{ impair} \\ \text{i.e. } n=2k+1 \text{ avec } k \in \mathbb{N}}} \frac{1}{n^2} \\ &= \sum_{k=1}^{+\infty} \frac{1}{(2k)^2} + \sum_{k=0}^{+\infty} \frac{1}{(2k+1)^2} = \frac{1}{4} \sum_{k=1}^{+\infty} \frac{1}{k^2} + \sum_{k=0}^{+\infty} \frac{1}{(2k+1)^2}. \end{aligned}$$

Ainsi, nous venons de prouver l'égalité

$$\frac{\pi^2}{6} = \frac{1}{4} \times \frac{\pi^2}{6} + \int_0^1 f \Leftrightarrow \int_0^1 f = \pi^2 \left(\frac{1}{6} - \frac{1}{24} \right) = \frac{\pi^2}{24} (4-1) = \frac{\pi^2 3}{24} = \frac{\pi^2}{8}.$$

Commentaires : Il s'agit d'une question ouverte qui demande autonomie, initiative et technicité de la part du candidat donc elle est nécessaire sélective (mais bien rétribuée par le barème). Si la permutation série-intégrale est formellement faite par un grand nombre de candidats, beaucoup trop de candidats ne rédigent pas correctement les hypothèses nécessaires au théorème utilisé. Je ne parle pas d'une minorité (importante) qui ne voit même pas qu'il y a un théorème à mettre en place ! Soyez méticuleux et rigoureux.

Q3. La fonction $f = \ln$ est deux fois dérivable sur $]0, +\infty[$ et sa dérivée seconde $f'' : t \mapsto -\frac{1}{t^2}$ y est négative. Par conséquent, f est concave donc, pour tout entier $n \geq 1$, pour tous réels $\lambda_1, \dots, \lambda_n$ positifs de somme 1, pour tous réels $x_1, \dots, x_n$ de $]0, +\infty[$, on a :

$$f\left(\sum_{i=1}^n \lambda_i x_i\right) \geq \sum_{i=1}^n \lambda_i f(x_i) \Leftrightarrow \ln\left(\sum_{i=1}^n \lambda_i x_i\right) \geq \sum_{i=1}^n \lambda_i \ln(x_i).$$

En choisissant $n = 3$, $\lambda_1 = \lambda_2 = \lambda_3 = \frac{1}{3}$ et $x_1 = a$, $x_2 = b$, $x_3 = c$, on obtient :

$$\ln\left(\frac{a+b+c}{3}\right) \geq \frac{1}{3} (\ln(a) + \ln(b) + \ln(c)) = \frac{1}{3} \ln(abc) = \ln\left((abc)^{1/3}\right).$$

En composant cette inégalité par la fonction exponentielle qui est strictement croissante, on obtient l'inégalité

$$\frac{a+b+c}{3} \geq \sqrt[3]{abc}.$$

Commentaires : Il s'agit d'une question de cours.

Q4. La fonction f est différentiable sur $]0, +\infty[^2$ (comme somme de fractions rationnelles qui sont définies sur cet ensemble). Si $A = (x, y)$ est un point critique de f sur cet ensemble, on a :

$$\begin{cases} \frac{\partial f}{\partial x}(x, y) = 0 \\ \frac{\partial f}{\partial y}(x, y) = 0 \end{cases} \Leftrightarrow \begin{cases} 1 - \frac{1}{x^2 y} = 0 \\ 1 - \frac{1}{xy^2} = 0 \end{cases} \Leftrightarrow \begin{cases} \frac{1}{x^2 y} = 1 \\ \frac{1}{xy^2} = 1 \end{cases} \Leftrightarrow \begin{cases} x^2 y = 1 \\ xy^2 = 1 \end{cases}$$

En divisant la première égalité par la seconde, on obtient l'égalité $\frac{x}{y} = 1 \Leftrightarrow x = y$ puis en réinjectant cette égalité dans l'une des deux précédentes, on obtient $x^3 = 1 \Leftrightarrow x = 1$ donc $A = (1, 1)$ est unique.

Comme $f(A) = f(1, 1) = 3$ et en appliquant l'inégalité de la question précédente avec $a = x$, $b = y$, $c = \frac{1}{xy}$, on peut écrire

$$\sqrt[3]{xy \times \frac{1}{xy}} \leq \frac{1}{3} \left(x + y + \frac{1}{xy} \right) \Leftrightarrow f(A) = 3 \leq f(x, y)$$

donc A est un minimum global de f sur $]0, +\infty[^2$. La fonction f n'admet aucun maximum local sur cet ensemble. En effet, si c'était le cas, ce maximum se réalise en un point B de $]0, +\infty[^2$. Comme f est différentiable sur cet ouvert, B est un point critique de f donc $B = A$, ce qui est absurde car A est un minimum local et la fonction f n'est manifestement pas constante.

Conclusion : f n'admet aucun maximum local et un unique minimum (local et global) en $(1, 1)$.

Commentaires : La recherche des points critiques est relativement aisée et ne doit pas vous poser de problème particulier. Par contre, le second point (déterminer les extrémums et leurs natures) est d'une autre difficulté. Elle exige du candidat une connaissance solide de son cours (par exemple, pour exploiter la hessienne de f en $(1, 1)$) et / ou faire preuve d'autonomie et d'initiative (faire le lien avec la question précédente). Ce second point est destinée aux meilleures copies.

Q5.

```
def factorielle_rec(n):
    if n==0:
        return 1
    else:
        return n*factorielle_rec(n-1)

def factorielle_it(n):
    f=1
    for i in range(1,n+1):
        f=f*i
    return f
```

Commentaires : Question de cours sans aucune difficulté.

Q6. *factorielle*(n) effectue n multiplications donc *binom*(n, p) en fait $n + p + (n - p) = 2n$ (appelle de *factorielle* sur les entrées $n, p, n - p$) donc ici $2 * 30 = 60$.
On remarque alors que

$$\begin{aligned} \binom{n}{p} &= \frac{n!}{(n-p)!} \times \frac{1}{p!} = \frac{n(n-1) \cdots (n-p+1)(n-p)!}{(n-p)!} \times \frac{1}{p!} \\ &= \frac{n(n-1) \cdots (n-p+1)}{p!} \end{aligned}$$

donc, avec cette formule, il a p produits au numérateur et p au dénominateur ce qui fait $2p$ produits en tout. Dans le cas présent, $p = 10$ donc $2p = 20$ produits.

Commentaires : Question sélective car elle demande de savoir simplifier (classiquement) l'écriture des combinaisons $\binom{n}{p}$.

Q7.

```
def binom_rec(n,p):
    if not (0<=p<=n):
        return 0
    if p==0:
        return 1
    return (n/p)*binom_rec(n-1,p-1)
```

Commentaires : La seule difficulté de cette question est de gérer correctement les tests d'arrêts (valeur nulle en dehors de $0 \leq p \leq n$! Cet oubli sera fortement sanctionné).

Q8.

```
def bernoulli(n):
    b = [0 for i in range(n+1)]
    b[0]=1
    for m in range(1,n+1):
        bm=0
        for k in range(m):
            bm = bm + binom(m+1,k)*b[k]
        b[m] = -bm/(m+1)
    return b[n]
```

Commentaires : Question à destination des candidat(e)s ayant une aisance en programmation donc elle est sélective en pratique.

Q9. Comme $a > 1$, on peut choisir un réel $\varepsilon > 0$ tel que $a - \varepsilon$ soit strictement supérieur à 1. D'après les croissances comparées, on a $\ln(n) = o(n^\varepsilon)$ donc

$$\frac{\ln(n)}{n^a} = \frac{o(n^\varepsilon)}{n^a} = o\left(\frac{1}{n^{a-\varepsilon}}\right).$$

Comme la série $\sum_n \frac{1}{n^{a-\varepsilon}}$ converge (série de Riemann de paramètre $a - \varepsilon > 1$) et qu'elle est à termes positifs, on peut affirmer que la série $\sum_n \frac{\ln(n)}{n^a}$ converge.

Commentaires : Question de difficulté standard pour ce concours. Si vous n'avez pas réussi cette question, retravaillez là.

Q10. Pour tout entier $n \geq 1$, $f_n : x \mapsto \exp(-x \ln(n))$ est de classe C^1 sur $]1, +\infty[$. La série $\sum_n f_n$ converge simplement sur cet intervalle (série de Riemann de paramètre $x > 1$). Pour tout entier n , on a

$$f'_n : x \mapsto -\ln(n) \exp(-x \ln(n)) = \frac{-\ln(n)}{n^x}$$

donc, pour tout segment $[a, b]$ inclus dans $]1, +\infty[$ (i.e. $a > 1$), la série

$$\sum_{n \geq 1} \sup_{x \in [a, b]} |f'_n(x)| = \sum_{n \geq 1} \frac{\ln(n)}{n^a}$$

converge. Ainsi, la série $\sum_n f'_n$ converge normalement, donc uniformément, sur tout segment de $]1, +\infty[$ ce qui assure le caractère C^1 de $\sum_{n=1}^{+\infty} f_n = \zeta$ sur cet intervalle. En outre, sur celui-ci, on a :

$$\forall x > 1, \zeta'(x) = \sum_{n=1}^{+\infty} f'_n(x) = - \sum_{n=1}^{+\infty} \frac{\ln(n)}{n^x} < 0$$

(somme de termes négatifs dont l'un n'est pas nul, par exemple pour $n = 2$) donc ζ est strictement décroissante sur $]1, +\infty[$.

Commentaires : Même remarque que précédemment.

Q11. Supposons que la la série $\sum_n f_n$ converge uniformément sur $]1, +\infty[$. Comme chaque fonction f_n admet une limite finie quand $x \rightarrow 1$ (valant $\frac{1}{n}$), le théorème de double limite montre que la fonction $\sum_{n=1}^{+\infty} f_n = \zeta$ admet une limite finie en 1 et que

$$\lim_{x \rightarrow 1} \zeta(x) = \sum_{n=1}^{+\infty} \lim_{x \rightarrow 1} f_n(x) = \sum_{n=1}^{+\infty} \frac{1}{n}$$

ce qui est absurde car cette série diverge (série de Riemann de paramètre $\alpha = 1 \leq 1$). Par conséquent, la série $\sum_n f_n$ ne converge pas uniformément sur $]1, +\infty[$.

Commentaires : Question difficile destinée aux bonnes copies. Il faut songer à une preuve par l'absurde et penser aux théorèmes de permutation limite-série. Il ne faut pas penser que la convergence uniforme sur tout segment entraîne la convergence uniforme sur tout l'intervalle (si ce n'est pas un segment comme ici!).

Q12. Pour chaque entier $n \geq 1$, $\lim_{x \rightarrow +\infty} f_n(x) = \begin{cases} 0 & \text{si } n > 1 \\ 1 & \text{si } n = 1 \end{cases}$. La série $\sum_{n \geq 1} f_n$ converge normalement sur $[2, +\infty[$ (car $\sum_n \sup_{x \in [2, +\infty[} |f_n(x)| = \sum_n \frac{1}{n^2}$ converge, série de Riemann de paramètre $2 > 1$) donc uniformément sur cet intervalle. D'après le théorème de double limite, la fonction $\sum_{n=1}^{+\infty} f_n = \zeta$ a une limite en $+\infty$ qui vaut

$$\lim_{x \rightarrow +\infty} \zeta(x) = \sum_{n=1}^{+\infty} \lim_{x \rightarrow +\infty} f_n(x) = 1 + 0 + \cdots + 0 + \cdots = 1.$$

Commentaires : Question d'application directe du cours. Elle ne doit pas vous poser de problème. Il faut songer également au cas $n = 1$ pour la limite (qui est différent des cas généraux).

Q13. Comme $x \geq 0$, la fonction $g : t \mapsto \frac{1}{t^x}$ est décroissante sur l'intervalle $[1, +\infty[$. En particulier, pour tout entier $n \geq 1$, on a

$$\forall t \in [n, n+1], g(n+1) \leq g(t) \leq g(n).$$

En intégrant cette relation sur l'intervalle $[n, n+1]$, on obtient

$$g(n+1) \int_n^{n+1} dt \leq \int_n^{n+1} g \leq g(n) \int_n^{n+1} dt \Leftrightarrow g(n+1) \leq \int_n^{n+1} g \leq g(n)$$

puis en sommant sur $n \in \mathbb{N}^*$, on obtient grâce à la relation de Chasles :

$$\begin{aligned} \sum_{n=1}^{+\infty} g(n+1) &\leq \int_1^{+\infty} g \leq \sum_{n=1}^{+\infty} g(n) \\ \Leftrightarrow \sum_{k=2}^{+\infty} g(k) &= \zeta(x) - g(1) = \zeta(x) - 1 \leq I(x) \leq \zeta(x) \\ \Leftrightarrow \begin{cases} \zeta(x) - 1 \leq I(x) \\ I(x) \leq \zeta(x) \end{cases} &\Leftrightarrow \begin{cases} \zeta(x) \leq I(x) + 1 \\ I(x) \leq \zeta(x) \end{cases} \Leftrightarrow I(x) \leq \zeta(x) \leq I(x) + 1. \end{aligned}$$

Comme

$$I(x) = \int_1^{+\infty} \frac{dt}{t^x} = \int_1^{+\infty} t^{-x} dt = \left[\frac{1}{-x+1} t^{-x+1} \right]_{t=1}^{t \rightarrow +\infty} = \frac{1}{x-1}$$

(car $-x+1 > 0$ donc $\lim_{t \rightarrow +\infty} t^{-x+1} = 0$), on a établi l'encadrement

$$\frac{1}{x-1} \leq \zeta(x) \leq \frac{1}{x-1} + 1 \leq 0 \leq \zeta(x) - \frac{1}{x-1} \leq 1.$$

Ainsi, la fonction $\zeta(x) - \frac{1}{x-1}$ est bornée c'est-à-dire

$$\zeta(x) - \frac{1}{x-1} \underset{x \rightarrow 1}{=} O(1) \Leftrightarrow \zeta(x) \underset{x \rightarrow 1}{=} \frac{1}{x-1} + O(1) \underset{x \rightarrow 1}{\sim} \frac{1}{x-1}$$

(car $\frac{1}{x-1} \rightarrow 1$ quand $x \rightarrow 1^+$).

Commentaires : Question de difficulté standard. L'idée maitresse est d'encadrer la somme par deux quantités de même ordre de grandeur. Il est classique de penser alors à la comparaison série-intégrale. La question est finalement très discriminante (pour gérer notamment correctement l'encadrement et l'équivalent final).

Q14. On applique Fubini avec la famille de nombres positifs

$$u_{a,b} = \frac{1}{(ab)^x} = \frac{1}{a^x} \times \frac{1}{b^x}.$$

Fixons $a \geq 1$. La série $\sum_{b \geq 1} u_{a,b} = \frac{1}{a^x} \sum_{b \geq 1} \frac{1}{b^x}$ converge (série de Riemann de paramètre $x > 1$).

On pose

$$S_a = \sum_{b=1}^{+\infty} u_{a,b} = \frac{C}{a^x} \text{ avec } C = \sum_{b=1}^{+\infty} \frac{1}{b^x} = \zeta(x) \text{ (indépendant de } a).$$

La série $\sum_a S_a$ converge (série de Riemann de paramètre $x > 1$). D'après le théorème de Fubini, la famille $(u_{a,b})_{(a,b) \in (\mathbb{N}^*)^2}$ est sommable et sa somme vaut

$$\sum_{(a,b) \in (\mathbb{N}^*)^2} u_{a,b} = \sum_{a=1}^{+\infty} \sum_{b=1}^{+\infty} u_{a,b} = \sum_{a=1}^{+\infty} \frac{\zeta(x)}{a^x} = \zeta(x) \sum_{a=1}^{+\infty} \frac{1}{a^x} = \zeta(x) \zeta(x) = (\zeta(x))^2.$$

La famille $(A_n)_{n \in \mathbb{N}^*}$ est une partition de $(\mathbb{N}^*)^2$. En effet,

- A_n et A_m ne peuvent avoir d'intersection commune lorsque $n \neq m$ car si (a,b) appartenait aux deux, on aurait $ab = n$ et $ab = m$;
- si $(u,v) \in (\mathbb{N}^*)^2$ alors $(u,v) \in A_{uv}$!

D'après le théorème de sommation par paquets, on a :

$$(\zeta(x))^2 = \sum_{(a,b) \in (\mathbb{N}^*)^2} u_{a,b} = \sum_{n=1}^{+\infty} \sum_{(a,b) \in A_n} u_{a,b}$$

Or, pour tout entier n , on a :

$$\sum_{(a,b) \in A_n} u_{a,b} = \sum_{\substack{(a,b) \in (\mathbb{N}^*)^2 \\ ab=n}} \frac{1}{(ab)^x} = \sum_{\substack{(a,b) \in (\mathbb{N}^*)^2 \\ ab=n}} \frac{1}{n^x} = \frac{1}{n^x} \sum_{\substack{(a,b) \in (\mathbb{N}^*)^2 \\ ab=n}} 1 = \frac{1}{n^x} \text{card}(A_n).$$

Si $(a,b) \in A_n$ alors a est un diviseur de n (car $ab = n$!). Si d est un diviseur de n alors $\frac{n}{d}$ est un entier et $(d, \frac{n}{d}) \in A_n$. Par conséquent, l'application $d \mapsto (d, \frac{n}{d})$ est une bijection entre les diviseurs de n et les éléments de A_n donc ces deux ensembles ont le même nombre d'éléments. Autrement dit, nous venons de prouver que $\text{card}(A_n) = d_n$ d'où

$$(\zeta(x))^2 = \sum_{n=1}^{+\infty} \frac{d_n}{n^x}.$$

Commentaires : Question difficile destinée aux bonnes copies. Il faut songer au théorème de Fubini et à la sommation par paquets. Chacun de ces résultats (appliqué et justifié correctement) sera fortement valorisé par le correcteur. Comme d'habitude, Fubini est assez bien traité et la sommation par paquet s'avère difficile pour un nombre important de candidats à ce concours. N'hésitez pas à travailler cette question car elle apparaît relativement souvent à ce concours (et à Mines et Centrale).

Q15. Par définition, on a :

$$(X \in a\mathbb{N}^*) \Leftrightarrow \exists k \in \mathbb{N}^*, X = ak \Leftrightarrow X \in \bigcup_{k=1}^{+\infty} (X = ak).$$

Cette dernière union étant disjointe, on peut écrire

$$\begin{aligned} P(X \in a\mathbb{N}^*) &= \sum_{k=1}^{+\infty} P(X = ak) = \sum_{k=1}^{+\infty} \frac{1}{(ak)^s \zeta(s)} \\ &= \frac{1}{\zeta(s) a^s} \sum_{k=1}^{+\infty} \frac{1}{k^s} = \frac{1}{\zeta(s) a^s} \zeta(s) = \frac{1}{a^s}. \end{aligned}$$

Commentaires : Question élémentaire où la rédaction se doit d'être rigoureuse sous peine d'être très lourdement sanctionnée (événements incompatibles).

Q16. Une implication est immédiate (et sans récurrence). Si $a_1 \cdots a_n$ divise N alors, comme a_i divise $a_1 \cdots a_n$ pour chaque $i \in \{1, \dots, n\}$, on est assuré que a_i divise N .

Prouvons par récurrence sur n la réciproque. Si $n = 1$, c'est immédiat. Soit $n \geq 2$ et supposons que l'assertion

$$(a_1|N, \dots, a_{n-1}|N) \Rightarrow (a_1 \cdots a_{n-1}|N)$$

soit vraie (phase d'hérédité). Si $(a_1|N, \dots, a_n|N)$ est vraie alors on a :

- $(a_1|N, \dots, a_{n-1}|N)$ est vérifiée donc par hypothèse de récurrence, $a_1 \cdots a_{n-1}$ divise N
- a_n divise N

Si d est un diviseur de $a_1 \cdots a_{n-1}$, comme $a_1, \dots, a_{n-1}$ sont premiers entre eux, le lemme de Gauss entraîne que d divise l'un des a_i , $1 \leq i \leq n-1$. En outre, si d divise a_n alors d divise l'un des a_i , $1 \leq i \leq n-1$, et a_n . Comme ces deux entiers sont premiers entre eux, on a nécessairement $d = 1$ donc $a_1 \cdots a_{n-1}$ et a_n sont premiers entre eux. Comme ils divisent tous deux N , par lemme de Gauss, on est assuré que $a_1 \cdots a_{n-1} a_n$ divise N , ce qui prouve l'hérédité et achève la récurrence.

Le résultat est faux si on suppose les entiers a_i premiers entre eux dans leur ensemble. Par exemple, pour $n = 3$, $a_1 = 2$, $a_2 = 3$, $a_3 = 6$, ces trois entiers sont premiers entre eux dans leur ensemble (le seul diviseur commun aux trois est 1). Chacun de ces entiers divise $N = 12$ au $a_1 a_2 a_3 = 36$ ne divise pas $N = 12$.

Commentaires : Question difficile pour la plupart des candidats à ce concours. L'arithmétique étant souvent mal maîtrisée, ainsi que les différents théorèmes (ou lemmes) de Gauss, la question est finalement destinée aux meilleures copies.

Q17. Si chaque évènement $(X \in a_i \mathbb{N}^*)$ est vérifiée ($1 \leq i \leq n$) alors a_i divise X pour chaque $i \in \{1, \dots, n\}$ donc $a_1 \cdots a_n$ divise X donc $X \in a_1 \cdots a_n \mathbb{N}^*$ (prendre éventuellement des $\omega \in \Omega$ alors a_i divise $X(\omega)$, etc) d'où l'inclusion

$$\bigcap_{i=1}^n (X \in a_i \mathbb{N}^*) \subset (X \in a_1 \cdots a_n \mathbb{N}^*).$$

Si l'évènement $(X \in a_1 \cdots a_n \mathbb{N}^*)$ est vérifiée alors $a_1 \cdots a_n$ divise X donc, pour chaque $i \in \{1, \dots, n\}$, a_i divise X c'est-à-dire $X \in a_i \mathbb{N}^*$. Autrement dit, on vient d'établir l'inclusion

$$(X \in a_1 \cdots a_n \mathbb{N}^*) \subset \bigcap_{i=1}^n (X \in a_i \mathbb{N}^*).$$

d'où l'égalité ensembliste

$$(X \in a_1 \cdots a_n \mathbb{N}^*) = \bigcap_{i=1}^n (X \in a_i \mathbb{N}^*).$$

On en déduit les égalités suivantes :

$$\begin{aligned} P\left(\bigcap_{i=1}^n (X \in a_i \mathbb{N}^*)\right) &= P((X \in a_1 \cdots a_n \mathbb{N}^*)) = \frac{1}{(a_1 \cdots a_n)^s} \\ &= \frac{1}{(a_1)^s} \times \cdots \times \frac{1}{(a_n)^s} \\ &= P(X \in a_1 \mathbb{N}^*) \cdots P(X \in a_n \mathbb{N}^*) \end{aligned}$$

ce qui prouve que les évènements $(X \in a_1 \mathbb{N}^*)$, ..., $(X \in a_n \mathbb{N}^*)$ sont indépendants.

Commentaires : Question de difficulté standard. Retravaillez la si elle vous a posé problème.

Q18. Par définition, $\omega \in B_n$ si et seulement si (p_1 ne divise pas $X(\omega)$) et (p_2 ne divise pas $X(\omega)$) et ... et (p_n ne divise pas $X(\omega)$) si et seulement si $(X(\omega) \notin p_1 \mathbb{N}^*)$ et ... $(X(\omega) \notin p_n \mathbb{N}^*)$ d'où l'égalité ensembliste

$$B_n = \overline{(X \in p_1 \mathbb{N}^*)} \cap \cdots \cap \overline{(X \in p_n \mathbb{N}^*)}$$

Les évènements $(X \in p_i \mathbb{N}^*)_{1 \leq i \leq n}$ étant mutuellement indépendants, le lemme des coalitions montre que les évènements contraires $\left(\overline{(X \in p_i \mathbb{N}^*)}\right)_{1 \leq i \leq n}$ sont également indépendants. On peut alors écrire

$$P(B_n) = \prod_{i=1}^n P\left(\overline{(X \in p_i \mathbb{N}^*)}\right) = \prod_{i=1}^n (1 - P(X \in p_i \mathbb{N}^*)) = \prod_{i=1}^n \left(1 - \frac{1}{(p_i)^s}\right).$$

Commentaires : Question destinée aux bonnes copies (sans être excellente). La question est classique (dans le cas $\Omega = \mathbb{N}$ et dans le cas $\Omega = \{1, \dots, n\}$, on retrouve ailleurs le calcul de l'indicatrice d'Euler $\varphi(n)$).

Q19. Si $\omega \in \bigcap_{n=1}^{+\infty} B_n$ alors, pour tout entier n , ni p_1 , ni p_2 , ..., ni p_n ne divise $X(\omega)$ c'est-à-dire que $X(\omega)$ n'est divisible par aucun nombre premier. Comme tout entier supérieur à 2 possède au moins un diviseur premier, on en déduit que $X(\omega)$ vaut forcément 1 (car il est non nul). Réciproquement, si $X(\omega)$ vaut 1, il appartient à tous les B_n , $n \in \mathbb{N}$. Nous venons de prouver l'égalité ensembliste

$$\bigcap_{n=1}^{+\infty} B_n = (X = 1).$$

Les événements $(B_n)_{n \geq 1}$ forment une famille d'événements décroissante car $\omega \in B_{n+1}$ donc ni p_1 , ni p_2 , ..., ni p_{n+1} ne divise $X(\omega)$ donc ni p_1 , ni p_2 , ..., ni p_n ne divise $X(\omega)$ donc $\omega \in B_n$. D'après le théorème de limite monotone, on a

$$\lim_{N \rightarrow +\infty} P(B_N) = P\left(\bigcap_{N=1}^{+\infty} B_N\right) = P(X = 1) = \frac{1}{\zeta(x)}$$

Et comme $P(B_N) = \prod_{i=1}^n \left(1 - \frac{1}{(p_i)^s}\right)$, on a établi la formule voulue.

Commentaires : Même remarque que précédemment.

Q20. Pour tout entier n , on a $\ln(u_n) = \sum_{k=1}^n \ln\left(1 - \frac{1}{p_k}\right)$. Comme $p_k \xrightarrow[k \rightarrow +\infty]{} +\infty$, on peut affirmer que $\frac{1}{p_k} \xrightarrow[k \rightarrow +\infty]{} 0$ ce qui nous donne l'équivalent suivant (obtenu l'équivalent de $\ln(1+x)$ en 0) :

$$\ln\left(1 - \frac{1}{p_k}\right) \underset{k \rightarrow +\infty}{\sim} -\frac{1}{p_k}.$$

Comme la série $\sum_k \frac{1}{p_k}$ converge (par hypothèse), la série $\sum_k \left(-\ln\left(1 - \frac{1}{p_k}\right)\right)$ converge donc la suite

$$\sum_{k=1}^n \left(-\ln\left(1 - \frac{1}{p_k}\right)\right) = -\ln(u_n)$$

converge vers un certain réel L . En composant par l'exponentielle, la suite u_n converge vers $\ell = \exp(-L)$ qui est un réel.

Pour tout entier $k \geq 1$, on a $\frac{1}{p_k} \leq 1$ donc la fonction $s \mapsto \frac{1}{(p_k)^s}$ est décroissante donc la fonction $s \mapsto 1 - \frac{1}{(p_k)^s}$ est croissante d'où la croissance de la fonction $s \mapsto \frac{1}{1 - \frac{1}{(p_k)^s}}$. En

particulier, on a

$$\forall s > 1, \forall k \geq 1, \frac{1}{1 - \frac{1}{(p_k)^s}} \leq \frac{1}{1 - \frac{1}{p_k}}.$$

En multipliant ces inégalités pour $k \in \{1, \dots, n\}$, on obtient

$$P(B_n) = \prod_{k=1}^n \frac{1}{1 - \frac{1}{(p_k)^s}} \leq u_n.$$

En faisant tendre n vers $+\infty$ dans cette inégalité, on obtient d'après la question précédente, l'inégalité :

$$\zeta(s) \leq \ell.$$

Ainsi, la fonction ζ est majorée sur l'intervalle $[1, +\infty[$, ce qui contredit la réponse à la question 13 (qui a pour conséquence que $I(s) = \frac{1}{s-1} \leq \zeta(s)$ est majorée sur cet intervalle alors qu'elle tend vers $+\infty$ quand $s \rightarrow 1$). Conclusion : la série $\sum_k \frac{1}{p_k}$ diverge.

Commentaires : *Question difficile destinée aux meilleures copies.*

**ÉPREUVE SPÉCIFIQUE - FILIÈRE MP****MATHÉMATIQUES 2****Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé d'un exercice et d'un problème indépendants.

Notations pour l'ensemble du sujet :

K désigne $\mathbb{R}$ ou $\mathbb{C}$.

On note, pour n entier naturel, $n \geq 2$:

- $M_n(K)$ le K -espace vectoriel des matrices carrées de taille n à coefficients dans K .
- $D_n(\mathbb{R})$ le sous-espace vectoriel des matrices diagonales de $M_n(\mathbb{R})$.

EXERCICE

Q1. On munit $M_n(\mathbb{R})$ du produit scalaire canonique ($\langle A|B \rangle = \text{trace}({}^t A.B)$), déterminer $(D_n(\mathbb{R}))^\perp$, l'orthogonal de $D_n(\mathbb{R})$ pour ce produit scalaire.

PROBLÈME - Théorème de décomposition de Dunford

On admet le théorème suivant que l'on pourra utiliser librement :

Si A est une matrice de $M_n(K)$ telle que son polynôme caractéristique χ_A soit scindé sur K , alors il existe un unique couple (D, N) de matrices de $M_n(K)$ vérifiant les quatre propriétés :

- (1) $A = D + N$;
- (2) D est diagonalisable dans $M_n(K)$ (pas nécessairement diagonale) ;
- (3) N est nilpotente ;
- (4) $DN = ND$.

De plus, D et N sont des polynômes en A et $\chi_A = \chi_D$.

Le couple (D, N) s'appelle la décomposition de Dunford de A .

Partie I - Quelques exemples

Q2. Donner le couple de la décomposition de Dunford d'une matrice A de $M_n(K)$ lorsque A est diagonalisable, puis lorsque la matrice A de $M_n(K)$ est nilpotente.

Justifier qu'une matrice trigonalisable vérifie l'hypothèse du théorème, admettant ainsi une décomposition de Dunford.

Le couple de matrices $\left(\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \right)$ est-il la décomposition de Dunford de la matrice

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

Q3. Donner un exemple d'une matrice de $M_2(\mathbb{R})$ n'admettant pas de décomposition de Dunford dans $M_2(\mathbb{R})$.

Q4. Soit la matrice $A = \begin{pmatrix} 3 & 0 & 8 \\ 3 & -1 & 6 \\ -2 & 0 & -5 \end{pmatrix}$.

Calculer son polynôme caractéristique χ_A , puis donner le couple (D, N) de la décomposition de Dunford de A (on utilisera le fait que $\chi_A = \chi_D$).

Q5. Application

Pour $A \in M_n(\mathbb{K})$, $\exp(A) = \sum_{k=0}^{+\infty} \frac{1}{k!} A^k$ est l'exponentielle de la matrice A .

Déduire de la question précédente l'exponentielle de la matrice A définie en **Q4**.

On pourra utiliser sans démonstration que si M et N sont deux matrices de $M_n(\mathbb{K})$ qui commutent, $\exp(M + N) = (\exp M)(\exp N)$.

Q6. Soit $A \in M_n(\mathbb{K})$ telle que $A^2(A - I_n) = 0$.

Justifier que le polynôme $X(X - 1)$ est annulateur de la matrice A^2 .

Démontrer que le couple (D, N) de la décomposition de Dunford de la matrice A est donné par : $D = A^2$ et $N = A - A^2$.

Partie II - Un exemple par deux méthodes

Soit la matrice $A = \begin{pmatrix} 3 & -1 & 1 \\ 2 & 0 & 1 \\ 1 & -1 & 2 \end{pmatrix}$.

On note u l'endomorphisme de $\mathbb{R}^3$ canoniquement associé à la matrice A .

On notera id l'application identité de $\mathbb{R}^3$.

Q7. La matrice A est-elle diagonalisable dans $M_3(\mathbb{R})$?

Démontrer qu'on a la somme directe : $\mathbb{R}^3 = \ker(u - \text{id}) \oplus \ker(u - 2\text{id})^2$.

Q8. Déterminer une base (e_1, e_2, e_3) de $\mathbb{R}^3$ telle que :

$\ker(u - \text{id}) = \text{vect}\{e_1\}$, $\ker(u - 2\text{id}) = \text{vect}\{e_2\}$ et $\ker(u - 2\text{id})^2 = \text{vect}\{e_2, e_3\}$.

Écrire la matrice B de u dans la base (e_1, e_2, e_3) de $\mathbb{R}^3$.

Q9. Déterminer le couple de la décomposition de Dunford de la matrice B et en déduire le couple (on calculera ces matrices) de la décomposition de Dunford de la matrice A .

Q10. Décomposer en éléments simples la fraction $\frac{1}{(X-1)(X-2)^2}$ et en déduire deux polynômes

U et V tels que :

$$(X-1)U(X) + (X-2)^2V(X) = 1 \text{ avec } \deg U < 2 \text{ et } \deg V < 1.$$

Q11. On pose les endomorphismes : $p = V(u) \circ (u - 2\text{id})^2$ et $q = U(u) \circ (u - \text{id})$.

Calculer $p(x) + q(x)$ pour tout x vecteur de $\mathbb{R}^3$.

Démontrer que p est le projecteur sur $\ker(u - \text{id})$ parallèlement à $\ker(u - 2\text{id})^2$ et q est le projecteur sur $\ker(u - 2\text{id})^2$ parallèlement à $\ker(u - \text{id})$.

- Q12.** On pose $d = p + 2q$. Écrire la matrice de d dans la base (e_1, e_2, e_3) de $\mathbb{R}^3$ (de la question Q8).
Déterminer le couple de la décomposition de Dunford de la matrice A en exprimant D et N comme polynômes de la matrice A (sous forme développée).

Partie III - Une preuve de l'unicité de la décomposition

- Q13.** Soit E un K -espace vectoriel de dimension n .
Soient u et v deux endomorphismes diagonalisables de E qui commutent. On note $\lambda_1, \lambda_2, \dots, \lambda_p$ les valeurs propres de u et pour tout $1 \leq i \leq p$, $E_{\lambda_i}(u)$ le sous-espace propre de u associé à la valeur propre λ_i .
Démontrer que tout sous-espace propre de u est stable par v .
En déduire qu'il existe une base commune de diagonalisation pour u et v .
Pour tout $1 \leq i \leq p$, on pourra noter v_i l'endomorphisme induit par v sur $E_{\lambda_i}(u)$.
- Q14.** Soient A et B deux matrices diagonalisables de $M_n(K)$ qui commutent. Démontrer que la matrice $A - B$ est diagonalisable.
- Q15.** Soient A et B deux matrices nilpotentes de $M_n(K)$ qui commutent, démontrer que la matrice $A - B$ est nilpotente.
- Q16.** Déterminer les matrices de $M_n(K)$ qui sont à la fois diagonalisables et nilpotentes.
- Q17.** Dans cette question, on admet, pour toute matrice carrée A de $M_n(K)$ à polynôme caractéristique scindé, l'existence d'un couple (D, N) vérifiant les conditions (1), (2), (3), (4) et tel que D et N soient des polynômes en A .
Établir l'unicité du couple (D, N) dans la décomposition de Dunford.

Partie IV - Non continuité de l'application $A \mapsto D$

- Q18.** On note $\mathcal{D}$ l'ensemble des matrices de $M_n(\mathbb{C})$ qui sont diagonalisables.
 $\mathcal{D}$ est-il un espace vectoriel ?
Si P est une matrice inversible de $M_n(\mathbb{C})$, justifier que l'application de $M_n(\mathbb{C})$ vers $M_n(\mathbb{C})$, $M \mapsto PMP^{-1}$ est continue.
- Q19.** Démontrer que $\mathcal{D}$ est dense dans $M_n(\mathbb{C})$.
- Q20.** Si (D, N) est le couple de la décomposition de Dunford d'une matrice A , on note φ l'application de $M_n(\mathbb{C})$ dans $\mathcal{D}$ qui à la matrice A associe la matrice D .
Justifier que φ est l'application identité sur $\mathcal{D}$ et en déduire que l'application φ n'est pas continue.

FIN

4/4

SOLUTION

Q1. Rappelons que

$$\langle A | B \rangle = \text{tr}(A^T B) = \sum_{i=1}^n (A^T B)_{i,i} = \sum_{i=1}^n \sum_{k=1}^n (A^T)_{i,k} B_{k,i} = \sum_{i=1}^n \sum_{k=1}^n A_{k,i} B_{k,i}.$$

Autrement dit, le produit scalaire de deux matrices est la somme des produits des coefficients de ces deux matrices aux mêmes positions. En outre, les matrices diagonales sont engendrées par les matrices élémentaires $(E_{i,i})_{1 \leq i \leq n}$ car

$$\begin{pmatrix} d_1 & & (0) \\ & d_2 & \\ & & \ddots \\ (0) & & & d_n \end{pmatrix} = d_1 \begin{pmatrix} 1 & & (0) \\ & 0 & \\ & & \ddots \\ (0) & & & 0 \end{pmatrix} + d_2 \begin{pmatrix} 0 & & (0) \\ & 1 & \\ & & \ddots \\ (0) & & & (0) \end{pmatrix} \\ + \cdots + d_n \begin{pmatrix} 0 & & (0) \\ & & \ddots \\ & & & 0 \\ (0) & & & & 1 \end{pmatrix}.$$

Par conséquent, une matrice $M = (M_{i,j})_{1 \leq i,j \leq n}$ de $\mathfrak{M}_n(\mathbb{R})$ appartient à l'orthogonal de $\mathcal{D}_n(\mathbb{R})$ si et seulement si elle est orthogonale à chacune des matrices $E_{j,j}$, $1 \leq j \leq n$. Comme

$$\langle M | E_{j,j} \rangle = \sum_{i=1}^n \sum_{k=1}^n M_{k,i} \underbrace{(E_{j,j})_{k,i}}_{=0 \text{ si } (k,i) \neq (j,j)} = M_{j,j} (E_{j,j})_{j,j} = M_{j,j},$$

la matrice M est orthogonale à $\mathcal{D}_n(\mathbb{R})$ si et seulement si tous ses coefficients diagonaux sont nuls.

Commentaires : Question difficile car elle est ouverte et demande autonomie, initiative et capacité à mettre en place une stratégie relativement complexe pour ce concours. La première difficulté rencontrée par les candidats est de connaître l'expression du produit scalaire à l'aide des coefficients des matrices. Ensuite, pensez à fixer une base de $\mathcal{D}_n(\mathbb{R})$ pour conclure. Au final, les réponses convenables se trouvent dans les très bonnes copies.

Q2. Si A est diagonalisable alors $D = A$ et $N = 0$ conviennent (car D diagonalisable, N nilpotente, $A = D + N$ et $DN = 0 = ND$). Si A est nilpotente alors $D = 0$ et $A = N$ conviennent (car D diagonalisable, N nilpotente, $A = D + N$ et $DN = 0 = ND$).

Si A est trigonalisable, il existe une matrice inversible P et une matrice diagonale par blocs

$$T = \begin{pmatrix} \lambda_1 I_{r_1} + N_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_m I_{r_m} + N_m \end{pmatrix}$$

avec chaque N_i triangulaire stricte, donc nilpotente, telle que $A = PTP^{-1}$. Posons

$$D' = \begin{pmatrix} \lambda_1 I_{r_1} & & (0) \\ & \ddots & \\ (0) & & \lambda_m I_{r_m} \end{pmatrix}, N' = \begin{pmatrix} N_1 & & (0) \\ & \ddots & \\ (0) & & N_m \end{pmatrix}$$

alors $T = D' + N'$. Chaque matrice N_i étant nilpotente (d'indice au plus n car elles sont toutes de taille au plus n), N' est aussi nilpotente car, par produit matriciel par blocs, on a :

$$(N')^n = \begin{pmatrix} (N_1)^n & & (0) \\ & \ddots & \\ (0) & & (N_m)^n \end{pmatrix} = \begin{pmatrix} 0 & & (0) \\ & \ddots & \\ (0) & & 0 \end{pmatrix} = 0_n.$$

En outre, toujours par calcul matriciel par blocs, on a :

$$D'N' = \begin{pmatrix} \lambda_1 I_{r_1} N_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_m I_{r_m} N_m \end{pmatrix} = \begin{pmatrix} \lambda_1 N_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_m N_m \end{pmatrix} = N'D'.$$

On peut écrire

$$A = P(D' + N')P^{-1} = PD'P^{-1} + PN'P^{-1}.$$

Si on pose $D = PD'P^{-1}$ qui est une matrice diagonalisable (par définition !) et $N = PN'P^{-1}$ qui est nilpotente car

$$N^n = P(N')^n P^{-1} = P0P^{-1} = 0$$

alors on a $A = D + N$ avec

$$DN = PD'P^{-1}PN'P^{-1} = PD'N'P^{-1} = PN'D'P^{-1} = PN'P^{-1}PD'P^{-1} = ND.$$

Par conséquent, (D, N) est une décomposition de Dunford de A .

$\left(\begin{pmatrix} 1 & 0 \\ 0 & 2 \end{pmatrix}, \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}\right)$ n'est pas une décomposition de Dunford car ces deux matrices ne commutent pas puisque :

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

Commentaires : Question élémentaire dans le cas diagonalisable et trigonalisable (il est attendu d'explicitier le couple (D, N) quand même !).

Pour le cas trigonalisable, la question s'avère plus sélective (normal car il faut connaître la bonne version du théorème de trigonalisation via les blocs avec les mêmes valeurs propres).

Pour l'exemple, c'est une question facile mais discriminante car elle demande un peu de rigueur de la part du candidat.

Q3. Il suffit de prendre une matrice A dont le polynôme caractéristique n'est pas scindé (si (D, N) est une décomposition de Dunford de A , $\chi_A = \chi_D$ est scindé). Par exemple, la matrice de rotation d'angle $\frac{\pi}{2}$, $A = \begin{pmatrix} 0 & -1 \\ 1 & 0 \end{pmatrix}$ dont le polynôme caractéristique est $\chi_A = X^2 + 1$.

Commentaires : Question de contre-exemple donc elle s'avère en pratique très sélective (cela demande une autre qualité que de connaître simplement son cours ou des techniques).

Q4. $A = \begin{pmatrix} 3 & 0 & 8 \\ 3 & -1 & 6 \\ -2 & 0 & -5 \end{pmatrix}$ alors

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

(en développant selon la deuxième colonne)

$$\begin{aligned} \chi_A(X) &= (X+1)[(X-3)(X+5)+16] \\ &= (X+1)[X^2+2X+1] = (X+1)^3 \end{aligned}$$

Première réponse : Le théorème de Cayley-Hamilton affirme que χ_A annule A donc

$$\chi_A(A) = 0 \Leftrightarrow (A+I)^3 = 0$$

donc $N = A + I$ est nilpotente et $D = -I$ est diagonalisable, $A = D + N$ avec les matrices D et N qui commutent donc $(-I, A+I)$ est une décomposition de Dunford de A .

Deuxième réponse. D'après Comme χ_A est scindé, A est trigonalisable et son unique valeur propre est -1 (unique racine de χ_A). Il existe P inversible et T triangulaire (avec uniquement des 1 sur la diagonale i.e. $T = \begin{pmatrix} -1 & a & b \\ 0 & -1 & c \\ 0 & 0 & -1 \end{pmatrix}$) telle que $A = PTP^{-1}$. Si on pose

$$D' = -I_3 \text{ et } N' = \begin{pmatrix} 0 & a & b \\ 0 & 0 & c \\ 0 & 0 & 0 \end{pmatrix}$$

alors $T = D' + N'$, D' et N' commutent. On a

$$A = P(D' + N')P^{-1} = PD'P^{-1} + PN'P^{-1}.$$

Si on pose

$$\begin{aligned} D &= PD'P^{-1} = P(-I)P^{-1} = -I \text{ et} \\ N &= PN'P^{-1} = P(T+I)P^{-1} = PTP^{-1} + PIP^{-1} = A + I_3, \end{aligned}$$

alors D et N commutent, D est diagonalisable, N est nilpotente (car N' l'est) et $A = D + N$ donc $(-I, A+I)$ est une décomposition de Dunford de A .

Commentaires : Question de difficulté standard qui s'avère discriminante (elle demande une autonomie et une technicité de la part du candidat). La détermination du couple (D, N) ne pose aucune difficulté aux candidats ayant réussi la trigonalisation.

Q5. Comme $A = D + N$ avec $D = I$ et $N = A - I$ qui commutent, on a :

$$\begin{aligned} e^A &= e^{D+N} = e^D e^N = \sum_{k=0}^{+\infty} \frac{(-I)^k}{k!} \sum_{k=0}^{+\infty} \underbrace{\frac{N^k}{k!}}_{=0 \text{ si } k \geq 3} = \sum_{k=0}^{+\infty} \frac{(-1)^k}{k!} I \left(I + N + \frac{N^2}{2} \right) \\ &= e^{-1} \left(I + N + \frac{N^2}{2} \right) = e \left(A + \frac{(A+I)^2}{2} \right). \end{aligned}$$

Commentaires : Question élémentaire (ne pas oublier de vérifier la commutation de D et N).

Q6. Si on pose $P = X(X - 1)$ alors

$$P(A^2) = A^2(A^2 - I) = A^2(A - I)(A + I) = 0(A + I) = 0$$

donc P annule bien A^2 . Le polynôme P étant scindé à racines simples et annulant A^2 , on peut affirmer que A^2 est diagonalisable donc $D = A^2$ l'est. La matrice

$$N = A - A^2 = A(I - A)$$

est nilpotente car, comme A et $I - A$ commutent, on peut écrire :

$$N^2 = A^2(I - A)^2 = A^2(A - I)(A - I) = 0(A - I) = 0.$$

Les matrices D et N commutent (ceux sont deux polynômes en A) et $A = D + N$ donc (D, N) est une décomposition de Dunford de A .

Commentaires : La première question est sans difficulté. Pour la seconde, la diagonalisation de A^2 s'avère sélective (penser au polynôme annulateur scindé à racines simples), le reste est aisé.

Q7. Calculons le polynôme caractéristique de A .

$$\begin{aligned} \chi_A(X) &= \det(XI_3 - A) = \begin{vmatrix} X-3 & 1 & -1 \\ -2 & X & -1 \\ -1 & 1 & X-2 \end{vmatrix} \stackrel{C_3 \leftarrow C_3 + C_2}{=} \begin{vmatrix} X-3 & 1 & 0 \\ -2 & X & X-1 \\ -1 & 1 & X-1 \end{vmatrix} \\ &= (X-1) \begin{vmatrix} X-3 & 1 & 0 \\ -2 & X & 1 \\ -1 & 1 & 1 \end{vmatrix} \stackrel{L_2 \leftarrow L_2 - L_3}{=} (X-1) \begin{vmatrix} X-3 & 1 & 0 \\ -1 & X-1 & 0 \\ -1 & 1 & 1 \end{vmatrix} \end{aligned}$$

En développant selon la dernière colonne, on en déduit que

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

Ainsi, les valeurs de A sont les racines de son polynôme caractéristique c'est-à-dire sont 1 et 2. Déterminons la dimension de l'espace propre associé à 2 via le rang de la matrice

$$A - 2I = \begin{pmatrix} 1 & -1 & 1 \\ 2 & -2 & 1 \\ 1 & -1 & 0 \end{pmatrix}.$$

Par définition, elle n'est pas inversible (car 2 est valeur propre de A) donc son rang est strictement inférieur à 3. Sa première et sa troisième colonne ne sont pas colinéaires donc son rang vaut au moins 2. Par conséquent, son rang vaut 2 donc, d'après le théorème du rang, on peut affirmer que

$$\dim(\ker(A - 2I)) = 3 - \text{rg}(A - 2I) = 1 \Leftrightarrow \dim(E_2(A)) = 1.$$

Ainsi, l'espace propre associé à 2 est de dimension 1 qui est différent de la multiplicité de 2 comme racine de χ_A donc A n'est pas diagonalisable.

Les polynômes $X - 1$ et $(X - 2)^2$ sont premiers entre eux (car scindés et sans racines communes). Comme $\chi_A = (X - 1)(X - 2)^2$, le lemme des noyaux permet d'écrire :

$$\ker(\chi_A(A)) = \ker(A - I) \oplus \ker(A - 2I)^2.$$

Comme $\chi_A(A) = 0$ (par le théorème de Cayley-Hamilton, le polynôme caractéristique de A annule A) et que la matrice nulle (de taille 3) a pour noyau $\mathfrak{M}_{3,1}(\mathbb{R})$ donc

$$\mathfrak{M}_{3,1}(\mathbb{R}) = \ker(A - I) \oplus \ker(A - 2I)^2.$$

En passant cette formule aux endomorphismes, on obtient la formule attendue.

Commentaires : Les deux questions sont des applications directes et rudimentaires du cours. Pourtant, elles s'avèrent toutes deux très discriminantes. Si l'une de ces questions vous a posé problème, il faut les retravailler et consolider vos connaissances associées.

Q8. Déterminons ces noyaux pour la matrice A , la conversion étant alors immédiate à u . Notons $(\varepsilon_1, \varepsilon_2, \varepsilon_3)$ la base canonique de $\mathbb{R}^3$.

Base de $\ker(u - \text{Id})$. Soient C_1, C_2, C_3 les colonnes de $A - I$ et $v = u - \text{Id}$ l'endomorphisme canoniquement associé à $A - I$. On a :

$$\ker(A - I) = \ker \begin{pmatrix} 2 & -1 & 1 \\ 2 & -1 & 1 \\ 1 & -1 & 1 \end{pmatrix}$$

Comme $A - I$ est de rang 2 (deux premières colonnes non colinéaires donc libres et elle n'est pas inversible car 1 est valeur propre de A), son noyau est de dimension $3 - 2 = 1$ (par le théorème du rang). Comme $C_2 = -C_3$ et que C_i sont les coordonnées de $v(\varepsilon_i)$, on peut affirmer que

$$\begin{aligned} v(\varepsilon_2) &= -v(\varepsilon_3) \Leftrightarrow v(\varepsilon_2) + v(\varepsilon_3) = 0 \Leftrightarrow v(\varepsilon_2 + \varepsilon_3) = 0 \\ &\Leftrightarrow \varepsilon_2 + \varepsilon_3 \in \ker(v) = \ker(u - \text{Id}) \end{aligned}$$

donc

$$e_1 = \varepsilon_2 + \varepsilon_3 = (0, 1, 1)$$

est une base de $\ker(u - \text{Id})$.

Soit C_1, C_2, C_3 les colonnes de $A - 2I$ et $v = u - 2\text{Id}$ l'endomorphisme canoniquement associé à $A - 2I$. On a :

$$\ker(A - 2I) = \ker \begin{pmatrix} 1 & -1 & 1 \\ 2 & -2 & 1 \\ 1 & -1 & 0 \end{pmatrix}$$

Comme $A - 2I$ est de rang 2 (première et troisième colonnes non colinéaires donc libres et elle n'est pas inversible car 2 est valeur propre de A), son noyau est de dimension $3 - 2 = 1$ (par le théorème du rang). Comme $C_1 = -C_2$ et que C_i sont les coordonnées de $v(\varepsilon_i)$, on peut affirmer que

$$\begin{aligned} v(\varepsilon_1) &= -v(\varepsilon_2) \Leftrightarrow v(\varepsilon_1) + v(\varepsilon_2) = 0 \Leftrightarrow v(\varepsilon_1 + \varepsilon_2) = 0 \\ &\Leftrightarrow \varepsilon_1 + \varepsilon_2 \in \ker(v) = \ker(u - 2\text{Id}) \end{aligned}$$

donc

$$e_2 = \varepsilon_1 + \varepsilon_2 = (1, 1, 0)$$

est une base de $\ker(u - 2\text{Id})$.

Soit C_1, C_2, C_3 les colonnes de $(A - 2I)^2$ et $v = (u - 2\text{Id})^2$ l'endomorphisme canoniquement associé à $(A - 2I)^2$. On a :

$$\ker(A - 2I)^2 = \ker \begin{pmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ -1 & 1 & 0 \end{pmatrix}$$

Comme $(A - 2I)^2$ est de rang 1 (première colonne non nul donc libre, les deux autres sont combinaisons linéaires de la première), son noyau est de dimension $3 - 1 = 2$ (par le théorème du rang). On peut affirmer que

$$e_2 = (1, 1, 0) \in \ker(A - 2I) \subset \ker(A - 2I)^2.$$

Comme $C_3 = 0$ et que C_i sont les coordonnées de $v(\varepsilon_i)$, on peut affirmer que

$$v(\varepsilon_3) = 0 \Leftrightarrow \varepsilon_3 \in \ker(v) = \ker(u - 2\text{Id})^2$$

donc $e_3 = (0, 0, 1)$ appartient à $\ker(u - 2\text{Id})^2$. Puisque les vecteurs e_2 et e_3 sont non colinéaires donc libres et qu'ils appartiennent à $\ker(u - 2\text{Id})^2$ qui est un espace vectoriel de dimension 2, on peut affirmer qu'il s'agit d'une base de $\ker(u - 2\text{Id})^2$.

Puisque

$$A \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \\ 2 \end{pmatrix} = e_2 + 2e_3,$$

on peut écrire :

$$u(e_1) = e_1, u(e_2) = 2e_2, u(e_3) = e_2 + 2e_3.$$

Ainsi, la matrice B est définie par :

$$B = \begin{pmatrix} u(e_1) & u(e_2) & u(e_3) \\ 1 & 0 & 0 \\ 0 & 2 & 1 \\ 0 & 0 & 2 \end{pmatrix} \begin{pmatrix} e_1 \\ e_2 \\ e_3 \end{pmatrix}$$

Commentaires : La détermination des bases est en pratique assez bien traitée par les candidats. Par contre, la détermination de B s'avère sélective à ce concours (elle demande une bonne compréhension de la notion de matrice dans une base donnée).

Q9. On pose

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

alors $B = D' + N'$, D' est diagonalisable, N' nilpotente, D' et N' commutent donc (D', N') est une décomposition de Dunford de B . Si on pose P la matrice de passage de la base canonique de $\mathbb{R}^3$ à (e_1, e_2, e_3) , c'est-à-dire

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

alors $D = PD'P^{-1}$ et $N = PN'P^{-1}$ forment une décomposition de Dunford de A .

Commentaires : La question est sélective car elle demande d'avoir une réponse convenable à la question précédente.

Q10. Il existe trois réels a, b, c tels que

$$(*) : \frac{1}{(X-1)(X-2)^2} = \frac{a}{X-1} + \frac{b}{X-2} + \frac{c}{(X-2)^2}.$$

En multipliant cette égalité par $X-1$ (respectivement $(X-2)^2$), on obtient

$$\begin{aligned} \frac{1}{(X-2)^2} &= a + (X-1) \left(\frac{b}{X-2} + \frac{c}{(X-2)^2} \right) \\ (\text{resp. } \frac{1}{X-1} &= c + (X-2)^2 \left(\frac{a}{X-1} + \frac{b}{X-2} \right)). \end{aligned}$$

On fait alors tendre X vers 1 (resp. vers 2), ce qui nous donne $a = 1$ et $c = 1$. En multipliant l'égalité par X puis en faisant tendre X vers $+\infty$, on obtient $0 = a + b$ donc $b = -1$ ce qui nous donne

$$\frac{1}{(X-1)(X-2)^2} = \frac{1}{X-1} - \frac{1}{X-2} + \frac{1}{(X-2)^2}.$$

En multipliant cette égalité par $(X-1)(X-2)^2$, on obtient l'égalité

$$\begin{aligned} 1 &= (X-2)^2 + (X-1)[- (X-2) + 1] \\ &= (X-2)^2 + (X-1)(3-X) \end{aligned}$$

Ainsi, $U = 3 - X$ et $V = 1$ conviennent.

Commentaires : La première question est une question d'application directe du cours. Elle s'avère néanmoins discriminante donc à retravailler si vous ne l'avez pas traitée correctement. Pour la seconde, elle est sélective (il avoir réussi la première et avoir un peu de recul).

Q11. Par construction, on a

$$\begin{aligned} p + q &= V(u) \circ (u - 2\text{Id})^2 + U(u) \circ (u - \text{Id}) \\ &= \left(V(X-2)^2 + U(X-1) \right) (u) = 1(u) = \text{Id} \end{aligned}$$

donc $p(x) + q(x) = x$. Si $x \in \ker(u - 2\text{Id})^2$ alors

$$(u - 2\text{Id})^2(x) = 0 \Rightarrow p(x) = V(u) \left((u - 2\text{Id})^2(x) \right) = V(u)(0_E) = 0_E.$$

Si $x \in \ker(u - \text{Id})$ alors

$$(u - \text{Id})(x) = 0 \Rightarrow q(x) = U(u) \left((u - \text{Id})(x) \right) = U(u)(0_E) = 0_E$$

donc $p(x) = x - q(x) = x$. Par conséquent, q est bien le projecteur sur $\ker(u - \text{Id})$ parallèlement à $\ker(u - 2\text{Id})^2$ donc $p = \text{Id} - q$ est son projecteur complémentaire sur $\ker(u - 2\text{Id})^2$ parallèlement à $\ker(u - \text{Id})$.

Commentaires : La première question est élémentaire. Pour la seconde, on peut démontrer directement que p est un projecteur en calculant p^2 . C'est lourd et il faudra néanmoins déterminer son noyau et son image sachant que la réponse est donnée par le sujet. Au final, la seconde question demande une bonne compréhension de l'algèbre linéaire par le candidat.

Q12. Par définition, on a :

$$\begin{aligned} p &= V(u) \circ (u - 2 \text{id})^2 = \text{Id} \circ (u - 2 \text{Id})^2 = (u - 2 \text{Id})^2 \\ q &= \text{Id} - p \\ d &= p + 2q = p + 2(\text{Id} - p) = 2\text{Id} - p. \end{aligned}$$

D'après la question 8, la matrice de u dans la base (e_1, e_2, e_3) est

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

donc la matrice de p dans cette base est

$$C = (B - 2\text{Id})^2 = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{pmatrix}^2 = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

donc la matrice de d dans la même base est

$$2I_3 - C = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix} = D'$$

(de la question 8).

D'après la question 8, $(D', N' = B - D')$ est la décomposition de Dunford de B donc $(d, u - d)$ est celle de u . Comme $d = 2\text{Id} - (u - 2\text{Id})^2$, la décomposition de Dunford de A est

$$\left(2I - (A - 2I)^2, A - \left(2I - (A - 2I)^2 \right) \right) = (-A^2 + 4A - 2I, A^2 - 3A + 2I).$$

Commentaires : Question à destination des meilleures copies et très difficile pour ce concours.

Q13. Soient λ une valeur propre de u et $x \in E_\lambda(u)$, on a :

$$\begin{aligned} u(x) &= \lambda x \Rightarrow v(u(x)) = \lambda v(x) \Leftrightarrow (v \circ u)(x) = \lambda v(x) \\ &\Leftrightarrow (u \circ v)(x) = \lambda v(x) \Leftrightarrow u(v(x)) = \lambda v(x) \end{aligned}$$

donc $v(x) \in E_\lambda(u)$ c'est-à-dire que tout espace propre de u est stable par v .

Soit $E_\lambda(u)$ un espace propre de u . Comme v laisse stable cet espace, la restriction v_λ de v à cet espace est un endomorphisme de $E_\lambda(u)$. Comme v est diagonalisable, v_λ l'est aussi (théorème de cours) c'est-à-dire qu'il existe une base $\mathcal{B}_\lambda$ de $E_\lambda(u)$ formé de vecteurs propres pour v . Par définition, tout vecteur non nul de $E_\lambda(u)$ est un vecteur propre de u donc tous les éléments de $\mathcal{B}_\lambda$ sont à la fois vecteurs propres de u et de v . En concaténant toutes ces bases

$\mathcal{B}_\lambda$ (λ parcourant les valeurs propres de u), on obtient une base $\mathcal{B}$ de $\bigoplus_{\lambda \in \text{Sp}(u)} E_\lambda(u) = E$ (car u est diagonalisable) formée de vecteurs propres à la fois de u et v c'est-à-dire qu'il existe bien une base de E diagonalisant à la fois u et v .

Commentaires : La première question est une question de cours. La seconde est un raisonnement classique (et probablement fait dans votre cours et / ou TD) mais elle reste à destination des bonnes copies en pratique (il demande une solidité certaine concernant la diagonalisation).

Q14. Notons a et b les endomorphismes canoniquement associés à A et B . Comme A et B commutent, a et b commutent également et comme A et B sont diagonalisables alors a et b sont aussi diagonalisables. D'après la question précédente, il existe une base $\mathcal{B} = (e_1, \dots, e_n)$ de diagonalisation commune à a et b . Les matrices D_a et D_b dans cette base sont diagonales et comme la matrice de $a - b$ dans cette base est $D_a - D_b$ qui est diagonale, on peut affirmer que $a - b$ est diagonalisable donc $A - B$ aussi.

Commentaires : Question élémentaire. Si elle vous a posé problème, il faut consolider vos connaissances sur la réduction (en priorité, les définitions initiales!).

Q15. Comme A et B sont nilpotentes et de taille n , on peut affirmer que $A^n = 0_n = B^n$. Puisque les matrices A et B commutent, la formule du binôme de Newton nous donne :

$$(A - B)^{2n} = \sum_{k=0}^{2n} \binom{2n}{k} A^k (-B)^{2n-k} = \sum_{k=0}^{2n} (-1)^{2n-k} \binom{2n}{k} A^k B^{2n-k}.$$

Si $k \in \{n, \dots, 2n\}$ alors $A^k = 0_n$ donc $A^k B^{2n-k} = 0_n$. Si $k \in \{0, \dots, n\}$ alors $2n - k \in \{n, \dots, 2n\}$ donc $B^{2n-k} = 0_n$ d'où $A^k B^{2n-k} = 0_n$. Autrement dit, tous les termes de la somme sont nuls donc $(A - B)^{2n} = 0_n$ ce qui prouve que $A - B$ est nilpotente.

Commentaires : Question de niveau standard et classique. La commutativité doit impérativement être mentionnée sous peine d'être lourdement sanctionnée.

Q16. Si A est diagonalisable, il existe une matrice inversible P et une matrice diagonale $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ telles que $A = PDP^{-1}$. Si en outre, A est nilpotente, on a

$$\begin{aligned} A^n &= 0_n \Leftrightarrow PD^nP^{-1} = 0_n \Leftrightarrow D^n = P^{-1}0_nP = 0_n \Leftrightarrow \text{diag}(\lambda_1^n, \dots, \lambda_n^n) = 0_n \\ &\Leftrightarrow \forall i \in \{1, \dots, n\}, \lambda_i^n = 0 \Leftrightarrow \forall i \in \{1, \dots, n\}, \lambda_i = 0 \Leftrightarrow D = 0 \end{aligned}$$

donc $A = PDP^{-1} = P0_nP^{-1} = 0_n$. Ainsi, seule la matrice nulle est à la fois diagonalisable et nilpotente.

Commentaires : Question élémentaire ... si le cours de diagonalisation est connu notamment la définition de celle-ci ! Si cela n'est pas votre cas, retravailler les fondamentaux du chapitre de réduction.

Q17. Supposons qu'il existe deux décompositions de Dunford (D, N) et (D', N') de A . On a :

$$D + N = A = D' + N' \Rightarrow D - D' = N' - N.$$

Comme D et D' sont des polynômes en A , elles commutent. En outre, elles sont chacune diagonalisables donc, d'après la question 14, la matrice $D - D'$ est diagonalisable.

Comme N et N' sont des polynômes en A , elles commutent. En outre, elles sont chacune nilpotentes donc, d'après la question 15, la matrice $N' - N$ est diagonalisable.

Par conséquent, la matrice $D - D'$ est à la fois diagonalisable et nilpotente (car égale à $N' - N$) donc elle est nulle d'où $D = D'$. Il est alors immédiat que

$$N' - N = D - D' = 0_n \Rightarrow N' = N$$

d'où l'unicité de la décomposition de Dunford.

Commentaires : Question sélective destinée aux bonnes copies (sans être excellentes), notamment pour les candidat(e)s ayant une maîtrise convenable de la réduction.

Q18. $A = \begin{pmatrix} -1 & 1 \\ 0 & 1 \end{pmatrix}$ et $B = \begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix}$ sont diagonalisables. En effet, elles sont triangulaires donc leurs valeurs propres sont leurs coefficients diagonaux c'est-à-dire 1 et -1 donc elles possèdent toutes deux 2 valeurs propres distinctes et elles sont de taille 2. Par contre, la matrice $A + B = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$ ne l'est pas car 0 est son unique valeur propre et si elle est diagonalisable, il existe P inversible telle que $A + B = P \operatorname{diag}(0, 0) P^{-1} = 0$, ce qui est absurde. Par conséquent, $\mathcal{D}$ n'est pas un espace vectoriel. L'application $f : M \in \mathfrak{M}_n(\mathbb{C}) \mapsto PMP^{-1}$ est linéaire en dimension finie donc elle est continue.

Commentaires : La première question étant une question de contre-exemple, elle s'avère difficile. La seconde est élémentaire en utilisant le chapitre d'espaces vectoriels normés. Si ce n'est pas le cas, retravailler la continuité des applications (dont la dimension finie et dont les applications linéaires ou polynomiales).

Q19. Soit $M \in \mathfrak{M}_n(\mathbb{C})$, elle est trigonalisable (son polynôme caractéristique χ_M est non constant et à coefficients dans $\mathbb{C}$ donc scindé par le théorème de D'Alembert-Gauss). Il existe donc des complexes $\lambda_1, \dots, \lambda_m$ et des matrices triangulaires strictes $N_1, \dots, N_m$ ainsi qu'une matrice inversible P telles que

$$A = P \underbrace{\begin{pmatrix} \lambda_1 I_{r_1} + N_1 & & (0) \\ & \ddots & \\ (0) & & \lambda_m I_{r_m} + N_m \end{pmatrix}}_{=T} P^{-1} = PTP^{-1}.$$

Pour chaque $i \in \{1, \dots, m\}$, on pose

$$T_i = \lambda_i I_{r_i} + N_i = \begin{pmatrix} \lambda_i & & (*) \\ & \ddots & \\ (0) & & \lambda_i \end{pmatrix}.$$

Nous allons approcher cette matrice par une matrice diagonalisable (en modifiant légèrement ses valeurs propres pour qu'elles deviennent deux à deux distinctes). Pour chaque entier

$k \geq 1$, on pose

$$D_{i,k} = \text{diag} \left(\lambda_i, \lambda_i + \frac{1}{k}, \lambda_i + \frac{2}{k}, \dots, \lambda_i + \frac{r_i}{k} \right)$$

$$T_{i,k} = D_{i,k} + N_i = \begin{pmatrix} \lambda_i & & & & \\ & \lambda_i + 1/k & & & \\ & & \lambda_i + 2/k & & \\ & & & \ddots & \\ (0) & & & & \lambda_i + r_i/k \end{pmatrix} \quad (*)$$

$$D_k = \text{diag}(D_{1,k}, \dots, D_{m,k}), \quad T_k = \text{diag}(T_{1,k}, \dots, T_{m,k}) \quad \text{et} \quad A_k = P T_k P^{-1}.$$

La suite $(D_{i,k})_k$ converge vers $\lambda_i I_{r_i}$ donc la suite $(T_{i,k})_k$ converge vers $\lambda_i I_{r_i} + N_i = T_i$. Par conséquent, la suite (T_k) converge vers T donc, par continuité de l'application $M \mapsto P M P^{-1}$ (cf. question précédente), la suite A_k converge vers $P T P^{-1} = A$.

Les valeurs propres de $T_{i,j}$ sont ses coefficients diagonaux (car elle est triangulaire) et elles sont deux à deux distinctes donc $T_{i,k}$ est diagonalisable. Ainsi, il existe une matrice inversible $Q_{i,k}$ tel que

$$T_{i,k} = Q_{i,k} D_{i,k} Q_{i,k}^{-1}.$$

On pose pour chaque entier $k \geq 1$,

$$Q_k = \text{diag}(Q_{1,k}, \dots, Q_{m,k}), \quad Q'_k = \text{diag}(Q_{1,k}^{-1}, \dots, Q_{m,k}^{-1})$$

$$D_k = \text{diag}(D_{1,k}, \dots, D_{m,k}) \quad \text{et} \quad T_k = \text{diag}(T_{1,k}, \dots, T_{m,k}).$$

alors Q_k est inversible d'inverse $Q'_{i,k}$ car, par le calcul matriciel par blocs

$$Q_k Q'_k = \text{diag}(Q_{1,k} Q_{1,k}^{-1}, \dots, Q_{m,k} Q_{m,k}^{-1}) = \text{diag}(I, \dots, I) = I_n.$$

La matrice D_k est diagonale et, par le calcul matriciel par blocs, on a pour tout entier $k \geq 1$:

$$T_k = Q_k D_k Q_k^{-1}$$

donc T_k est diagonalisable ce qui entraîne la diagonalisabilité de A_k . Par conséquent, $\mathcal{D}$ est dense dans $\mathfrak{M}_n(\mathbb{C})$.

Commentaires : Question très difficile pour ce concours (et difficile pour les concours Mines-Ponts et CentraleSupElec) donc elle n'est accessible qu'aux meilleures copies.

Q20. Si A est diagonalisable alors $(A, 0_n)$ est sa décomposition de Dunford donc $\varphi(A) = A$ pour tout $A \in \mathcal{D}$. Si φ est continue, comme l'application $\text{Id} : A \rightarrow A$ est également continue, ces deux applications sont continues et égales sur l'ensemble dense $\mathcal{D}$ donc elles sont égales partout. Autrement dit, pour toute matrice $A \in \mathfrak{M}_n(\mathbb{C})$, $\varphi(A) = A$ donc A est diagonalisable (car $\varphi(A)$ l'est par définition) ce qui est absurde (il existe des matrices non diagonalisables dans $\mathfrak{M}_n(\mathbb{C})$, par exemple toute matrice nilpotente non nulle dont les matrices triangulaires strictes et non nulles sont des exemples).

Commentaires : La première question est élémentaire ... mais elle nécessite que le candidat ait assimilé la notion de décomposition de Dunford ... et tente cette question ! Pour la seconde, elle s'adresse aux bonnes copies (aptées à utiliser les notions de topologie dont la continuité et la densité ainsi qu'une maîtrise assurée du chapitre de réduction).

ÉPREUVE MUTUALISÉE AVEC E3A-POLYTECH**ÉPREUVE SPÉCIFIQUE - FILIÈRE MP**

INFORMATIQUE**Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de trois parties indépendantes.

Partie I - Étude des partitions non croisées

Dans cette partie, on introduit et on étudie de façon élémentaire les partitions non croisées, objets combinatoires apparaissant dans divers domaines des mathématiques, notamment dans la théorie des probabilités libres et des matrices aléatoires.

Le langage de programmation utilisé dans cette partie est le langage Python.

Dans la suite, pour tout couple $(i, n) \in \mathbb{N}^2$, les notations $\llbracket n \rrbracket$ et $\llbracket i, n \rrbracket$ désignent respectivement les ensembles $\{1, 2, \dots, n\}$ et $\{i, i+1, \dots, n\}$. Par convention, $\llbracket 0 \rrbracket = \emptyset$ et si $i > n$ alors $\llbracket i, n \rrbracket = \emptyset$.

Définition 1 (Partition)

Soit A un sous-ensemble non vide de $\mathbb{N}$. Une **partition** $\mathcal{P}$ de A est un ensemble de parties de A tel que :

1. $\forall P \in \mathcal{P}, P \neq \emptyset$;
2. $\forall (P, Q) \in \mathcal{P}^2, (P \neq Q) \implies (P \cap Q = \emptyset)$;
3. $\bigcup_{P \in \mathcal{P}} P = A$.

Par exemple, $\{\{2, 4, 5\}, \{7, 9\}, \{8\}\}$ est une partition de l'ensemble $\{2, 4, 5, 7, 8, 9\}$.

Définition 2 (Classe)

Soit $\mathcal{P}$ une partition d'un sous-ensemble A non vide de $\mathbb{N}$. Soit i un élément de A . La **classe** de i est l'unique élément P de $\mathcal{P}$ tel que $i \in P$. Elle est notée $\text{Cl}(i)$.

Par exemple, pour $\mathcal{P} = \{\{2, 4, 5\}, \{7, 9\}, \{8\}\}$, on a $\text{Cl}(4) = \{2, 4, 5\}$.

Définition 3 (Partition non croisée)

Soit $\mathcal{P}$ une partition d'un sous-ensemble A non vide de $\mathbb{N}$. On dit que $\mathcal{P}$ est **non croisée** si pour tout quadruplet $(a, b, c, d) \in A^4$ tel que $a < b < c < d$ on a :

$$(\text{Cl}(a) = \text{Cl}(c), \text{Cl}(b) = \text{Cl}(d)) \implies (\text{Cl}(a) = \text{Cl}(b) = \text{Cl}(c) = \text{Cl}(d)).$$

Par exemple, $\mathcal{P} = \{\{2, 4, 5\}, \{7, 9\}, \{8\}\}$ est bien une partition non croisée de $\{2, 4, 5, 7, 8, 9\}$. En effet, aucun quadruplet $(a, b, c, d) \in \{2, 4, 5, 7, 8, 9\}^4$ tel que $a < b < c < d$ ne vérifie la condition $\text{Cl}(a) = \text{Cl}(c), \text{Cl}(b) = \text{Cl}(d)$.

De même, $\mathcal{Q} = \{\{2, 7, 8, 9\}, \{4, 5\}\}$ en est bien une. En effet, $(a, b, c, d) = (2, 7, 8, 9)$ est l'unique quadruplet tel que $a < b < c < d$ vérifiant $\text{Cl}(a) = \text{Cl}(c), \text{Cl}(b) = \text{Cl}(d)$ et ces quatre éléments sont bien dans la même classe.

Par contre, $\mathcal{R} = \{\{2, 5\}, \{4, 7, 9\}, \{8\}\}$ n'en est pas une. En effet, $\text{Cl}(2) = \text{Cl}(5)$ et $\text{Cl}(4) = \text{Cl}(7)$ mais $\text{Cl}(2) \neq \text{Cl}(4)$.

Le terme « non croisée » découle naturellement des représentations picturales des partitions (**figure 1**).

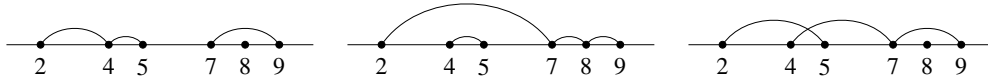


Figure 1 - Représentations picturales de $\mathcal{P}, \mathcal{Q}, \mathcal{R}$

Dans la suite, on s'intéresse uniquement aux partitions d'un sous-ensemble fini de $\mathbb{N}$. On les représente en Python à l'aide de listes de listes croissantes d'entiers triées dans l'ordre lexicographique.

Par exemple, les partitions $\mathcal{P} = \{\{2, 4, 5\}, \{7, 9\}, \{8\}\}$ et $\mathcal{R} = \{\{2, 5\}, \{4, 7, 9\}, \{8\}\}$ sont respectivement représentées par les listes $P = [[2, 4, 5], [7, 9], [8]]$ et $R = [[2, 5], [4, 7, 9], [8]]$.

I.1 - Exemples et fonctions élémentaires (Informatique Pour Tous)

Q1. Justifier brièvement que $\mathcal{P} = \{\{1, 7\}, \{2\}, \{3, 4, 5\}, \{6\}\}$ est une partition non croisée de $\llbracket 7 \rrbracket$ et que $\mathcal{Q} = \{\{1, 6\}, \{2\}, \{3, 4, 5, 7\}\}$ n'en est pas une.

Q2. Parmi les ensembles suivants, indiquer sans justification lesquels sont des partitions de $\llbracket 5 \rrbracket$. Parmi les partitions, préciser sans justification lesquelles sont non croisées.

1. $\mathcal{P}_1 = \{\{1, 3\}, \{2, 4, 5\}\},$ 2. $\mathcal{P}_2 = \{\{1, 3\}, \{1, 2\}, \{4, 5\}\},$
3. $\mathcal{P}_3 = \{\{1, 3\}, \{2, 4\}\},$ 4. $\mathcal{P}_4 = \{\{1, 4, 5\}, \{2, 3\}\}.$

Q3. Décrire l'ensemble des partitions non croisées de $\llbracket 4 \rrbracket$.

Pour **Q4**, **Q5** et **Q6**, on se fixe un ensemble fini $A \subset \mathbb{N}$. Cet ensemble est représenté en Python par la liste croissante d'entiers A , définie au préalable. On suppose également que pour ces questions, la variable P désigne une liste de listes croissantes d'entiers triée dans l'ordre lexicographique.

On définit la fonction Python `mystere(P)` par :

```
def mystere(P) :
    L = [False for _ in range(len(A))]
    for i in range(len(P)) :
        if P[i] == [] :
            return False
        else :
            for j in range(len(P[i])) :
                if P[i][j] not in A :
                    return False
                else :
                    # A.index(a) renvoie l'indice de a dans A
                    k = A.index(P[i][j])
                    if L[k] :
                        return False
                    else :
                        L[k] = True
    return not (False in L)
```

Q4. Uniquement dans cette question, on suppose que $A = [1, 2, 3, 4, 5]$. Expliciter sans justification les valeurs de :

1. `mystere([[1, 3], [1, 5], [2, 4]])` 2. `mystere([[1, 3, 4], [2]])`
3. `mystere([[1, 3, 5], [2, 4]])` 4. `mystere([], [1, 2, 3, 4, 5])`.

Q5. Écrire une fonction Python `classe(P, i)` prenant en arguments une variable P représentant une partition $\mathcal{P}$ de A et un entier $i \in A$ qui renvoie une liste L représentant $Cl(i)$.

On définit la fonction Python au code incomplet `est_nc(P)` suivante :

```
def est_nc(P) :
    # On rappelle que A est une liste triée dans l'ordre croissant
    N = len(A)
    for i in range(N) :
        for j in range(i+1,N) :
            for k in range(j+1,N) :
                for l in range(k+1,N) :
                    . . .
```

- Q6.** Recopier et compléter le code de la fonction `est_nc(P)`, sachant qu'elle prend en argument une variable `P` représentant une partition $\mathcal{P}$ de A et qu'elle renvoie `True` si $\mathcal{P}$ est non croisée et `False` sinon.

I.2 - Nombre de partitions non croisées

Pour tout $n \geq 1$, on note C_n le nombre de partitions non croisées d'un ensemble $A \subset \mathbb{N}$ ayant exactement n éléments et $\text{NC}(A)$ l'ensemble des partitions non croisées de A . Par convention, $C_0 = 1$ et $\text{NC}(\emptyset) = \{\emptyset\}$.

Définition 4 (Partition non croisée emboîtée)

Soient A un ensemble fini de $\mathbb{N}$ de cardinal n et $\mathcal{P}$ une partition non croisée de A . On dit que $\mathcal{P}$ est **emboîtée** si le minimum m et le maximum M de A sont dans la même classe ($\text{Cl}(m) = \text{Cl}(M)$). L'ensemble des partitions non croisées et emboîtées de A et son cardinal sont respectivement notés $\text{NCE}(A)$ et D_n .

Par exemple, $\mathcal{S} = \{\{1, 2, 5, 9\}, \{3, 4\}, \{6, 7, 8\}\}$ est une partition non croisée et emboîtée de $\llbracket 9 \rrbracket$.

Le terme « emboîtée » découle naturellement des représentations picturales des partitions (**figure 2**).

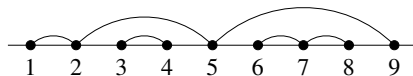


Figure 2 - Représentations picturale de $\mathcal{S}$

- Q7.** Montrer que pour tout $n \in \mathbb{N}$, on a $C_n = D_{n+1}$.

- Q8.** Soit $n \in \mathbb{N}$. On définit l'application Φ_n comme suit :

$$\begin{aligned} \Phi_n : \bigcup_{i=1}^{n+1} \text{NCE}(\llbracket i \rrbracket) \times \text{NC}(\llbracket i+1, n+1 \rrbracket) &\rightarrow \text{NC}(\llbracket n+1 \rrbracket) \\ (Q_1, Q_2) &\mapsto Q_1 \cup Q_2. \end{aligned}$$

En admettant que Φ_n est une application bijective, montrer que $C_{n+1} = \sum_{k=0}^n C_k C_{n-k}$.

- Q9.** (Informatique Pour Tous) Écrire une fonction Python `calcul_C(n)` qui prend en argument un entier naturel `n` et qui renvoie la valeur C_n .

Partie II - Logique et étude du problème Horn-Sat

Dans cette partie, $\wedge$, $\vee$, $\neg$, désignent respectivement les connecteurs de conjonction, de disjonction et de négation. Les symboles V, F désignent respectivement les valeurs de vérité VRAI et FAUX du calcul propositionnel.

Dans la suite, on s'intéresse au problème Horn-Sat, qui peut être décrit de la façon suivante : étant donnée une formule de Horn P , existe-t-il une interprétation I telle que $[P]_I = V$?

L'objectif est d'expliciter un algorithme permettant de résoudre ce problème.

Définition 5 (Formule propositionnelle)

Soit X un ensemble de symboles appelés variables propositionnelles. Les **formules propositionnelles** sur X sont alors définies de façon inductive comme suit :

- V, F sont des formules propositionnelles ;
- tout élément x de X est une formule propositionnelle ;
- si P et Q sont des formules propositionnelles, alors $\neg P$, $(P \wedge Q)$, $(P \vee Q)$ sont des formules propositionnelles.

Dans la suite, les variables propositionnelles et formules propositionnelles considérées sont construites à l'aide d'un ensemble X qui ne sera pas explicité.

Définition 6 (Littéral)

Soit x une variable propositionnelle. Un **littéral** est la formule x ou la formule $\neg x$. Le littéral x (respectivement $\neg x$) est un littéral positif (respectivement négatif).

Définition 7 (Clause disjonctive)

Soient $n \in \mathbb{N}$, $x_1, x_2, \dots, x_n$ des littéraux. La formule $x_1 \vee x_2 \vee \dots \vee x_n$ est appelée **clause disjonctive** à n littéraux (ou de taille n). Par convention, $()$ désigne la **clause disjonctive vide**, c'est-à-dire la clause sans littéral.

Une **clause de Horn** est une clause disjonctive contenant au plus un littéral positif.

Une **clause unitaire** est une clause composée d'un unique littéral.

Définition 8 (Forme normale conjonctive)

Soit P une formule propositionnelle. On dit que P est une **forme normale conjonctive** s'il existe un entier $n \in \mathbb{N}$, $C_1, C_2, \dots, C_n$ des clauses disjonctives vérifiant $P = C_1 \wedge C_2 \wedge \dots \wedge C_n$. Par convention, une forme normale conjonctive sans clause est représentée par $\emptyset$.

Définition 9 (formule de Horn)

Soit P une formule propositionnelle. On dit que P est une **formule de Horn** s'il existe $n \in \mathbb{N}$, $C_1, C_2, \dots, C_n$ des clauses de Horn vérifiant $P = C_1 \wedge C_2 \wedge \dots \wedge C_n$.

Définition 10 (Interprétation)

Soient $n \in \mathbb{N}$ et $x_1, x_2, \dots, x_n$ des variables propositionnelles. Une **interprétation** est une application $I : \{x_1, x_2, \dots, x_n\} \rightarrow \{V, F\}$.

Définition 11 (Évaluation)

Soient P une formule propositionnelle, $x_1, x_2, \dots, x_n$ les variables propositionnelles apparaissant dans P et I une interprétation sur $\{x_1, x_2, \dots, x_n\}$. L'**évaluation** de P suivant l'interprétation I que l'on note $[P]_I$ est définie par récurrence de la façon suivante :

- si $P \in \{V, F\}$, alors $[P]_I = P$;
- si $P \in \{x_1, x_2, \dots, x_n\}$, alors $[P]_I = I(P)$;
- si $P = \neg Q$ où Q est une formule propositionnelle, alors $[P]_I = \neg[Q]_I$;
- si $P = A \circ B$ où A, B sont des formules propositionnelles et $\circ \in \{\wedge, \vee\}$, alors $[P]_I = [A]_I \circ [B]_I$.

Par convention, $[\emptyset]_I = F$ et $[\emptyset]_I = V$.

Définition 12 (Satisfiable)

Soit P une formule propositionnelle. On dit que P est **satisfiable** s'il existe une interprétation I telle que $[P]_I = V$.

Q10. Pour chacune des formules suivantes, montrer qu'elle est satisfiable ou qu'elle est non satisfiable.

1. $P_1 = (x \vee y) \wedge (\neg x \vee \neg y) \wedge (x \vee \neg y)$,
2. $P_2 = (x) \wedge (\neg x \vee \neg y) \wedge (\neg y \vee z) \wedge (z)$,
3. $P_3 = ()$,
4. $P_4 = (x \vee \neg y \vee \neg t) \wedge (z \vee \neg t \vee \neg x \vee \neg y)$.

Q11. Parmi les fomules de la **Q10**, lesquelles sont des formules de Horn? On ne justifiera pas la réponse.

Définition 13 (Propagation unitaire)

Soit P une forme normale conjonctive contenant une clause unitaire (x) . On construit une formule P' à partir de P de la façon suivante :

1. supprimer de P toutes les clauses où x apparaît;
2. enlever toutes les occurrences du littéral $\neg x$.

Cette procédure de simplification est appelée **propagation unitaire**.

Par exemple, si $P = (x \vee \neg y \vee z) \wedge (\neg x \vee y \vee z) \wedge (x)$, on a alors $P' = (y \vee z)$. Si $P = (x) \wedge (\neg x)$, on a alors $P' = ()$.

Dans la suite, on note $\Pi(P)$ une formule sans clause unitaire obtenue en itérant la propagation unitaire sur P . On admet que si $\Pi(P)$ ne contient pas de clause vide $()$ alors $\Pi(P)$ est unique et que si $\Pi(P)$ contient au moins une clause vide alors toute formule sans clause unitaire obtenue par itération de la propagation unitaire sur P contient au moins une clause vide.

Q12. On pose :

$$P = (x_1) \wedge (x_1 \vee x_2 \vee x_3) \wedge (\neg x_1 \vee x_3 \vee x_4) \wedge (\neg x_1 \vee \neg x_2) \wedge (\neg x_2 \vee \neg x_1 \vee x_3) \\ \wedge (x_3 \vee \neg x_4 \vee x_5) \wedge (\neg x_1 \vee x_2 \vee x_5).$$

Calculer $\Pi(P)$. On pourra donner le résultat directement sans détailler les calculs.

Q13. Soit P une formule de Horn. Montrer que $\Pi(P)$ est une formule de Horn.

Q14. Soit P une forme normale conjonctive. Montrer que P est satisfiable si et seulement si $\Pi(P)$ est satisfiable.

Q15. Soit P une forme normale conjonctive. Montrer que si $()$ apparaît dans $\Pi(P)$, alors P n'est pas satisfiable.

Q16. Soit C une clause de Horn. Montrer que si C n'est ni la clause vide ni une clause unitaire positive, alors C est satisfiable.

Q17. Soit P une formule de Horn ne contenant ni de clause vide ni de clause unitaire positive. Montrer que P est satisfiable.

Q18. Soit P une formule de Horn. Montrer que P n'est pas satisfiable si et seulement si $()$ apparaît dans $\Pi(P)$.

Partie III - Étude des classes sylvestres

Étant donné un arbre binaire de recherche T , sa classe sylvestre est l'ensemble des mots qui donnent l'arbre T après insertion dans l'arbre vide. L'objectif de cette partie est de donner une caractérisation et une description de cet ensemble. On commence par rappeler la structure des arbres binaires de recherche ainsi que leurs propriétés usuelles. Puis, on introduit la notion de S-équivalence sur les mots qui permet de caractériser la classe sylvestre d'un arbre T . Enfin, à l'aide du produit de mélange, on présente un algorithme calculant la classe sylvestre d'un arbre donné.

Dans toute la suite, Σ désigne un alphabet fini totalement ordonné. Le symbole ε désigne le mot vide.

III.1 - Algorithme d'insertion dans un arbre binaire

Définition 14 (Arbre binaire)

Un **arbre binaire** T (étiqueté par les éléments de Σ) est récursivement soit :

- l'arbre vide que l'on note $\circ$;
- un triplet (T_g, r, T_d) où r est un élément de Σ , T_g et T_d des arbres binaires. Les éléments r , T_g et T_d sont respectivement appelés racine, sous-arbre gauche et sous-arbre droit de T .

Définition 15 (Arbre binaire de recherche)

Un **Arbre Binaire de Recherche** (abrégé en **ABR**) T est récursivement soit :

- l'arbre vide ;
- un triplet (T_g, r, T_d) où r est un élément de Σ , T_g et T_d des **ABR**. De plus, toute valeur apparaissant dans T_g est strictement inférieure à r et toute valeur apparaissant dans T_d est supérieure ou égale à r .

Définition 16 (Insertion dans un arbre binaire)

L'**insertion** d'un élément a de Σ dans un arbre binaire T est une opération que l'on note $T \leftarrow a$ définie récursivement comme suit :

- si $T = \circ$, alors $T \leftarrow a = (\circ, a, \circ)$;
- si $T = (T_g, r, T_d)$ et $r \leq a$, alors $T \leftarrow a = (T_g, r, T_d \leftarrow a)$;
- si $T = (T_g, r, T_d)$ et $r > a$, alors $T \leftarrow a = (T_g \leftarrow a, r, T_d)$.

On définit récursivement alors l'insertion d'un mot w dans un arbre binaire T noté également $T \leftarrow w$ comme suit :

- si $w = \varepsilon$, alors $T \leftarrow w = T$;
- si $w = av$ avec $a \in \Sigma$, alors $T \leftarrow w = (T \leftarrow a) \leftarrow v$.

Dans le cas où T est un **ABR**, on admet que $T \leftarrow a$ et $T \leftarrow w$ sont également des **ABR**.

Étant donné un mot w sur Σ , l'**arbre binaire de recherche associé** à w est l'arbre $\circ \leftarrow w$.

Définition 17 (Classe sylvestre)

Soit T un arbre binaire de recherche. La **classe sylvestre** de T que l'on note $\text{Syl}(T)$ est l'ensemble des mots w vérifiant : $(\circ \leftarrow w) = T$.

Par exemple, si $T = ((\circ, a, \circ), b, (\circ, c, \circ))$, on a $\text{Syl}(T) = \{bac, bca\}$.

Et si $T = \circ$, alors $\text{Syl}(T) = \{\varepsilon\}$.

Pour simplifier la représentation des arbres binaires, on peut utiliser des graphes.

Par exemple, l'arbre $T = ((\circ, a, ((\circ, b, \circ), a, \circ)), f, (\circ, c, \circ))$ est représenté dans la **figure 3**.

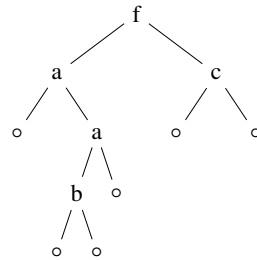


Figure 3 - Représentation de T

Dans la suite, les lettres de Σ sont représentées en Ocaml par des `char` et les mots sur l'alphabet Σ par des `char list`. Ainsi, le mot $w = \text{"arbre"}$ est représenté par la liste $w = ['a'; 'r'; 'b'; 'r'; 'e']$. On représente les arbres binaires en Ocaml à l'aide de la structure `arbre` suivante :

```
type 'a arbre = Vide | Noeud of ('a arbre) * 'a * ('a arbre).
```

Ainsi, l'arbre $T = (\circ, a, (\circ, b, \circ))$ est représenté en Ocaml par :

```
Noeud(Vide, 'a', Noeud(Vide, 'b', Vide)).
```

Q19. Représenter le graphe de l'ABR associé au mot "fantastique", l'ordre sur les lettres étant l'ordre alphabétique.

Q20. Écrire une fonction récursive en Ocaml de signature :

```
insertion_lettre : char -> char arbre -> char arbre
```

et telle que `insertion_lettre a t` est l'arbre binaire obtenu en insérant la lettre `a` dans l'arbre binaire `t`.

Q21. Écrire une fonction récursive en Ocaml de signature :

```
insertion_mot : char list -> char arbre -> char arbre
```

et telle que `insertion_mot w t` est l'arbre binaire obtenu en insérant le mot `w` dans l'arbre binaire `t`.

Définition 18 (Lecture préfixe)

Soit T un arbre binaire. La **lecture préfixe** de T que l'on note w_T est le mot défini récursivement par :

- si $T = \circ$, alors $w_T = \varepsilon$;
- si $T = (T_g, r, T_d)$, alors $w_T = rw_gw_d$ où w_g et w_d sont les lectures préfixes respectives de T_g et de T_d .

Q22. Expliciter w_T pour l'arbre binaire T représenté ci-dessous :

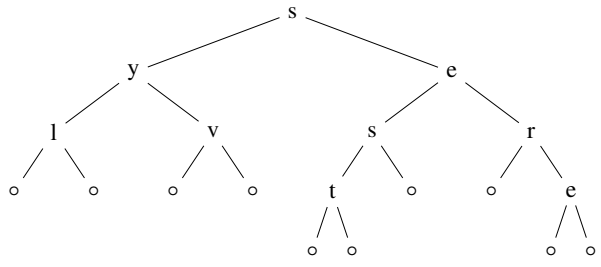


Figure 4 - Représentation de T de Q22

Q23. Écrire une fonction récursive en Ocaml de signature :

`prefixe : char arbre -> char list`

et telle que `prefixe t` est le mot qui correspond à la lecture préfixe de l'arbre `t`.

Q24. Soit T un **ABR**, soit w_T la lecture préfixe de T . Montrer que T est l'**ABR** associé à w_T .

III.2 - Une relation d'équivalence sur les mots

Définition 19 (S-adjacence)

Soient u et v deux mots sur Σ . On dit que u est **S-adjacent** à v (ou que u et v sont S-adjacents) s'il existe trois lettres $\mathbf{a} < \mathbf{b} \leq \mathbf{c}$ de Σ et trois mots f_1, f_2, f_3 sur Σ vérifiant :

$$(u = f_1 \mathbf{b} f_2 \mathbf{a} c f_3 \text{ et } v = f_1 \mathbf{b} f_2 \mathbf{c} a f_3) \text{ ou } (u = f_1 \mathbf{b} f_2 \mathbf{c} a f_3 \text{ et } v = f_1 \mathbf{b} f_2 \mathbf{a} c f_3).$$

Définition 20 (S-équivalence)

Soient u et v deux mots sur Σ . On dit que u est **S-équivalent** à v (ou que u et v sont S-équivalents) s'il existe $n \in \mathbb{N}$, $(w_0, w_1, \dots, w_n) \in (\Sigma^*)^{n+1}$ vérifiant :

$$u = w_0, v = w_n, \forall i \in \{0, 1, \dots, n-1\}, w_i \text{ est S-adjacent à } w_{i+1}.$$

Q25. Montrer que la relation " être S-équivalent à " est bien une relation d'équivalence sur Σ^* .

Q26. Soient a, b, c trois lettres de Σ vérifiant $a < b \leq c$. Montrer que pour tout arbre binaire de recherche T contenant au moins la lettre b , on a : $T \leftarrow ac = T \leftarrow ca$. On pourra raisonner par récurrence sur le nombre de nœuds de T .

Q27. Montrer que si deux mots u et v sont S-équivalents, alors les **ABR** $(\circ \leftarrow u)$ et $(\circ \leftarrow v)$ sont égaux.

Q28. Soit w un mot de longueur $n \geq 1$ sur Σ , soit r la première lettre de w . On veut montrer qu'il existe un mot $w' = ruv$ où u (respectivement v) est un mot dont toutes les lettres sont plus petites strictement (respectivement plus grandes au sens large) que r et tels que w et w' sont S-équivalents. On note A l'ensemble des mots S-équivalents à w .

Pour tout $a = a_0a_1 \dots a_{n-1} \in A$, on pose :

$$N(a) = \{(i, j) \in \llbracket 1, n-1 \rrbracket^2 \mid i < j, a_j < r \leq a_i\}.$$

- (a) Justifier que l'ensemble A est fini.
- (b) Justifier que tous les mots de A commencent par r .
- (c) Soit $a \in A$. Montrer que si $N(a) \neq \emptyset$, alors il existe $k \in \llbracket 1, n-2 \rrbracket$ tel que $(k, k+1) \in N(a)$.
- (d) Soit $a \in A$. Montrer que si $N(a) = \emptyset$, alors il existe u (respectivement v) un mot dont toutes les lettres sont plus petites strictes (respectivement plus grandes ou égales) que r et tels que $a = ruv$.
- (e) Soit a un élément de A tel que le nombre d'éléments de $N(a)$ soit le plus petit possible. Montrer par l'absurde que $N(a) = \emptyset$ et en déduire qu'il existe u, v vérifiant les conditions de la question tels que ruv et w sont S-équivalents.

Q29. Montrer que tout mot w est S-équivalent à w_T , où w_T est la lecture préfixe de $T = (\circ \leftarrow w)$. On pourra raisonner par récurrence sur la longueur du mot w et exploiter les résultats de la **Q28**.

Q30. En déduire que deux mots sont S-équivalents si et seulement si ils sont des éléments d'une même classe sylvestre.

III.3 - Construction de classes sylvestres

Pour construire des classes sylvestres, il est utile d'utiliser le produit de mélange.

Définition 21 (Mélange)

Soient u et v deux mots sur Σ . Le **mélange** de u et v , noté $u \sqcup v$, est l'ensemble récursivement défini comme suit :

- si $v = \varepsilon$, alors $u \sqcup v = \{u\}$;
- si $u = \varepsilon$, alors $u \sqcup v = \{v\}$;
- si $u = au'$ et $v = bv'$ où a et b sont des lettres, u' et v' des mots, alors :

$$u \sqcup v = \{aw \mid \exists w \in u' \sqcup v\} \cup \{bw \mid \exists w \in u \sqcup v'\}.$$

Définition 22 (Mélange de langages)

Soient L et M deux langages. Le **mélange des langages** L et M , noté $L \sqcup M$, est défini par :

$$L \sqcup M = \bigcup_{u \in L, v \in M} u \sqcup v.$$

Si au moins un des deux langages est égal à l'ensemble $\emptyset (\neq \{\varepsilon\})$, alors on a $L \sqcup M = \emptyset$.

Étant donné un **ABR** $T = (T_g, r, T_d)$, on peut montrer que :

$$\text{Syl}(T) = \{rw \mid \exists w \in \text{Syl}(T_g) \sqcup \text{Syl}(T_d)\}.$$

On admet ce résultat pour la suite de cette sous-partie. On note que $\text{Syl}(\circ) = \{\varepsilon\}$.

Q31. Expliciter sans justification tous les éléments de l'ensemble $abba \sqcup ba$.

Q32. Écrire une fonction récursive en Ocaml de signature :

`ajout_lettre : char -> char list list -> char list list`

et telle que `ajout_lettre a liste` est la liste de mots de la forme $a:w$ où w est un élément de `liste`.

Q33. Écrire une fonction récursive en Ocaml de signature :

`shuffle : char list -> char list -> char list list`

et telle que `shuffle u v` est une liste contenant exactement tous les mots de $u \sqcup v$ avec répétitions possibles d'un même mot. On devra utiliser la fonction auxiliaire `ajout_lettre` et l'opérateur de concaténation `@`.

Q34. Écrire une fonction récursive en Ocaml de signature :

`shuffle_l : char list list -> char list list -> char list list`

et telle que `shuffle_l l m` est une liste contenant exactement tous les mots de $L \sqcup M$ où la liste `l` (respectivement `m`) représente le langage L (respectivement M), avec répétitions possibles d'un même mot. Seules les fonctions définies précédemment peuvent être utilisées en fonctions auxiliaires.

Q35. Écrire une fonction récursive en Ocaml de signature :

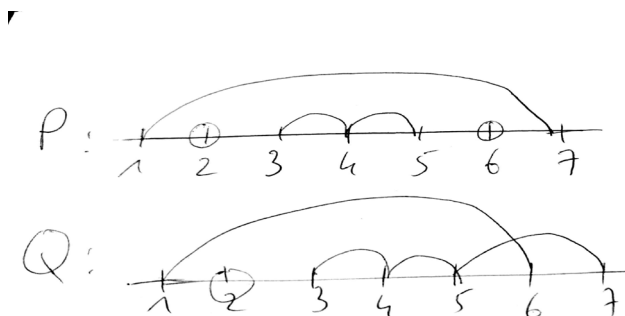
`sylvestre_T : char arbre -> char list list`

et telle que `sylvestre_T t` est une liste contenant exactement tous les éléments de la classe sylvestre de `t`. Seules les fonctions définies précédemment peuvent être utilisées en fonctions auxiliaires.

FIN

SOLUTION

Q1. Voici la représentation graphique de P et Q .



P : Les ensembles $\{1, 7\}$, $\{2\}$, $\{3, 4, 5\}$, $\{6\}$ forment bien une partition de $[[7]]$. En outre, aucune de ces classes ne se croisent (il suffit de voir le dessin) donc elle est non croisée.

Q : Les ensembles $\{1, 6\}$, $\{2\}$, $\{3, 4, 5, 7\}$ forment bien une partition de $[[7]]$. Par contre, les classe $\{1, 6\}$ et $\{3, 4, 5, 7\}$ se croisent (cela se voit sur le dessin ou bien, en choisissant

$$a = 1 < b = 3 < c = 6 < d = 7,$$

les classes de a et c sont identiques, les classes de b et d sont identiques mais les classes de a et b sont différentes). Par conséquent, Q n'est pas une partition non croisée.

Commentaires : Question visant à familiariser le candidat avec cette notion. Y prêter beaucoup d'attention (utile pour le codage futur) et ne pas hésiter à faire des dessins comme le prône le sujet.

Q2. P_2 n'est pas une partition (1 apparaît deux fois), P_3 non plus (il manque 5), P_1 et P_4 sont des partitions. P_4 n'est pas croisée mais P_1 l'est (les parties $\{1, 3\}$ et $\{2, 4, 5\}$ se croisent, soit par un dessin, soit en choisissant

$$a = 1 < b = 2 < c = 3 < d = 4$$

avec $Cl(a) = Cl(c)$, $Cl(b) = Cl(d)$ et $Cl(a) \neq Cl(b)$).

Commentaires : Même remarque qu'à la question précédente.

Q3. Voici les partitions non croisées demandées :

- parties à quatre éléments : $\{\{1, 2, 3, 4\}\}$
- parties dont une a trois éléments : a) $\{\{1, 2, 3\}, \{4\}\}$, b) $\{\{1, 2, 4\}, \{3\}\}$, c) $\{\{1, 3, 4\}, \{2\}\}$, d) $\{\{1\}, \{2, 3, 4\}\}$
- parties dont toutes ont deux éléments : a) $\{\{1, 2\}, \{3, 4\}\}$, b) $\{\{1, 4\}, \{2, 3\}\}$
- parties dont au moins une a deux éléments : a) $\{\{1, 2\}, \{3\}, \{4\}\}$, b) $\{\{1, 3\}, \{2\}, \{4\}\}$, c) $\{\{1, 4\}, \{2\}, \{3\}\}$, d) $\{\{2, 3\}, \{1\}, \{4\}\}$, e) $\{\{2, 4\}, \{1\}, \{3\}\}$, f) $\{\{3, 4\}, \{1\}, \{2\}\}$
- parties a un seul élément : $\{\{1\}, \{2\}, \{3\}, \{4\}\}$.

Commentaires : Question sans difficulté mais il faut être attentif, la réponse attendue doit être sans aucune erreur (tout ou rien).

Q4. Voici les valeurs renvoyées : 1. False, 2. False, 3. True, 4. False.

Le premier *if* vérifie que chaque partie est non vide (donc 4. renvoie False). La première boucle *for j* vérifie que les éléments de chaque partie appartient à A. Le *else* marque un élément $P[i][j]$ lorsque sa valeur est rencontré. S'il est rencontré une deuxième fois, cet élément apparait deux fois et l'entrée n'est pas une partition (donc 1. renvoie False). La commande *not (False in L)* vérifie que tous les éléments de A ont été rencontré (donc 2. renvoie False). Comme 3. n'amène aucune exclusion, il renvoie True.

Commentaires : Question testant la capacité du candidat à faire tourner un code qu'il n'a jamais rencontré et qui n'est pas immédiat à interpréter. A nouveau, c'est une question tout ou rien.

Q5.

```
def classe(P,i):
    for c in P:
        if i in c:
            return c
```

Commentaires : La difficulté n'est pas dans le code informatique mais dans la compréhension de la structure informatique modélisant les partitions. On peut utiliser aussi les indices des éléments de P puis des indices pour les éléments d'un élément de i !

Q6.

```
def est_nc(P):
    N = len (A)
    for i in range (N):
        for j in range (i +1 ,N):
            for k in range ( j +1 ,N):
                for l in range ( k+1 ,N) :
                    if classe(P,i)==classe(P,L[k] and classe(
                        P,j)==classe(P,l) and classe(P,i) !=
                        classe(P,j):
                        return False
    return True
```

Commentaires : Question sélective car elle demande à la fois de bien comprendre la modélisation ET de ne pas oublier qu'il faut faire les tester sur des quadruplets ordonnés de nombres a,b,c,d avec $a < b < c < d$. On peut utiliser une quadruple boucle *for* imbriquée et mettre des tests booléens du type *if a < b* etc.

Q7. Soit P est une partition non croisée de $\{1, \dots, n\}$ alors, en ajoutant $n + 1$ à la partie de P contenant 1, on obtient toujours une partition P_E non croisée et qui est emboîtée de $\{1, \dots, n + 1\}$. Réciproquement, si Q est une partition non croisée et emboîtée de $\{1, \dots, n + 1\}$,

en enlevant $n + 1$ à la partie de Q contenant 1, on obtient une partition Q' non croisée de $\{1, \dots, n\}$. Par conséquent, la donnée d'une partition non croisée de $\{1, \dots, n\}$ est équivalente (bijectivement) à celle d'une partition non croisée et emboîtée de $\{1, \dots, n + 1\}$ (formellement $Q \mapsto Q'$ et $P \mapsto P_E$ sont les bijections réciproques entre ces deux ensembles).

Commentaires : Comme toute question de dénombrement, peu de candidats fournissent une réponse un tant soit peu pertinente. On ne peut se limiter à ajouter ou enlever un élément, il faut vérifier que le processus est réversible et que les associations sont uniques. Souvent les candidats ne font qu'une implication (qu'ils pensent être une équivalence !). Pas besoin de trop formaliser, au moins faire comprendre au correcteur que vous voyez les points clés de l'argument (la double association réversible $Q \mapsto Q'$ et $Q' \mapsto Q$ en l'expliquant en français).

Q8. Comme Φ_n est bijective, les ensembles de départ et d'arrivée ont le même nombre d'éléments donc

$$\begin{aligned} C_{n+1} &= \text{card} \left(\bigcup_{i=1}^{n+1} NCE([i]) \times NC[[i+1, n+1]] \right) \\ &= \sum_{i=1}^{n+1} \text{card}(NCE([i]) \times NC[[i+1, n+1]]) \end{aligned}$$

(union disjointe) et comme $\text{card}(A \times B) = \text{card}(A) \text{card}(B)$, on obtient

$$\begin{aligned} C_{n+1} &= \sum_{i=1}^{n+1} \text{card}(NCE([i])) \text{card}(NC[[i+1, n+1]]) \\ &= \sum_{i=1}^{n+1} D_i C_{n+1-i} \end{aligned}$$

(car $[[i+1, n+1]]$ est un ensemble à $(n+1) - (i+1) + 1 = n - i + 1$ éléments). D'après la question précédente, on a $D_i = C_{i-1}$ et en faisant le changement de variable $k = i - 1$ ($i = 1$ donc $k = 0$ et $i = n + 1$ donc $k = n$, $i = k + 1$ donc $n + 1 - i = n - k$), ce qui nous donne

$$C_{n+1} = \sum_{k=0}^n C_k C_{n-k}.$$

Commentaires : Comme le sujet donne beaucoup d'indications, seule la rigueur sera valorisée : cardinal d'une union (en spécifiant le caractère disjoint), d'un produit cartésien, etc.

Q9.

```
# complexité linéaire
def C(n):
    tab = [1 for i in range(n+1) ]
    for i in range(n):
        S=0
        for k in range(i+1):
            S=S+tab[k]*tab[i-k]
```

```

        tab[i+1]=S
    return tab[n]

# complexité exponentielle

def C(n):
    if n==0:
        return 1
    S=0
    for k in range(n):
        S=S+C(k)*C(n-1-k)
    return S

```

Commentaires : Question élémentaire de programmation qui ne doit pas poser de difficulté. Toute réponse correcte est récompensée indépendamment de sa complexité car le sujet n'en parle pas (linéaire ou exponentielle)

Q10. P1 est satisfiable en posant $I(x) = V$ et $I(y) = F$. P2 est satisfiable en posant $I(x) = V$, $I(y) = F$, $I(z) = V$. P3 n'est pas satisfiable par définition. P4 est satisfiable en posant $I(y) = F$ par exemple.

Commentaires : Question élémentaire. C'est une réponse tout ou rien donc soyez attentif.

Q11. Il s'agit de P2, P3 et P4. P1 ne l'est pas (sa première clause contient deux littéraux positifs x et y).

Commentaires : Le but est de permettre au candidat de se familiariser avec la notion. Toujours une valorisation tout ou rien.

Q12. La première clause de P est unitaire (x_1). On efface toutes les clauses de P contenant x_1 et on efface toutes ses négations $\neg x_1$. On obtient formule

$$P_1 = (x_3 \vee x_4) \wedge (\neg x_2) \wedge (\neg x_2 \vee x_3) \wedge (x_3 \vee \neg x_4 \vee x_5) \wedge (x_2 \vee x_5).$$

Elle contient la clause unitaire ($\neg x_2$). On efface toutes les clauses de P contenant $\neg x_2$ et on efface toutes ses négations $\neg(\neg x_2) = x_2$. On obtient formule

$$P_2 = (x_3 \vee x_4) \wedge (x_3 \vee \neg x_4 \vee x_5) \wedge (x_5).$$

Elle contient la clause unitaire (x_5). On efface toutes les clauses de P contenant x_5 et on efface toutes ses négations $\neg x_5$. On obtient formule

$$P_2 = (x_3 \vee x_4).$$

Elle ne contient plus de clause unitaire et le processus s'arrête donc

$$\Pi(P) = P_2 = (x_3 \vee x_4).$$

Commentaires : Mêmes remarques qu'à la question précédente.

Q13. Si P est une formule de Horn. Si elle ne contient aucune clause unitaire alors $\Pi(P) = P$ est une formule de Horn. Sinon, elle en possède une (x) . En éliminant toutes les clauses contenant x , la nouvelle formule reste toujours de Horn (c'est toujours une forme normale conjonctive, chaque clause contient toujours au plus un littéral positif). Ensuite, en éliminant $\neg x$ dans toutes les clauses restantes, la formule obtenue P' est toujours une formule de Horn (c'est toujours une forme normale conjonctive, le nombre de littéraux positifs ne change pas). En itérant cela, la formule finale est toujours une formule de Horn donc $\Pi(P)$ l'est.

Commentaires : Question sans grande difficulté ... si le candidat s'est familiarisé avec ces nouvelles notions (Horn, propagation unitaire).

Q14. Si P ne possède pas de clause unitaire alors $\Pi(P) = P$ est aussi satisfiable. Supposons que P possède une clause unitaire (x) . Si P est satisfiable donc il existe une interprétation I telle que $P_I = V$ donc, pour toute clause C de P , $C_I = V$. Soit C est une clause de P . Si elle contient ni x , ni $\neg x$, cette clause n'est pas modifiée et $C_I = V$. Si elle contient x , elle est effacée et n'affectera pas la valeur de P' . Si elle contient $\neg x$, elle s'écrit $C = (\neg x \vee \ell_i \vee \dots \vee \ell_j)$ où $\ell_i, \dots, \ell_j$ sont des littéraux autres que x et $\neg x$ donc elle devient la clause $C' = (\neg x \vee \ell_i \vee \dots \vee \ell_j)$ où $\ell_i, \dots, \ell_j$ sont des littéraux autres que x et $\neg x$. En notant I' la même interprétation que I mais en n'attribuant aucune valeur à x (on le fait disparaître des variables) alors $C'_{I'} = C_I = V$ par construction d'où $P'_{I'} = P_I = V$ ce qui assure la satisfiabilité de P' . Par itération du processus $P \mapsto P'$, on est assuré que $\Pi(P)$ est satisfaite (en effaçant de l'interprétation I toutes les variables associées aux littéraux unitaires qui seront éliminées au cours du processus).

Supposons que $\Pi(P)$ est satisfiable pour une certaine interprétation J (portant sur les variables intervenant dans $\Pi(P)$). Pour chaque variable x dont le littéral unitaire associé sera effacé au cours du processus de propagation, on attribue la valeur $J(x) = V$ si le littéral associé est x et $J(x) = F$ si le littéral associé est $\neg x$, ce qui nous donne une nouvelle interprétation J_P sur l'ensemble des variables de $\Pi(P)$ auquel on ajoute les variables éliminées lors de la propagation (c'est-à-dire l'ensemble des variables de P). Il est immédiat que $P_{J_P} = \Pi(P)_J$ (car cela est vrai pour les clauses à chaque étape donc pour chaque formule à chaque étape) et comme $\Pi(P)_J = V$, on peut affirmer que P est satisfiable.

Commentaires : Question difficile et réussie probablement que par les meilleurs candidats.

Q15. Si $()$ apparaît dans $\Pi(P)$ alors $\Pi(P)$ n'est pas satisfiable (car $()$ n'est pas satisfiable par définition) donc, d'après la question précédente, P ne l'est pas non plus.

Commentaires : Question élémentaire ... si le candidat lit bien le sujet :-).

Q16. Comme C est une clause non vide et non unitaire positive, elle contient au moins un littéral négatif $\neg x$ donc elle s'écrit $C = (\neg x \vee \ell_i \vee \dots \vee \ell_j)$ où $\ell_i, \dots, \ell_j$ sont des littéraux autres que x et $\neg x$. En choisissant n'importe quelle interprétation I tel que $I(x) = F$, on a $I(\neg x) = V$ donc $C_I = V$ d'où la satisfiabilité de C .

Commentaires : Même remarque qu'à la question précédente.

Q17. P s'écrit $P = C_1 \wedge \dots \wedge C_m$ avec, pour chaque i , C_i une clause non vide et non unitaire positive. Pour chaque i , on choisit une variable x_i telle que $\neg x_i$ est un littéral négatif de C_i . On considère alors l'interprétation $I(x_i) = F$ pour chaque i , et on attribue n'importe quelle

valeur aux autres variables intervenant dans P . Il est immédiat que

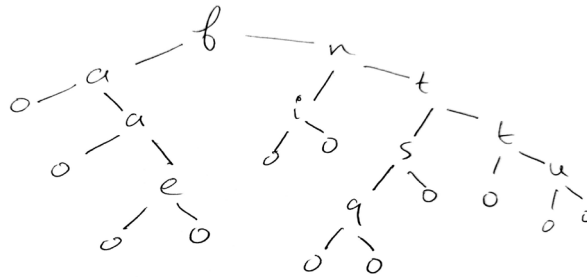
$$P_I = (C_1)_I \wedge \cdots (C_m)_I = V \wedge \cdots \wedge V = V$$

donc P est satisfiable.

Commentaires : Question de niveau standard pour les candidats ayant une maîtrise convenable de la logique booléenne. Elle s'avère donc discriminante.

Q18. D'après la question 15, si $()$ apparaît dans $\Pi(P)$ alors P n'est pas satisfiable. Si $()$ n'apparaît pas dans $\Pi(P)$ alors, par construction, $\Pi(P)$ ne contient pas aussi de clause unitaire positive (l'itération de la propagation unitaire les a toutes fait disparaître). Ainsi, $\Pi(P)$ est une formule de Horn ne contenant pas $()$ et aucune clause unitaire positive donc, d'après la question précédente, $\Pi(P)$ est satisfiable donc P aussi (cf. question 14). Cela prouve l'équivalence voulue.

Commentaires : Question pour les meilleures copies qui sont aptes à synthétiser les différents résultats et à structurer correctement les résultats.



Q19. **Commentaires :** Question visant à familiariser le candidat avec cette notion. Y prêter beaucoup d'attention (utile pour le codage futur ou les preuves théoriques).

Q20.

```
let rec insertion_lettre a t =
  match t with
  | Vide -> Noeud ( Vide , a , Vide )
  | Noeud(g,r,d) when r <= a -> Noeud (g,r,insertion_lettre a
    d )
  | Noeud(g,r,d) -> Noeud(insertion_lettre a g,r,d) ;;
```

Commentaires : Question de cours qui ne doit pas vous poser de difficulté.

Q21.

```
let rec insertion_mot w t =
  match w with
```



```
| [ ] -> t
| a::q -> insertion_mot q (insertion_lettre a t );;
```

Commentaires : Question élémentaire sans difficulté particulière.

Q22. La lecture correspond à un parcours racine-gauche-droite c'est-à-dire un parcours préfixe. On obtient $w_T = \text{syvestre}$.

Commentaires : Question visant à familiariser le candidat avec cette notion ET c'est quasiment une question de cours.

Q23.

```
let rec prefixe t =
  match t with
  | Vide -> []
  | Noeud(g ,r,d) -> r::(prefixe g)@(prefixe d);;
```

Commentaires : Mêmes remarques qu'à la question précédente.

Q24. On procède par récurrence forte sur le nombre de noeuds de l'arbre. Si l'arbre T est vide alors $w_T = \varepsilon$ qui correspond bien à la lecture préfixe de T . Supposons la propriété vérifiée pour tout arbre ayant un nombre de noeuds strictement à n , n entier non nul. Soit T un arbre à n noeuds donc non vide (car $n \geq 1$) d'où T s'écrit (g, r, d) . Comme g et d sont deux arbres ayant strictement moins de n noeuds, par hypothèse de récurrence, la lecture préfixe de g (resp. d) forme le mot w_g (resp. w_d) donc la lecture préfixe de T est $r w_g w_d$ c'est-à-dire $w_T = r w_g w_d$. Ceci prouve la propriété au rang n et achève la récurrence.

Commentaires : Question difficile pour de nombreux candidats car ... ils ne font pas de preuve. Il est attendu un argument rigoureux (pas une itération ou des exemples ou des arguments non solidement fondés). La preuve doit être fondée sur les définitions et propriétés démontrées. N'hésitez pas à retravailler ce type de preuve, cela sert fréquemment à ce concours et aux autres (Mines-Ponts, CentraleSupElec). Ce type de preuve est accessible aux étudiants de niveau moyen ou plus ... avec un peu d'entraînement.

Q25. symétrie. Si u est S -équivalent à v , il existe un entier n et des mots $w_0, \dots, w_n$ avec $w_0 = u$, $w_n = v$ et, pour tout i , w_i est S -adjacent à w_{i+1} . En posant $w'_0 = w_n$, $w'_1 = w_{n-1}$, ..., $w'_n = w_0$ (i.e. $w'_i = w_{n-i}$ pour chaque i), on peut affirmer que $w'_0 = w_n = v$, $w'_n = w_0 = u$, $w_n = w'_0$ est S -adjacent à $w_{n-1} = w'_1$, $w_{n-1} = w'_1$ est S -adjacent à $w_{n-2} = w'_2$, ..., $w_1 = w'_{n-1}$ est S -adjacent à $w_0 = w'_n$ (car la S -adjacence est manifestement symétrique) donc v est S -équivalent à u .

transitivité. Si u est S -équivalent à v , il existe un entier n et des mots $w_0, \dots, w_n$ avec $w_0 = u$, $w_n = v$ et, pour tout i , w_i est S -adjacent à w_{i+1} . Si v est S -équivalent à w , il existe un entier m et des mots $w'_0, \dots, w'_m$ avec $w'_0 = v$, $w'_m = w$ et, pour tout i , w'_i est S -adjacent à w'_{i+1} . On pose

$$z_0 = w_0 = u, z_1 = w_1, \dots, z_n = w_n = v, z_{n+1} = w'_1, z_{n+2} = w'_2, \dots, z_{n+m} = w'_m$$

Par définition, $u = w_0 = z_0$ est S -adjacent à $w_1 = z_1$, $w_1 = z_1$ est S -adjacent à $w_2 = z_2$, ..., $w_{n-1} = z_{n-1}$ est S -adjacent à $w_n = z_n = v$, $z_n = v = w'_0$ est S -adjacent à $w'_1 = z_{n+1}$,

$w'_1 = z_{n+1}$ est S -adjacent à $w_2 = z_{n+2}$, ..., $w'_{m-1} = z_{n+m-1}$ est S -adjacent à $w'_m = z_{n+m}$. Par conséquent, z_i est S -adjacent à z_{i+1} pour chaque $i \in \{0, \dots, n+m-1\}$ avec $z_0 = u$ et $z_{n+m} = w$ donc u est S -équivalent à w .

réflexivité. En posant $n = 0$, $w_0 = u$, $w_0 = u$, $w_n = u$ (!) et la propriété w_i S -adjacent à w_{i+1} est vraie (puisque l'on ne peut trouver de contre-exemple!) donc u est S -adjacent à u .

Commentaires : La difficulté de la question est essentiellement de connaître la définition d'une relation d'équivalence. Ensuite, vous n'êtes pas obligé d'être précis sur l'indexation mais bien faire comprendre au lecteur que vous maîtrisez ce qu'il faut établir et comment le faire (des phrases en français clairement exposées et précises suffisent). Il s'agit d'une question de niveau standard pour ce concours.

Q26. On procède par récurrence forte sur le nombre de noeuds de T . Si T ne contient qu'un noeud (il contient nécessairement b) alors $T = (o, b, o)$. Par construction de l'insertion (comme pour les arbres de recherche), on a

$$\begin{aligned} T &\leftarrow ac = ((o, b, o) \leftarrow a) \leftarrow c = (a, b, o) \leftarrow c = (a, b, c) \\ T &\leftarrow ca = ((o, b, o) \leftarrow c) \leftarrow a = (o, b, c) \leftarrow a = (a, b, c) \end{aligned}$$

donc $(T \leftarrow ac) = (T \leftarrow ca)$ et la propriété est vérifiée.

Supposons la propriété vérifiée pour tout arbre ayant strictement moins de n noeuds avec $n \geq 2$. Soit T un arbre avec n noeuds. Il est de la forme $T = (g, r, d)$ avec b apparaissant dans T . Pour traiter l'insertion, nous allons distinguer les différents placements de a et c donc la position de r par rapport au segment $[a, c]$.

Premier cas $a < r \leq c$. Par définition de l'insertion, on a :

$$\begin{aligned} T &\leftarrow ac = ((g, r, d) \leftarrow a) \leftarrow c = (g \leftarrow a, r, d) \leftarrow c = (g \leftarrow a, r, d \leftarrow c) \\ T &\leftarrow ca = ((g, r, d) \leftarrow c) \leftarrow a = (g, r, d \leftarrow c) \leftarrow a = (g \leftarrow a, r, d \leftarrow c) \end{aligned}$$

donc $T \leftarrow ac = T \leftarrow ca$.

Deuxième cas $c < r$. Comme $a < b \leq c$, on peut affirmer que $a < b \leq c < r$ donc b apparaît dans l'arbre g . Cet arbre ayant strictement moins de n noeuds, on a par hypothèse de récurrence $g \leftarrow ac = g \leftarrow ca$. Par définition de l'insertion, on a :

$$\begin{aligned} T &\leftarrow ac = ((g, r, d) \leftarrow a) \leftarrow c = (g \leftarrow a, r, d) \leftarrow c = ((g \leftarrow a) \leftarrow c, r, d) \\ &= (g \leftarrow ac, r, d) = (g \leftarrow ca, r, d) = ((g \leftarrow c) \leftarrow a, r, d) \\ &= (g \leftarrow c, r, d) \leftarrow a = T \leftarrow ca. \end{aligned}$$

Troisième cas $r \leq a$. Comme $a < b \leq c$, on peut affirmer que $r \leq a < b \leq c$ donc b apparaît dans l'arbre d . Cet arbre ayant strictement moins de n noeuds, on a par hypothèse de récurrence $d \leftarrow ac = d \leftarrow ca$. Par définition de l'insertion, on a :

$$\begin{aligned} T &\leftarrow ac = ((g, r, d) \leftarrow a) \leftarrow c = (g, r, d \leftarrow a) \leftarrow c = (g, r, (d \leftarrow a) \leftarrow c) \\ &= (g, r, d \leftarrow ac) = (g, r, d \leftarrow ca) = (g, r, (d \leftarrow c) \leftarrow a) \\ &= (g, r, d \leftarrow c) \leftarrow a = T \leftarrow ca. \end{aligned}$$

Ainsi, la propriété est vraie au rang n ce qui achève la récurrence.

Commentaires : Question en pratique difficile pour ce concours et destinée en pratique aux bonnes copies. Néanmoins, les remarques de la question 24 s'y appliquent. Entraînez-vous à ce type de preuve (de niveau élémentaire).

Q27. Vérifions la propriété lorsque les mots sont S -adjacents. Le cas S -équivalent en résulte par itération immédiate. En effet, dans ce cas, il existe des mots $w_0 = u, w_1, \dots, w_n = u$ avec, pour tout i , w_i est S -adjacent à w_{i+1} donc $o \leftarrow w_i = o \leftarrow w_{i+1}$. Par conséquent, on obtient

$$o \leftarrow u = o \leftarrow w_0 = o \leftarrow w_n = o \leftarrow v.$$

Soient u et v deux mots S -adjacents. Par hypothèse, il existe trois lettres $a < b \leq c$ et des mots f_1, f_2, f_3 tels que

$$(u = f_1 b f_2 a c f_3 \text{ et } v = f_1 b f_2 c a f_3) \text{ ou } (u = f_1 b f_2 c a f_3 \text{ et } v = f_1 b f_2 a c f_3)$$

Traitons le premier cas, le deuxième cas en résulte immédiatement en échangeant u et v (ce qui est compatible avec la S -adjacence car elle est symétrique). Posons $m = f_1 b f_2$ et $m' = f_3$ alors

$$u = m a c m' \text{ et } v = m c a m'.$$

Par définition de l'insertion, on a :

$$\begin{aligned} o \leftarrow u &= o \leftarrow m a c m' = (o \leftarrow m a c) \leftarrow m' \\ &= o((o \leftarrow m) \leftarrow a c) \leftarrow m' \stackrel{\text{cf Q26}}{=} ((o \leftarrow m) \leftarrow c a) \leftarrow m' \\ &= (o \leftarrow m c a) \leftarrow m' = o \leftarrow m c a m' = o \leftarrow v. \end{aligned}$$

Commentaires : Question sélective car elle nécessite du candidat de savoir faire des preuves ET de maîtriser correctement les nouvelles notions introduits dans le sujet. En pratique, seules les meilleures copies fournissent une réponse convenable.

Q28.

- (a) Par construction, deux mots u et v sont S -adjacents alors ils ont le même nombre de lettres. Par conséquent, si les deux mots u et v sont S -équivalents alors ils ont le même nombre de lettres. Le mot w ayant n lettres, l'ensemble A est un sous-ensemble des mots à n lettres sur l'alphabet fini Σ . Comme ce dernier ensemble est fini, A est aussi fini.

Commentaires :

Question élémentaire mais toujours sélective car il s'agit de dénombrement (simple mais cela suffit à faire fuir beaucoup de candidats).

- (b) Par définition, un mot u est S -adjacent à w a un préfixe non vide commun à w ($f_1 b$ selon les notations du sujet) donc la première lettre de u et w sont identiques. Si u est S -équivalent à w , il existe des mots $w_0 = u, w_1, \dots, w_n = w$ avec, pour chaque i , w_i est S -adjacent à w_{i+1} donc w_i et w_{i+1} ont la même première lettre. En particulier, $w_0 = u$ et $w_n = w$ ont la même première lettre donc tous les éléments de A commencent par r .

Commentaires : Question qui demande un peu de compréhension par le candidat. Elle s'avère discriminante (mais sans excès).

- (c) Par définition, il existe deux entiers i et j avec $i < j$ et $r \in [a_j, a_i[$. Si on raisonne graphiquement (au moins par analogie avec les nombres), la suite $m \mapsto a_m$ commence au dessus de r à l'instant i et finit strictement en dessous à l'instant j . On va regarder la dernière fois où elle est strictement au dessus de r , on note k cet élément alors à l'instant suivant $k + 1$ (qui est avant j donc

$$k + 1 \leq j \Leftrightarrow k \leq j - 1 \leq n - 1 - 1 = n - 2),$$

elle se trouve en dessous de r donc $a_{k+1} < r \leq a_k$ c'est-à-dire $(k, k+1) \in N(a)$. On peut formaliser cela en posant

$$k = \max \{ \ell \in \{i, \dots, j-1\}, a_\ell \geq r \}.$$

Commentaires : Question difficile réussie uniquement par les bonnes copies.

- (d) D'après la question b, il existe un mot a' tel que $a = ra'$.

Premier cas. Toutes les lettres de a' sont strictement inférieures à r . On pose $u = a'$ et $v = \varepsilon$.

Deuxième cas. Toutes les lettres de a' sont supérieures à r . On pose $u = \varepsilon$ et $v = a'$.

Troisième cas. Il y a des lettres de a' strictement inférieures à r et d'autres supérieures à r . Les lettres strictement inférieures à r sont toutes avant celles supérieures à r sinon $N(a)$ seraient non vide. En effet, si a_i est une lettre strictement inférieure à r et qu'elle succède une lettre a_j supérieure à r , on a nécessairement $j < i$ puisque a_i succède à a_j et $a_j < r \leq a_i$ donc $(i, j) \in N(a)$. Ceci est absurde. Si on désigne par m la dernière position où se trouvent les lettres strictement inférieures à r alors $u = a_1 \cdots a_m$ est constitué de lettres strictement inférieures à r et $v = a_{m+1} \cdots a_{n-1}$ est constitué de lettres supérieures à r avec $a = ruv$.

Commentaires : Même remarque qu'à la question précédente.

- (e) Supposons que $N(a)$ soit non vide. D'après la question c, il existe un entier k tel que $(k, k+1) \in N(a)$ donc $a_{k+1} < r \leq a_k$. Les mots

$$a = a_0 \cdots a_k a_{k+1} \cdots a_{n-1} \text{ et } a' = a_0 \cdots a_{k-1} a_{k+1} a_k a_{k+2} \cdots a_{n-1}$$

sont S -adjacents (prendre

$$f_1 b f_2 = a_0 \cdots a_{k-1}, a = a_k, c = a_{k+1}, f_3 = a_{k+2} \cdots a_{n-1}$$

dans la définition de la S -adjacence). Par définition, $N(a')$ possède un élément de moins que $N(a)$ (puisque l'on a enlevé le couple $(k, k+1)$) donc $N(a')$ est strictement plus petit que $N(a)$. Ceci est absurde car $N(a)$ est le plus petit possible. Par conséquent, $N(a)$ est vide.

Comme l'ensemble A est fini, l'ensemble $\{N(a), a \in A\}$ l'est aussi donc il possède un plus petit élément (au sens du nombre d'éléments) qui est forcément vide. Pour cet élément a (tel que $N(a) = \emptyset$) qui appartient à A donc qui est S -équivalent à w , la question précédente montre que l'on peut l'écrire $a = ruv$ avec toutes les lettres de u qui sont strictement inférieures à r et celles de v supérieures à r .

Commentaires : Bravo aux candidats à ce concours qui ont donné une réponse satisfaisante. Ils furent certainement peu nombreux.

- Q29.** On procède par récurrence sur le nombre de lettres de w . Si w est vide alors $T = o \leftarrow \varepsilon$ donc $w_T = \varepsilon = w$ par définition de l'insertion. Supposons la propriété vérifiée pour tout mot de longueur strictement à n , où n est un entier non nul. Soit w un mot de longueur n et r sa première lettre. D'après la question 28, il existe deux mots u et v tels que w et ruv sont S -équivalents, toutes les lettres de u sont strictement inférieures à r et toutes les lettres de v sont supérieures à r . On pose $T = o \leftarrow w$ (arbre obtenu par l'insertion de w dans l'arbre vide). D'après la question 27, on a :

$$T = o \leftarrow ruv = ((o \leftarrow r)) \leftarrow uv = (o, r, o) \leftarrow uv = (o \leftarrow u, r, o \leftarrow v)$$

par définition de l'insertion et du fait que les lettres de u sont strictement inférieures à r , celles de v supérieures à r . Si on pose $g = o \leftarrow u$ et $d = o \leftarrow v$ alors la lecture préfixe de T nous donne

$$w_T = rw_gw_d$$

Or, les mots u et v ayant strictement moins de n lettres, par hypothèses de récurrence on peut affirmer que w_g est S -équivalent à u et w_d est S -équivalent à v . Il est alors aisé de voir que $rw_gw_d = w_T$ est S -équivalent à ruv (je laisse le lecteur faire la justification) et que ruv est S -équivalent à w donc w_T est S -équivalent à w . La propriété est prouvée au rang n , ce qui achève la récurrence.

Commentaires : *Bravo aux candidats à ce concours qui ont donné une réponse satisfaisante. Ils furent certainement peu nombreux.*

Q30. Soient w et w' deux mots, on pose $T = o \leftarrow w$ et $T' = o \leftarrow w'$.

Si w et w' sont S -équivalents alors, d'après la question 27, on a $T = T'$ donc w et w' ont la même classe sylvestre.

Si $T = T'$ alors, d'après la question précédente, w est S -équivalent à $w_T = w_{T'}$ et w' est S -équivalent à $w_{T'}$ donc w et w' sont S -équivalents.

Commentaires : *Bravo aux candidats à ce concours qui ont donné une réponse satisfaisante. Ils furent certainement peu nombreux.*

Q31. On obtient : *abbaba, abbbaa, babbaa, bababa, ababba, baabba.*

Commentaires : *Question visant à familiariser le candidat avec cette notion. Y prêter beaucoup d'attention (utile pour le codage futur). La réponse est tout ou rien.*

Q32.

```
let rec ajout_lettre a l =
  match l with
  | [] -> []
  | b::q -> (a::b)::(ajout_lettre a q) ;;
```

Commentaires : *Question de programmation sans difficulté particulière.*

Q33.

```
let rec shuffle u v =
  match u,v with
  | [],_ -> [v]
  | _,[] -> [u]
  | a::q, b::r -> (ajout_lettre a (shuffle q v))@(ajout_lettre
    b (shuffle u r)) ;;
```

Commentaires : *Question sélective car elle demande une assez bonne compréhension par le candidat de la notion de mélange.*

Q34.

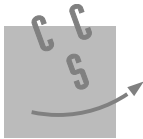
```
let rec shuffle_l l m =  
  match l ,m with  
  | [],_ -> []  
  | _,[] -> []  
  | a::q , b::r -> let p1 = shuffle a b and p2 = shuffle_l l r  
    and  
    p3 = shuffle_l q m in p1@p2@p3 ;;
```

Commentaires : *Même remarque qu'à la question précédente* .

Q35.

```
let rec sylvestre_T t =  
  match t with  
  | Vide -> [[]]  
  | Noeud(g,r,d) -> let l = sylvestre_T g and m = sylvestre_T d  
    in  
    ajout_lettre r (shuffle_l l m) ;;
```

Commentaires : *Question pour les meilleures copies qui ont été aptes à rédiger correctement les fonctions précédentes. Elle est accessible et un nombre significatif de candidats y ont certainement répondu correctement.*



CONCOURS CENTRALE-SUPÉLEC

Mathématiques 1

4 heures

Calculatrice autorisée

MP

2021

La loi du demi-cercle

Notations

- Pour tous entiers naturels p et q tels que $p \leq q$ on note $\llbracket p, q \rrbracket$ l'ensemble $\{i \in \mathbb{N} \mid p \leq i \leq q\}$.
- Pour tout entier naturel n tel que $n \geq 1$, on note $\mathcal{M}_n(\mathbb{R})$ l'ensemble des matrices à n lignes et n colonnes à coefficients réels, $\mathcal{S}_n(\mathbb{R})$ l'ensemble des matrices symétriques et à coefficients réels de taille n et $\mathcal{O}_n(\mathbb{R})$ l'ensemble des matrices réelles orthogonales de taille n .
- Pour toute matrice symétrique réelle $M \in \mathcal{S}_n(\mathbb{R})$ on note $\lambda_1(M) \geq \lambda_2(M) \geq \dots \geq \lambda_n(M)$ ses valeurs propres rangées dans l'ordre décroissant.
- On note $M^\top$ la transposée d'une matrice M .
- Pour tout $(i, j) \in \llbracket 1, n \rrbracket^2$, on note E_{ij} la matrice de la base canonique de $\mathcal{M}_n(\mathbb{R})$ dont tous les coefficients sont nuls sauf celui situé sur la i -ème ligne et la j -ième colonne qui vaut 1.
- Pour toute matrice $M \in \mathcal{M}_n(\mathbb{R})$, on note $\|M\|_F = \sqrt{\text{tr}(MM^\top)}$ sa norme euclidienne canonique.
- Dans tout le problème, on note $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé. Toutes les variables aléatoires considérées sont définies sur Ω .
- Étant donnée une variable aléatoire discrète X à valeurs réelles admettant une espérance, on note $\mathbb{E}(X)$ son espérance. Si X admet une variance, on note $\mathbb{V}(X)$ sa variance.

Problématique

En essayant d'expliquer la répartition des niveaux d'énergie des noyaux des atomes lourds, Eugene Wigner a été amené, dans les années 1950, à étudier le spectre de matrices symétriques réelles aléatoires de grande taille. La figure 1 montre un histogramme qui représente la répartition des valeurs propres d'une matrice symétrique réelle de taille $n = 2500$ constituée de variables aléatoires indépendantes identiquement distribuées, d'espérance nulle et de variance égale à 1.

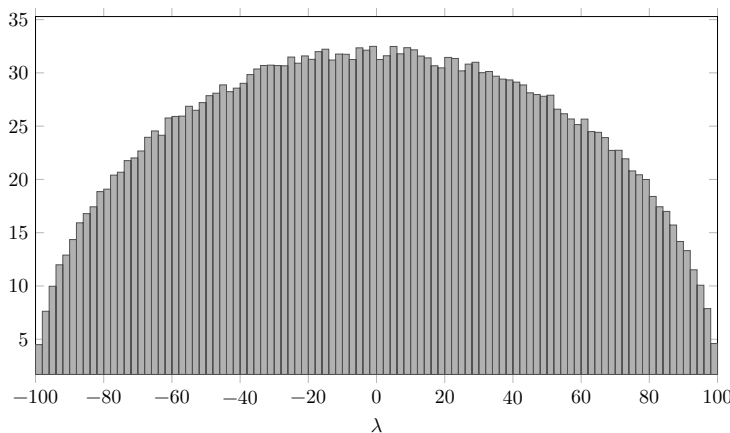


Figure 1

On voit que les valeurs propres se répartissent suivant un profil en demi-cercle de rayon $2\sqrt{n}$. En normalisant par un facteur $1/\sqrt{n}$ on obtient le profil d'un demi-cercle de rayon 2.

Plus précisément, on considère $(X_{ij})_{(i,j) \in (\mathbb{N}^*)^2}$ une famille de variables aléatoires discrètes réelles telles que

- pour tout $(i, j) \in (\mathbb{N}^*)^2$, $X_{ij} = X_{ji}$;
- les variables aléatoires X_{ij} sont de même loi, d'espérance nulle et de variance 1 ;
- pour $n \geq 1$, les variables aléatoires X_{ij} , pour $1 \leq i \leq j \leq n$, sont mutuellement indépendantes.

Pour tout $n \in \mathbb{N}^*$ et pour tout $\omega \in \Omega$, on note $M_n(\omega)$ la matrice $(X_{ij}(\omega))_{1 \leq i, j \leq n} \in \mathcal{M}_n(\mathbb{R})$.

Pour tout $\omega \in \Omega$, on note $\Lambda_{1,n}(\omega) \geq \dots \geq \Lambda_{n,n}(\omega)$ les valeurs propres de $\frac{1}{\sqrt{n}} M_n(\omega)$.

On définit ainsi, pour $n \in \mathbb{N}^*$ et $i \in \llbracket 1, n \rrbracket$, des variables aléatoires réelles discrètes $\Lambda_{i,n}$ sur $(\Omega, \mathcal{A}, \mathbb{P})$.

On se propose de montrer le résultat suivant :

— **Loi du demi-cercle** —

Pour toute fonction $f : \mathbb{R} \rightarrow \mathbb{R}$, continue et bornée,

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) \xrightarrow{n \rightarrow +\infty} \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx.$$

Dans la partie I, on établit une inégalité portant sur les valeurs propres d'un couple de matrices symétriques. La partie II est consacrée à la résolution d'un problème de dénombrement qui est utilisée dans la partie III où la loi du demi-cercle est démontrée pour des variables aléatoires uniformément bornées. Dans la partie IV, on établit la loi du demi-cercle dans le cas général en utilisant les résultats des parties I et III.

I Inégalité de Hoffman-Wielandt

I.A — Soient A et B deux matrices de $\mathcal{S}_n(\mathbb{R})$.

Q 1. Montrer que, pour tout M dans $\mathcal{M}_n(\mathbb{R})$ et pour tous P et Q dans $\mathcal{O}_n(\mathbb{R})$, on a $\|PMQ\|_F = \|M\|_F$.

Q 2. On note $D_A = \text{diag}(\lambda_1(A), \dots, \lambda_n(A))$ et $D_B = \text{diag}(\lambda_1(B), \dots, \lambda_n(B))$. Montrer qu'il existe une matrice orthogonale $P = (p_{i,j})_{1 \leq i, j \leq n}$ telle que $\|A - B\|_F^2 = \|D_A P - P D_B\|_F^2$.

Q 3. Montrer que

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

I.B — On note $\mathcal{B}_n(\mathbb{R})$ l'ensemble des matrices *bistochastiques* de $\mathcal{M}_n(\mathbb{R})$, c'est-à-dire l'ensemble des matrices $M = (m_{i,j})_{1 \leq i, j \leq n}$ dont tous les coefficients sont positifs ou nuls et tels que $\sum_{j=1}^n m_{i,j} = \sum_{j=1}^n m_{j,i} = 1$ pour tout $i \in \llbracket 1, n \rrbracket$.

$$\text{On note } f : \begin{cases} \mathcal{M}_n(\mathbb{R}) & \rightarrow \mathbb{R} \\ M & \mapsto \sum_{1 \leq i, j \leq n} m_{i,j} (\lambda_i(A) - \lambda_j(B))^2. \end{cases}$$

Q 4. Justifier que f admet un minimum sur $\mathcal{B}_n(\mathbb{R})$.

On se propose de montrer que ce minimum est atteint en la matrice identité.

Q 5. Soit $(i, j, k) \in \llbracket 1, n \rrbracket^3$ tel que $j \geq i$ et $k \geq i$. Montrer que, pour $M \in \mathcal{M}_n(\mathbb{R})$ et pour $x \in \mathbb{R}^+$,

$$f(M + xE_{ii} + xE_{jk} - xE_{ik} - xE_{ji}) - f(M) = 2x(\lambda_i(A) - \lambda_j(A))(\lambda_k(B) - \lambda_i(B)) \leq 0$$

Q 6. Soient $n \geq 2$ et $M = (m_{i,j})_{1 \leq i, j \leq n} \in \mathcal{B}_n(\mathbb{R})$ une matrice différente de l'identité. On note i le plus petit entier appartenant à $\llbracket 1, n \rrbracket$ tel que $m_{i,i} \neq 1$. Montrer qu'il existe une matrice $M' = (m'_{i,j})_{1 \leq i, j \leq n} \in \mathcal{B}_n(\mathbb{R})$ telle que $f(M') \leq f(M)$ et $m'_{j,j} = 1$ pour tout $j \in \llbracket 1, i \rrbracket$.

Q 7. En déduire que

$$\min\{f(M) \mid M \in \mathcal{B}_n(\mathbb{R})\} = f(I_n).$$

I.C –

Q 8. En déduire que

$$\forall (A, B) \in \mathcal{S}_n(\mathbb{R})^2, \quad \sum_{i=1}^n (\lambda_i(A) - \lambda_i(B))^2 \leq \|A - B\|_F^2.$$

II Dénombrement des mots bien parenthésés

Dans cette partie, on s'intéresse à des chaînes de caractères constituées uniquement des deux caractères parenthèse ouvrante et parenthèse fermante. On dit qu'un mot est *bien parenthésé* s'il commence par une parenthèse ouvrante et qu'à toute parenthèse ouvrante est associée une (unique) parenthèse fermante *qui lui est postérieure*. Par exemple le mot

$() (())$

est bien parenthésé. En revanche, le mot

$()) ()$

n'est pas bien parenthésé. Un mot bien parenthésé est ainsi forcément constitué d'un nombre pair de caractères, chaque parenthèse qui s'ouvre doit se refermer.

Pour tout entier $n \geq 1$, on note C_n le nombre de mots bien parenthésés de longueur $2n$. On pose par commodité $C_0 = 1$.

II.A –

Q 9. En énumérant les différents mots bien parenthésés de longueur 2, 4 et 6, montrer que $C_1 = 1$, $C_2 = 2$ et déterminer C_3 .

Q 10. Montrer que, pour tout entier naturel n , $C_n \leq 2^{2n}$. Que peut-on en déduire pour le rayon de convergence de la série entière $\sum C_k x^k$?

Q 11. Montrer par un raisonnement combinatoire que, pour tout entier $k \geq 1$,

$$C_k = \sum_{i=0}^{k-1} C_i C_{k-i-1}.$$

On peut remarquer qu'un mot bien parenthésé est forcément de la forme $(m)m'$ avec m et m' deux mots bien parenthésés, éventuellement vides.

II.B – Pour tout $x \in \left] -\frac{1}{4}, \frac{1}{4} \right[$, on pose $F(x) = \sum_{k=0}^{+\infty} C_k x^k$.

Q 12. Montrer que, pour tout $x \in \left] -\frac{1}{4}, \frac{1}{4} \right[$, $F(x) = 1 + x(F(x))^2$.

Q 13. Montrer que la fonction $f : \left] -\frac{1}{4}, \frac{1}{4} \right[\rightarrow \mathbb{R}$ ne s'annule pas.

$$\begin{array}{ccc} x & \mapsto & 2xF(x) - 1 \end{array}$$

Q 14. Déterminer, pour tout $x \in \left] -\frac{1}{4}, \frac{1}{4} \right[$, une expression de $F(x)$ en fonction de x .

Q 15. Déterminer le développement en série entière de la fonction $u \mapsto \sqrt{1-u}$. On écrira les coefficients sous la forme d'un quotient de factorielles et de puissances de 2.

Q 16. Montrer que, pour tout entier naturel n ,

$$C_n = \frac{(2n)!}{(n+1)!n!}.$$

III Loi du demi-cercle, cas uniformément borné

On suppose, uniquement dans cette partie, que les variables aléatoires X_{ij} sont uniformément bornées :

$$\exists K \in \mathbb{R} \quad \forall (i, j) \in (\mathbb{N}^*)^2 \quad |X_{ij}| \leq K.$$

III.A – Pour tout entier naturel k on pose

$$m_k = \frac{1}{2\pi} \int_{-2}^2 x^k \sqrt{4 - x^2} \, dx.$$

Q 17. Pour $k \in \mathbb{N}$, que vaut m_{2k+1} ?

Q 18. En utilisant le changement de variable $x = 2 \sin t$, calculer m_0 .

Q 19. À l'aide d'une intégration par parties, montrer que, pour tout entier naturel k ,

$$m_{2k+2} = \frac{2(2k+1)}{k+2} m_{2k}.$$

Q 20. En déduire que

$$m_k = \begin{cases} C_{k/2} & \text{si } k \text{ est pair,} \\ 0 & \text{si } k \text{ est impair.} \end{cases}$$

III.B – Soit k un entier naturel. On se propose de montrer que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = m_k,$$

où $\Lambda_{i,n}^k = (\Lambda_{i,n})^k$ est la puissance k -ième de $\Lambda_{i,n}$.

Q 21. Justifier que la variable aléatoire $\sum_{i=1}^n \Lambda_{i,n}^k$ admet une espérance et que

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = \frac{1}{n^{1+k/2}} \mathbb{E}(\text{tr}(M_n^k)) = \frac{1}{n^{1+k/2}} \sum_{(i_1, \dots, i_k) \in \llbracket 1, n \rrbracket^k} \mathbb{E}(X_{i_1 i_2} X_{i_2 i_3} \cdots X_{i_{k-1} i_k} X_{i_k i_1}).$$

On appelle *cycle* de longueur k à valeurs dans $\llbracket 1, n \rrbracket$, tout $(k+1)$ -uplet $\vec{i} = (i_1, i_2, \dots, i_k, i_1)$ d'éléments de $\llbracket 1, n \rrbracket$. Les éléments $i_1, \dots, i_k$ sont appelés *sommets* du cycle $\vec{i}$. On dit aussi que le cycle *passse* par ces sommets. On note $|\vec{i}|$ le nombre de sommets distincts du cycle $\vec{i}$.

On appelle *arêtes* du cycle $(i_1, i_2, \dots, i_k, i_1)$ les couples non ordonnés (l'ordre des deux éléments de chaque couple n'est pas significatif) $(i_1, i_2), (i_2, i_3), \dots, (i_k, i_1)$.

Par exemple $\vec{i} = (1, 3, 5, 3, 2, 2, 1)$ est un cycle de longueur 6 dans $\llbracket 1, 5 \rrbracket$. Les sommets de ce cycle sont les éléments 1, 2, 3 et 5, donc $|\vec{i}| = 4$. Les arêtes distinctes de ce cycle sont $(1, 3)$, $(3, 5)$, $(3, 2)$, $(2, 2)$ et $(2, 1)$. Les arêtes $(3, 5)$ et $(5, 3)$ sont les mêmes.

Q 22. Montrer que le nombre de cycles de longueur k dans $\llbracket 1, n \rrbracket$ passant par ℓ sommets distincts est inférieur ou égal à $n^\ell \ell^k$.

Q 23. En déduire que

$$\frac{1}{n^{1+k/2}} \sum_{\substack{\vec{i} \in \llbracket 1, n \rrbracket^k \\ |\vec{i}| \leq (k+1)/2}} |\mathbb{E}(X_{i_1 i_2} X_{i_2 i_3} \cdots X_{i_{k-1} i_k} X_{i_k i_1})| \xrightarrow{n \rightarrow +\infty} 0.$$

On classe les cycles de longueur k en trois sous-ensembles :

- l'ensemble $\mathcal{A}_k$, constitué des cycles où au moins une arête n'apparaît qu'une fois ;
- l'ensemble $\mathcal{B}_k$, constitué des cycles où toutes les arêtes apparaissent exactement deux fois ;
- l'ensemble $\mathcal{C}_k$, constitué des cycles où toutes les arêtes apparaissent au moins deux fois et il en existe au moins une qui apparaît au moins trois fois.

Q 24. Montrer que, si le cycle $(i_1, i_2, \dots, i_k, i_1)$ appartient à $\mathcal{A}_k$, alors

$$\mathbb{E}(X_{i_1 i_2} X_{i_2 i_3} \cdots X_{i_{k-1} i_k} X_{i_k i_1}) = 0.$$

Q 25. Montrer que, pour tout cycle $\vec{i}$ appartenant à $\mathcal{C}_k$, $|\vec{i}| \leq \frac{k+1}{2}$.

Q 26. Que peut-on dire de $\mathcal{B}_k$ si k est impair ? En déduire que $\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = 0$ dans ce cas.

On suppose dans la suite que k est pair et que $\vec{i} \in \mathcal{B}_k$ est un cycle passant par $\frac{k}{2} + 1$ sommets distincts. Autrement dit $|\vec{i}| = \frac{k}{2} + 1$.

On parcourt les arêtes de $\vec{i}$ dans l'ordre. À chaque arête de $\vec{i}$ on associe une parenthèse ouvrante si cette arête apparaît pour la première fois et une parenthèse fermante si elle apparaît pour la deuxième fois. Par exemple, au cycle $(1, 3, 2, 3, 1)$ correspond $(())$, au cycle $(1, 2, 1, 3, 1)$ correspond $()()$.

Q 27. Justifier que l'on obtient ainsi un mot bien parenthésé de longueur k .

Q 28. Dénombrer les cycles $\vec{i}$ qui correspondent à un mot bien parenthésé fixé.

Q 29. Déduire de ce qui précède que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = C_{k/2}.$$

III.C –

Q 30. En déduire que, pour tout polynôme $P \in \mathbb{R}[X]$,

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n P(\Lambda_{i,n}) \right) = \frac{1}{2\pi} \int_{-2}^2 P(x) \sqrt{4-x^2} dx.$$

III.D – Soit $A > 2$.

Q 31. Montrer que, pour tout $(p, q) \in \mathbb{N}^2$,

$$\mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \right) \leq \frac{1}{A^{p+2q}} \mathbb{E} \left(\sum_{i=1}^n |\Lambda_{i,n}|^{2(p+q)} \right).$$

Q 32. Montrer que

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \right) = 0.$$

Q 33. Soient f une fonction continue et bornée de $\mathbb{R}$ dans $\mathbb{R}$ et P un polynôme de degré p . Justifier qu'il existe une constante K telle que

$$\forall x \in \mathbb{R} \setminus]-A, A[, \quad |f(x) - P(x)| \leq K|x|^p.$$

Q 34. En déduire que

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |f - P|(\Lambda_{i,n}) \right) = 0.$$

III.E –

Q 35. Soit f une fonction continue et bornée de $\mathbb{R}$ dans $\mathbb{R}$. Montrer que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) = \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} \, dx.$$

IV Loi du demi-cercle, cas général

On revient au cas général. On note $\mathbb{1}_A$ la variable aléatoire indicatrice d'un événement A .

Pour tout $(i, j) \in (\mathbb{N}^*)^2$ et pour tout $C > 0$, on pose

$$\sigma_{ij}(C) = \sqrt{\mathbb{V}(X_{ij} \mathbb{1}_{|X_{ij}| \leq C})}.$$

Si $\sigma_{ij}(C) \neq 0$, on pose

$$\widehat{X}_{ij}(C) = \frac{1}{\sigma_{ij}(C)} (X_{ij} \mathbb{1}_{|X_{ij}| \leq C} - \mathbb{E}(X_{ij} \mathbb{1}_{|X_{ij}| \leq C})).$$

IV.A –

Q 36. Soit X une variable aléatoire discrète d'espérance finie. Montrer que

$$\mathbb{E}(X \mathbb{1}_{|X| \leq C}) \xrightarrow{C \rightarrow +\infty} \mathbb{E}(X).$$

Q 37. En déduire que

$$\lim_{C \rightarrow +\infty} \sigma_{ij}(C) = 1.$$

Q 38. Justifier que, pour C assez grand, les variables $\widehat{X}_{ij}(C)$ sont bien définies et qu'elles sont alors bornées, centrées, de variance 1 et qu'elles sont mutuellement indépendantes pour $1 \leq i \leq j$.

Q 39. Montrer que

$$X_{ij} - \widehat{X}_{ij}(C) = \left(1 - \frac{1}{\sigma_{ij}(C)}\right) X_{ij} + \frac{1}{\sigma_{ij}(C)} (X_{ij} \mathbb{1}_{|X_{ij}| > C} - \mathbb{E}(X_{ij} \mathbb{1}_{|X_{ij}| > C})).$$

Q 40. Montrer que

$$\lim_{C \rightarrow +\infty} \mathbb{E}((X_{ij} - \widehat{X}_{ij}(C))^2) = 0.$$

IV.B – Pour tout entier n tel que $n \geq 1$, on note $\widehat{M}_n(C) = (\widehat{X}_{ij}(C))_{1 \leq i, j \leq n}$. Pour tout $\omega \in \Omega$, on note $\widehat{\Lambda}_{1,n}(\omega) \geq \dots \geq \widehat{\Lambda}_{n,n}(\omega)$ les valeurs propres de $\frac{1}{\sqrt{n}} \widehat{M}_n(C)(\omega)$ rangées dans l'ordre décroissant.

On obtient ainsi des variables aléatoires réelles discrètes $\widehat{\Lambda}_{1,n}, \dots, \widehat{\Lambda}_{n,n}$ sur $(\Omega, \mathcal{A}, \mathbb{P})$.

Soit $f : \mathbb{R} \rightarrow \mathbb{R}$ une fonction K -lipschitzienne.

Q 41. Montrer que

$$\left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\widehat{\Lambda}_{i,n}) \right) \right| \leq \frac{K}{n} \mathbb{E}(\|M_n - \widehat{M}_n(C)\|_F).$$

Q 42. On suppose de plus f bornée. Montrer

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) \xrightarrow{n \rightarrow +\infty} \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} \, dx.$$

IV.C –

Q 43. Montrer la loi du demi-cercle dans le cas général.

• • • FIN • • •

SOLUTION

I. Inégalité de Hoffman-Wielandt

Question I.A

1. Soient $P, Q \in O_n(\mathbb{R})$ et $M \in \mathcal{M}_n(\mathbb{R})$. On a $QQ^T = I_n$ et donc

$$\|PMQ\|_F^2 = \text{Tr}(PMQQ^TM^TP^T) = \text{Tr}(PMM^TP^T)$$

Comme $P^T = P^{-1}$ et comme la trace est invariante par similitude, on a donc

$$\|PMQ\|_F^2 = \text{Tr}(MM^T) = \|M\|_F^2$$

Ainsi,

$$\boxed{\forall M \in \mathcal{M}_n(\mathbb{R}), \forall P, Q \in O_n(\mathbb{R}), \|PMQ\|_F = \|M\|_F}$$

Remarque : on peut aussi utiliser propriété $\text{Tr}(AB) = \text{Tr}(BA)$. Dans ce cas, il faut citer ce résultat et bien préciser avec quelles matrices A et B on l'utilise. Ce résultat est souvent mal employé, par exemple sous la forme $\text{Tr}(ABC) = \text{Tr}(ACB)$, résultat qui est bien sûr faux en général.

2. Par théorème spectral (orthodiagonalisabilité des matrices symétriques réelles), il existe $R, Q \in O_n(\mathbb{R})$ telles que

$$R^{-1}AR = D_A \quad \text{et} \quad Q^{-1}BQ = D_B$$

En utilisant la question 1 (multipliant à droite par Q et à gauche par R^{-1} qui sont des matrices orthogonales)

$$\|A - B\|_F^2 = \|RD_AR^{-1} - QD_BQ^{-1}\|_F^2 = \|D_AR^{-1}Q - R^{-1}QD_B\|_F^2$$

Comme $O_n(\mathbb{R})$ est un groupe, on a $P = R^{-1}Q \in O_n(\mathbb{R})$. On a donc montré que

$$\boxed{\exists P \in O_n(\mathbb{R}), \|A - B\|_F^2 = \|D_AP - PD_B\|_F^2}$$

Remarque : quand vous utilisez le théorème spectral, pensez à bien préciser que les matrices considérées sont symétriques ET réelles. La seule symétrie ne suffit bien sûr pas. Rappelons aussi, qu'une autre façon d'utiliser le théorème spectral est de dire qu'il existe une base orthonormée de vecteurs propres (correspondant aux colonnes de la matrice de passage ci-dessus). Selon les cas, on préfère travailler avec l'une ou l'autre des traductions.

3. On a tout d'abord $(D_AP)_{i,j} = \lambda_i(A)p_{i,j}$ et $(PD_B)_{i,j} = \lambda_j(B)p_{i,j}$. Par ailleurs, le carré de la norme est la somme des carrés des coefficients :

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

On a donc

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

Remarque : pour une matrice réelle M , le carré de la norme euclidienne s'écrit $\text{Tr}(M^T M)$. C'est une formule théorique utile mais en pratique, il faut bien sûr savoir que ce carré de norme est la somme des carrés des coefficients.

Question I.B

Remarque : on trouve très souvent en probabilité des matrices stochastiques dont la somme des coefficients par colonne vaut 1. Ces matrices apparaissent lors de l'utilisation de la formule des probabilités totales. Il est bon de connaître les propriétés spectrales des matrices stochastiques (ou bistochastiques) qui apparaissent très souvent dans les sujets. Ainsi (ce n'est qu'un exemple), une matrice stochastique par ligne à coefficients positifs n'a que des valeurs propres de module ≤ 1 et les valeurs propres de module 1 sont des racines de l'unité.

4. $\mathcal{B}_n(\mathbb{R})$ est un ensemble fermé comme intersection de fermés :

- les images réciproques du fermé $\mathbb{R}^+$ par les applications continues $M \mapsto m_{i,j}$
- les images réciproques du fermé $\{1\}$ par les applications continues $M \mapsto \sum_{j=1}^n m_{i,j}$ et $M \mapsto \sum_{j=1}^n m_{j,i}$.

De plus les coefficients d'une matrice bistochastique sont entre 0 et 1 donc

$$\forall M \in \mathcal{B}_n(\mathbb{R}), \|M\|_F \leq \sqrt{n^2} = n$$

et $\mathcal{B}_n(\mathbb{R})$ est donc borné. C'est finalement (on est en dimension finie) un compact. Comme f est continue (par exemple car linéaire en dimension finie), un théorème de compacité indique que f est bornée et atteint ses bornes. Ainsi

$$f \text{ admet un minimum sur } \mathcal{B}_n(\mathbb{R})$$

Remarque : quand vous montrez la compacité en justifiant les caractères fermé et borné, n'oubliez pas de donner l'argument de dimension finie. Il existe en dimension non finie des parties fermées et bornées non compactes (seule l'implication fermé et borné donne compact reste vraie).

5. On a directement (par linéarité de f)

$$\begin{aligned} f(M + xE_{i,i} + xE_{j,k} - xE_{i,k} - xE_{i,j}) - f(M) &= x(\lambda_i(A) - \lambda_i(B))^2 + x(\lambda_j(A) - \lambda_k(B))^2 \\ &\quad - x(\lambda_i(A) - \lambda_k(B))^2 - x(\lambda_j(A) - \lambda_i(B))^2 \\ &= 2x(-\lambda_i(A)\lambda_i(B) - \lambda_j(A)\lambda_k(B) \\ &\quad + \lambda_i(A)\lambda_k(B) + \lambda_j(A)\lambda_i(B)) \end{aligned}$$

On a donc

$$f(M + xE_{i,i} + xE_{j,k} - xE_{i,k} - xE_{i,j}) - f(M) = 2x(\lambda_i(A) - \lambda_j(A))(\lambda_k(B) - \lambda_i(B))$$

Le valeurs propres étant ordonnées dans l'ordre décroissant, la première parenthèse est positive et la seconde est négative. Pour $x \geq 0$, le produit est positif.

$$\text{Si } j, k \geq i \text{ et } x \geq 0, f(M + xE_{i,i} + xE_{j,k} - xE_{i,k} - xE_{i,j}) - f(M) \leq 0$$

6. Comme on suppose $M \in \mathcal{B}_n(\mathbb{R})$, si tous les $m_{i,i}$ valent 1, alors $M = I_n$.

En supposant $M \neq I_n$, il existe bien un plus petit i tel que $m_{i,i} \neq 1$ et M (étant bistochastique) s'écrit

$$M = \begin{pmatrix} I_{i-1} & 0 \\ 0 & N \end{pmatrix} \text{ avec } N \in \mathcal{B}_{n+1-i}(\mathbb{R})$$

On va transformer M par des opérations comme dans la question 5 en conservant le caractère bistochastique.

- Il existe $k > i$ tel que $m_{i,k} \neq 0$ (car sinon la somme sur la ligne i ne vaut pas 1) et de même un $j > i$ tel que $m_{j,i} \neq 0$. Notons $x = \min(m_{i,k}, m_{j,i})$. En transformant M en $M_1 = M + xE_{i,i} + xE_{j,k} - xE_{i,k} - xE_{j,i}$, on garde le caractère bistochastique (on a toujours des coefficients positifs et on n'a pas changé la somme par ligne ou colonne). De plus, $M_1 = \begin{pmatrix} I_{i-1} & 0 \\ 0 & N_1 \end{pmatrix}$ et N_1 possède au moins un zéro de plus sur la ligne i ou la colonne i (selon que $x = m_{i,k}$ ou $x = m_{j,i}$).
- Si $(N_1)_{i,i} = 1$, on a terminé. Sinon on peut recommencer pour faire apparaître un zéro de plus sur la ligne i ou la colonne i . En un nombre fini d'étape, on se sera ramené à une matrice bistochastique du même type avec le seul coefficient non nul sur la ligne i et la colonne i qui est en position (i, i) et vaut donc 1.

Remarque : une rédaction complète s'effectuerait par récurrence sur le nombre de zéros présents sur la ligne et la colonne i mais la formalisation récurrente ne me semble pas apporter de clarté au raisonnement. On pourrait aussi raisonner par l'absurde. Il est cependant indispensable de bien vérifier que les opérations effectuées gardent le caractère bistochastique de la matrice et de justifier que le processus finit par s'arrêter.

Comme les opérations de la question 5 diminuent la valeur de f , on a montré que

$$\boxed{\exists M' \in \mathcal{B}_n(\mathbb{R}), f(M') \leq f(M) \text{ et } \forall j \in \llbracket 1, i \rrbracket, m'_{j,j} = 1}$$

7. On montre par récurrence décroissante sur $k \in \llbracket 0, n \rrbracket$ que si $M \in \mathcal{B}_n(\mathbb{R})$ possède un bloc supérieur gauche égal à I_k , alors $f(I_n) \leq f(M)$.

- Le résultat est immédiat si $k = n$ puisqu'alors $M = I_n$.
- Supposons le résultat vrai jusqu'à un rang $k \geq 1$. Soit $M \in \mathcal{B}_n(\mathbb{R})$ possédant un bloc supérieur gauche égal à I_{k-1} .
Si M possède un bloc supérieur gauche égal à I_k , la conclusion est directe avec l'hypothèse au rang k .
Sinon, on est dans la situation de la question 6 avec $i = k$ et on peut obtenir M' telle que $f(M') \leq f(M)$ et à laquelle on peut appliquer le résultat au rang k . On a donc $f(I_n) \leq f(M') \leq f(M)$.

$f(I_n)$ minore $f(\mathcal{B}_n(\mathbb{R}))$ (pour $k = 0$, on peut considérer toutes les matrices de $\mathcal{B}_n(\mathbb{R})$) et est dans l'ensemble : c'est le minimum.

$$\boxed{\min\{f(M) \mid M \in \mathcal{B}_n(\mathbb{R})\} = f(I_n)}$$

Remarque : ici, on pourrait sans doute, comme en question 6, décrire un processus récurrent qui, à partir de $M \in \mathcal{B}_n(\mathbb{R})$, permet de construire une suite de matrices dans $\mathcal{B}_n(\mathbb{R})$ avec de plus en plus de 1 sur la diagonale et dont les images par f sont de plus en plus petites. On finit par tomber sur une matrice bistochastique à diagonale de 1 et donc égale à I_n . La formalisation de la récurrence, là encore, ne me semble pas apporter grand chose. Dans cette question, comme dans la précédente, il s'agit de convaincre le correcteur que l'on maîtrise le processus et ce n'est pas forcément en formalisant lourdement que l'on y parviendra.

Question I.C

8. Soit P la matrice de la question 2.

Lignes et colonnes de P sont des éléments de $\mathbb{R}^n$ de norme (euclidienne) valant 1. Ainsi, la matrice $M = (p_{i,j}^2)$ est bistochastique (les coefficients sont bien positifs).

Or, $\|A - B\|_F^2 = f(M)$ et donc $\|A - B\|_F^2 \geq f(I_n)$. Dans le calcul de $f(I_n)$, et comme le coefficient (i, j) de I_n est nul si $i \neq j$ et vaut 1 sinon, il ne reste que la somme des $(\lambda_i(A) - \lambda_i(B))^2$.

On a montré que

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

II. Dénombrement des mots bien parenthésés

Question II.A

9. $()$ est le seul mot bien parenthésé (on dira mot BP dans la suite) de taille 2.

Les mots BP de taille 4 sont $()()$ et $((()))$.

Les mots BP de taille 6 sont $()()()$, $()((()))$, $((())())$, $((()()))$ et $((()))()$.

$$C_1 = 1, C_2 = 2, C_3 = 5$$

10. Il a 2^{2n} mots de taille $2n$ (puisque deux choix pour chaque caractère). Ainsi $C_n \leq 2^{2n}$.
Pour tout réel x , $|C_n x^n| \leq |4x|^n$ est borné si $|4x| \leq 1$ et donc (lemme d'Abel)

$$\sum (C_n x^n) \text{ est de rayon convergence plus grand que } 1/4.$$

Remarque : l'obtention d'une majoration du terme positif C_n doit immédiatement faire penser au lemme d'Abel (avec lequel on définit la notion de rayon de convergence). Très souvent, c'est la recherche du caractère borné de la suite $(a_n x^n)$ qui donne une minoration du rayon de convergence et c'est vraiment le réflexe de base à acquérir.

11. Un mot bien parenthésé commence par une parenthèse ouvrante et il est associé à celle-ci une unique parenthèse fermante. Le mot s'écrit donc $(m)m'$ avec m, m' qui sont bien parenthésés ou vides. Réciproquement, de tels mots sont bien parenthésés.

Remarque : l'énoncé faisait une remarque sur la forme des mots bien parenthésés. Sans le dire, il indiquait l'existence d'une bijection entre bons ensembles. Sans entrer dans le détail en explicitant une bijection, il est bon de développer l'indication donnée.

On va partitionner l'ensemble C_k des mots BP $(m)m'$ de longueur $2k$ selon la longueur $2i$ de m (m est de longueur paire) avec $i \in \llbracket 0, k-1 \rrbracket$.

Pour un m donné de longueur $2i$, m' est de longueur $2k - 2i - 2 = 2(k - i - 1)$ et il y a C_{k-i-1} choix pour m' . Il y en a C_i pour m et les choix sont indépendants. On a donc $C_i C_{k-i-1}$ tels mots. On note qu'avec la convention choisie pour C_0 , ceci est valable même si $i = 0$ ou $i = k - 1$.

Comme on a une partition, on somme ces quantités et on obtient

$$C_k = \sum_{i=0}^{k-1} C_i C_{k-i-1}$$

Remarque : on trouve là des techniques classiques de dénombrement. L'apparition d'une somme est l'indice d'une réunion disjointe. L'apparition d'un produit est celui de choix indépendants. Il faut insister sur ces adjectifs (disjointe et indépendants). En revanche, la formalisation en termes de bijection pour obtenir des cardinaux me semble inutile ici. Les entiers C_i s'appellent les nombres de Catalan. On les retrouve quand on cherche, par exemple, à dénombrer les arbres binaires ayant un nombre donné de noeuds. La façon de résoudre ce type de récurrence (questions suivantes) est à connaître.

Question II.B

12. Par théorème sur le produit de Cauchy de séries entières, et comme le rayon de convergence est plus grand que $1/4$, pour $|x| < 1/4$ on a

$$F(x)^2 = \sum_{n=0}^{\infty} (C * C)_n x^n$$

où $*$ est le produit de Cauchy et où on a donc (avec la question 11)

$$(C * C)_n = \sum_{i=0}^n C_i C_{n-i} = C_{n+1}$$

On en déduit que

$$xF(x)^2 = \sum_{n=0}^{\infty} C_{n+1} x^{n+1} = \sum_{k=1}^{\infty} C_k x^k = F(x) - C_0 x^0 = F(x) - 1$$

On a ainsi montré que

$$\forall |x| < \frac{1}{4}, F(x) = 1 + xF(x)^2$$

Remarque : l'énoncé donnant le résultat, il faut détailler les calculs et surtout bien préciser les résultats utilisés. Il y a deux résultats au programme sur le produit de Cauchy : celui avec les séries absolument convergentes et celui avec les séries entières (le second étant conséquence du premier). Dans le contexte de ce problème, c'est directement le résultat sur les séries entières que l'on invoque. De façon générale, il faut bien se ramener au cas traité dans le cours quand on utilise un produit de Cauchy, c'est à dire celui de séries définies à partir du rang 0.

13. Supposons, par l'absurde, qu'il existe $x \in]-1/4, 1/4[$ tel que $2xF(x) - 1 = 0$. On a alors $x \neq 0$ et $F(x) = \frac{1}{2x}$. La question précédente donne alors $\frac{1}{2x} = 1 + \frac{x}{4x^2}$ et donc $4x^2 = x$ ce qui est faux pour $|x| < 1/4$ et $x \neq 0$: on a une contradiction.

$$x \mapsto 2xF(x) - 1 \text{ ne s'annule pas sur }]-1/4, 1/4[$$

Remarque : le fait que le résultat est "négatif" (la fonction ne s'annule PAS), invite au raisonnement par l'absurde. Il faut annoncer le raisonnement et le conclure proprement. Notez aussi que quand on écrit une quantité (ici $\frac{1}{2x}$) on doit s'assurer qu'elle existe.

14. Soit $x \neq 0$ tel que $|x| < 1/4$. Les solutions de $xr^2 - r + 1 = 0$, d'inconnue r , sont $\frac{1 \pm \sqrt{1-4x}}{2x}$. On a donc $2xF(x) - 1 = \pm\sqrt{1-4x}$. Or, $x \mapsto 2xF(x) - 1$ est continue et ne s'annule pas. Sur chacun des intervalles $] -1/4, 0[$ et $]0, 1/4[$, elle garde donc un signe constant (théorème des valeurs intermédiaires). Comme la limite en 0^+ et 0^- de cette quantité vaut -1 , ce signe est négatif. On a donc

$$\forall x \in]-\frac{1}{4}, \frac{1}{4}[, F(x) = \frac{1-\sqrt{1-4x}}{2x} \text{ et } F(0) = 1$$

Remarque : il y a ici deux expressions possibles pour $F(x)$ pour un x donné, disons $F_1(x)$ et $F_2(x)$. On a donc

$$\forall x \in]-\frac{1}{4}, \frac{1}{4}[\setminus\{0\}, (F(x) = F_1(x) \text{ ou } F(x) = F_2(x))$$

ce qui n'est pas du tout la même phrase que

$$(\forall x \in]-\frac{1}{4}, \frac{1}{4}[\setminus\{0\}, F(x) = F_1(x)) \text{ ou } (\forall x \in]-\frac{1}{4}, \frac{1}{4}[\setminus\{0\}, F(x) = F_2(x))$$

Il faut trouver un argument pour dire que l'on est toujours dans la première situation OU toujours dans la seconde. Attention aux erreurs de logique.

15. On sait de façon générale que

$$\forall |u| < 1, (1-u)^\alpha = 1 + \sum_{n=1}^{\infty} \frac{\alpha(\alpha-1)\dots(\alpha-n+1)}{n!} (-u)^n$$

Dans le cas où $\alpha = 1/2$, on a

$$\alpha(\alpha-1)\dots(\alpha-n+1) = \frac{1}{2} \left(-\frac{1}{2}\right) \dots \left(-\frac{2n-3}{2}\right) = \frac{(-1)^{n-1}}{2^n} \prod_{k=0}^{n-2} (2k+1)$$

On multiplie haut et bas par les nombres pairs de 2 à $2n-2$ pour obtenir

$$\frac{(-1)^{n-1}}{2^n} \frac{(2n-2)!}{2^{n-1}(n-1)!}$$

On a finalement

$$\forall |u| < 1, \sqrt{1-u} = 1 - \sum_{n=1}^{\infty} \frac{(2n-2)!}{2^{2n-1}(n-1)!n!} u^n$$

L'énoncé ne donne pas le résultat et il n'est donc pas nécessaire de trop détailler les calculs. Il faut cependant quelques jalons pour suivre ce calcul et surtout être attentif à ces calculs si on veut pouvoir conclure ensuite. Il semble nécessaire d'isoler le terme d'indice 0 puisque c'est le seul ayant un coefficient positif.

16. Pour x non nul et $|x| < 1/4$, on a donc

$$F(x) = \frac{1}{2x} \sum_{n=1}^{\infty} \frac{(2n-2)!}{2^{2n-1}(n-1)!n!} (4x)^n = \sum_{n=1}^{\infty} \frac{(2n-2)!}{(n-1)!n!} x^{n-1}$$

On change d'indice pour se ramener à $n \in \mathbb{N}$:

$$F(x) = \sum_{k=0}^{\infty} \frac{(2k!)}{k!(k+1)!} x^k$$

La relation reste vraie pour $x = 0$. Par unicité d'un DSE (le rayon est > 0) on peut conclure que

$$\forall n \in \mathbb{N}, C_n = \frac{(2n)!}{(n+1)!n!}$$

Remarque : il est crucial de savoir que $R > 0$ pour pouvoir identifier les coefficients des séries entières.

III. Loi du demi-cercle, cas uniformément borné

Question III.A

17. L'intégrale d'une fonction impaire sur un segment centré sur l'origine est nulle :

$$\forall k \in \mathbb{N}, m_{2k+1} = 0$$

Si on a sauté la majorité de la partie 2, il faut préciser le calcul. Par exemple effectuer le changement de variable $y = -x$ dans l'intégrale.

18. $t \mapsto \sin(2t)$ étant de classe C^1 sur $] -\pi/2, \pi/2[$ à dérivée ne s'annulant pas (et donc réalisant une bijection de l'intervalle dans son image $] -1, 1[$), le changement de variable proposé est licite et donne

$$m_0 = \frac{1}{2\pi} \int_{-\pi/2}^{\pi/2} \sqrt{4(1 - \sin^2(t))} (2 \cos(t)) dt = \frac{2}{\pi} \int_{-\pi/2}^{\pi/2} |\cos(t)| \cos(t) dt$$

Le cosinus est positif sur l'intervalle et $|\cos(t)| \cos(t) = \frac{\cos(2t)+1}{2}$. Le calcul se termine alors aisément et donne

$$m_0 = 1$$

Remarque : n'oubliez pas les différents aspects des changements de variables à savoir les hypothèses sur la fonction de changement de variable, les bornes à modifier et l'élément différentiel à prendre en compte. Notez aussi l'importance de l'argument de signe ici (dans une autre situation, on pourrait avoir $\sqrt{1 - \sin^2(t)} = -\cos(t)$).

19. On primitive $u'(x) = x\sqrt{4-x^2}$ en $u(x) = -\frac{1}{3}(4-x^2)^{3/2}$ et on dérive $v(x) = x^{2k+1}$ en $v'(x) = (2k+1)x^{2k}$. $u, v \in C^1([-2, 2])$ et on peut intégrer par parties pour obtenir

$$2\pi m_{2k+2} = \left[-\frac{1}{3}(4-x^2)^{3/2} x^{2k+1} \right]_{-2}^2 + \frac{2k+1}{3} \int_{-2}^2 (4-x^2)^{3/2} x^{2k} dx$$

En écrivant que $(4-x^2)^{3/2} x^{2k} = \sqrt{4-x^2} (4x^{2k} - x^{2k+2})$, on a donc

$$m_{2k+2} = \frac{2k+1}{3} (4m_{2k} - m_{2k+2})$$

ou encore, après simplification,

$$m_{2k+2} = \frac{2(2k+1)}{k+2} m_{2k}$$

Remarque : l'énoncé donne le résultat et on ne peut se contenter de le recopier. C'est pourquoi nous avons insisté sur les calculs.

20. Montrons par récurrence que $m_{2k} = C_k$.

- C'est vrai si $k = 0$ ($m_0 = C_0 = 1$)
- Supposons le résultat vrai jusqu'à un rang k . On a alors

$$m_{2k+2} = \frac{2(2k+1)}{k+2} C_k = \frac{2(2k+1)}{k+2} \frac{(2k)!}{(k+1)!k!} = \frac{(2k+2)!}{(k+2)!(k+1)!} = C_{k+1}$$

Comme m_{2k+1} est nul, on a finalement

$$m_k = \begin{cases} C_{k/2} & \text{si } k \text{ est pair} \\ 0 & \text{si } k \text{ est impair} \end{cases}$$

Question III.B

21. Pour une matrice diagonalisable (trigonalisable suffit), la trace est la somme des valeurs propres.

Par ailleurs, si $M \sim \text{diag}(\lambda_1, \dots, \lambda_n)$ alors $M^k \sim \text{diag}(\lambda_1^k, \dots, \lambda_n^k)$.

On en déduit ici que

$$\sum_{i=1}^n \Lambda_{i,n}^k = \text{Tr}((M_n/\sqrt{n})^k) = \frac{1}{n^{k/2}} \text{Tr}(M_n^k)$$

Or, les coefficients de M_n étant des fonctions bornées, il en va de même de ceux de M_n^k (à k fixé, la borne dépend de k) et donc de la trace de M_n^k . En tant que variable bornée,

$$\sum_{i=1}^n \Lambda_{i,n}^k \text{ admet une espérance}$$

Avec ce qui précède et la linéarité de l'espérance

$$\mathbb{E} \left(\sum_{i=1}^n \Lambda_{i,n}^k \right) = \frac{1}{n^{k/2}} \mathbb{E}(\text{Tr}(M_n^k))$$

et, toujours avec la linéarité de l'espérance

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = \frac{1}{n^{1+k/2}} \mathbb{E}(\text{Tr}(M_n^k))$$

D'après la formule du produit matriciel,

$$(A^k)_{i,j} = \sum_{i_2, \dots, i_k \in \llbracket 1, n \rrbracket} A_{i,i_2} A_{i_2,i_3} \dots A_{i_{k-1},i_k} A_{i_k,j}$$

La trace de A^k est donc la somme des $A_{i_1,i_2} A_{i_2,i_3} \dots A_{i_{k-1},i_k} A_{i_k,i_1}$ pour tous les $i_1, \dots, i_k$ de $\llbracket 1, n \rrbracket$. En particulier, l'espérance étant linéaire,

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = \frac{1}{n^{1+k/2}} \sum_{i_1, \dots, i_k \in \llbracket 1, n \rrbracket} \mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})$$

Remarque : on utilise rarement la formule théorique donnant le coefficient générique d'un produit de plus de deux matrices. La question 21 étant technique dans un sujet très long, il ne semble pas utile de la justifier.

22. Il y a $\binom{n}{\ell}$ façons de choisir ℓ sommets distincts. Une fois ces sommets choisis, un cycle de longueur k ne pouvant passer que par ces sommets (mais pas forcément tous) est caractérisé par un k -uplet de ces sommets (et non un $k+1$ -uplet puisque le premier sommet est obligatoirement répété à la fin) et il y a ℓ^k choix. Le nombre de cycles de longueur k passant par ℓ sommets distincts est donc plus petit que $\binom{n}{\ell} \ell^k$ (pas forcément égalité car on ne passe pas forcément par tous les sommets pour les cycles comptés). Comme $\binom{n}{\ell} \leq n^\ell$,

Il y a au plus $n^\ell \ell^k$ cycles de longueur k dans $\llbracket 1, n \rrbracket$ passant par ℓ sommets

23. Le produit de k des variables $X_{i,j}$ est en module plus petit que K^k . L'espérance d'un tel produit est donc de module plus petit que K^k . Avec la question 22, on a

$$\sum_{\substack{\vec{i} \in \llbracket 1, n \rrbracket^k \\ |\vec{i}| = \ell}} |\mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})| \leq K^k n^\ell \ell^k$$

ou encore

$$0 \leq \frac{1}{n^{1+k/2}} \sum_{\substack{\vec{i} \in \llbracket 1, n \rrbracket^k \\ |\vec{i}| = \ell}} |\mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})| \leq K^k n^{\ell-1-k/2} \ell^k$$

Pour tout $\ell \in [1, \frac{k+1}{2}]$, le terme précédent est de limite nulle quand $n \rightarrow +\infty$ car la puissance de n est $\leq -1/2 < 0$.

k étant fixé, en sommant pour les entiers ℓ entre 1 et $\frac{k+1}{2}$, on a encore une suite de limite nulle quand $n \rightarrow +\infty$, c'est à dire

$$\lim_{n \rightarrow +\infty} \frac{1}{n^{1+k/2}} \sum_{\substack{\vec{i} \in \llbracket 1, n \rrbracket^k \\ |\vec{i}| \leq (k+1)/2}} |\mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})| = 0$$

24. Pour un cycle $(i_1, \dots, i_k, i_1) \in \mathcal{A}_k$, l'une des arêtes (i_j, i_{j+1}) (avec $i_{k+1} = i_1$) n'apparaît qu'une fois. $X_{i_j, i_{j+1}}$ est, par lemme des coalitions, indépendante de toute fonction des AUTRES $X_{a,b}$ (ceci est vrai quand $i_j \leq i_{j+1}$ et $a \leq b$ mais les arêtes (u, v) et (v, u) sont identifiées et on se ramène à ce cas...). On a ainsi

$$\mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1}) = \mathbb{E}(X_{i_j, i_{j+1}}) \mathbb{E} \left(\prod_{t \neq j} X_{i_t, i_{t+1}} \right) = 0$$

puisque les $X_{a,b}$ sont centrées.

$$\text{Si } (i_1, \dots, i_k, i_1) \in \mathcal{A}_k, \text{ alors } \mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1}) = 0$$

Remarque : l'hypothèse d'indépendance est souvent cruciale en probabilité. L'un des outils permettant de justifier une telle indépendance est le lemme des coalitions. Quand on l'utilise, il est indispensable de le citer et d'en vérifier rigoureusement les hypothèses.

25. Considérons un cycle possédant a arêtes distinctes et regardons le nombre de sommets différents qu'il peut posséder. On part d'un premier sommet et à chaque fois que l'on ajoute une nouvelle arête, on découvre au plus un sommet. Quand on ajoute une arête déjà rencontrée, on ne découvre pas de nouveau sommet. Ainsi le cycle contient au plus $a + 1$ sommets différents.

Supposons $\vec{i} \in \mathcal{C}_k$. On a k arêtes au total. L'une est au moins présente 3 fois. Les x différentes autres étant toutes présentes au moins 2 fois, on a $k \geq 2x + 3$ et donc $x \leq \frac{k-3}{2}$. Il y a donc au plus $x + 1 = \frac{k-3}{2} + 1 = \frac{k-1}{2}$ arêtes différentes. Le nombre de sommets différents est donc au plus égal à $1 + \frac{k-1}{2}$. Ainsi,

$$\boxed{\forall \vec{i} \in \mathcal{C}_k, |\vec{i}| \leq \frac{k+1}{2}}$$

26. Dans un cycle de longueur k , il y a $k + 1$ arêtes (pas forcément différentes). Si k est pair, le nombre des arêtes est impair et il n'est pas possible que chaque arête apparaisse 2 fois. Ainsi

$$\boxed{\text{Si } k \text{ est impair, } \mathcal{B}_k = \emptyset}$$

Dans ce cas, dans la somme de la question 21, on peut se contenter des uplets $(i_1, \dots, i_k)$ tels que $(i_1, \dots, i_k, i_1)$ est dans $\mathcal{C}_k$: ceux dans $\mathcal{A}_k$ ont une contribution nulle (question 24) et il n'y en a pas dans $\mathcal{B}_k$.

Par inégalité triangulaire, le module de la somme qui reste est majoré par la quantité de la question 23 et est de limite nulle.

$$\boxed{\text{Si } k \text{ est impair, } \lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = 0}$$

27. Ce processus donne un mot ayant autant de lettres qu'il y a d'arêtes dans le cycle, c'est-à-dire k . A tout instant, le nombre des parenthèses ouvrantes est plus grand que celui des parenthèses fermantes (par définition, la première rencontre d'une arête a lieu avant la deuxième rencontre). Enfin, le nombre des parenthèses ouvrantes et fermantes sont égaux (par définition de $\mathcal{B}_k$).

Tout ceci nous donne donc un mot bien parenthésé.

$$\boxed{\text{Un mot associé à } \vec{i} \in \mathcal{B}_k \text{ avec } k \text{ pair est bien parenthésé}}$$

On a juste besoin ici que $\vec{i} \in \mathcal{B}_k$ et pas que le cycle passe par $1 + \frac{k}{2}$ sommets distincts.

28. On a ici $k = 2p$ et le nombre d'arêtes distinctes est exactement p (puisque toutes les arêtes apparaissent 2 fois). Le nombre maximum de sommets apparaissant dans le cycle est donc $p + 1 = \frac{k}{2} + 1$. On fait ici l'hypothèse que l'on est dans cette "configuration maximale". Il faut donc, si on reprend l'analyse de la question 25, qu'à partir du sommet d'origine, toute arête que l'on rencontre la première fois fasse découvrir un nouveau sommet.

Pour choisir un tel cycle, on doit choisir $p + 1$ sommets et l'ordre dans lequel on les découvre. Il y a $n(n - 1) \dots (n - p)$ tels choix. Si on fixe le mot parenthésé associé au cycle, on fixe les "positions" des arêtes qui apparaissent pour la première fois. Je prétends que ceci impose tout le cycle.

Prenons un exemple. On se place dans le cas $k = 10$ et on commence par choisir six sommets et l'ordre dans lequel on les trouve. Par exemple, 1,2,3,4,5,6. On choisit ensuite un mot bien parenthésé, par exemple $((())((())))$.

- Un cycle associé part de 1 puis découvre deux sommets. Il débute donc par (1, 2, 3).
- Le mot présente alors une parenthèse fermante et on doit emprunter une arête déjà vue. C'est forcément celle de 3 à 2 et le cycle se poursuit par (1, 2, 3, 2).
- La parenthèse ouvrante suivante nous fait découvrir 4 et on en est à (1, 2, 3, 2, 4).
- Le caractère suivant est) est il faut repasser par une arête déjà vue et c'est forcément celle qui va de 4 à 2. On en est à (1, 2, 3, 2, 4, 2).
- On découvre alors deux sommets. On en est à (1, 2, 3, 2, 4, 2, 5, 6).
- On doit alors repasser par trois arêtes connues et on doit obtenir (1, 2, 3, 2, 4, 2, 5, 6, 5, 2, 1)

Finalement,

le nombre de cycles correspondant à un mot bien parenthésé donné est $n(n - 1) \dots (n - k/2)$

29. On découpe la somme de la question 21 selon l'appartenance de $\vec{i}$ aux ensembles $\mathcal{A}_k$, $\mathcal{B}_k$ ou $\mathcal{C}_k$ comme expliqué en question 26. Et comme dans cette question, seule la partie relative à $\mathcal{B}_k$ a une limite non nulle. On veut donc en fait la limite de

$$\frac{1}{n^{1+k/2}} \sum_{\substack{\vec{i} \in [1, n]^k \\ \vec{i} \in \mathcal{B}_k}} \mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})$$

Avec la question 23, on peut même se limiter à des $\vec{i}$ tels que $\vec{i} > \frac{k+1}{2}$ c'est à dire à des $\vec{i}$ comme ceux considérés en question 28 :

$$\frac{1}{n^{1+k/2}} \sum_{\substack{\vec{i} \in [1, n]^k \\ |\vec{i}| = 1 + \frac{k}{2}, \vec{i} \in \mathcal{B}_k}} \mathbb{E}(X_{i_1, i_2} \dots X_{i_k, i_1})$$

Dans ce produit, les termes se regroupent par deux et on obtient des $X_{a,b}^2$ avec des $X_{a,b}$ indépendantes. L'espérance du produit est le produit des espérances des $X_{i,j}^2$ qui valent 1 ($1 = V(X_{i,j}) = \mathbb{E}(X_{i,j}^2)$ car $\mathbb{E}(X_{i,j}) = 1$). Il s'agit donc de chercher la limite de

$$\frac{\alpha_{n,k}}{n^{1+k/2}}$$

où $\alpha_{n,k}$ est le nombre d'éléments de $\mathcal{B}_k$ ayant $1 + \frac{k}{2}$ sommets différents. D'après la question précédente, ce nombre est égal à

$$\alpha_{n,k} = n(n - 1) \dots (n - k/2) C_{k/2} \sim n^{1+k/2} C_{k/2}$$

On en conclut que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n \Lambda_{i,n}^k \right) = C_{k/2}$$

Question III.C

30. Par linéarité de l'espérance et du passage à l'intégrale, il suffit de prouver le résultat quand P est un monôme c'est-à-dire quand $P = X^k$.
 Si k est impair, les questions 26 et 17 montrent l'égalité (les quantités sont nulles).
 Si k est pair, les questions 20 et 29 permettent de conclure (les quantités valent $C_{k/2}$).
 Par combinaisons linéaires (avec les linéarités évoquées) on a donc

$$\forall P \in \mathbb{R}[X], \quad \lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n P(\Lambda_{i,n}) \right) = \frac{1}{2\pi} \int_{-2}^2 P(x) \sqrt{4 - x^2} dx$$

Remarque : quand on veut prouver qu'une formule est vraie pour tout polynôme, il est souvent intéressant par le cas des monômes pour comprendre le fonctionnement. Si, en plus il y a linéarité, la conclusion est même alors immédiate. Cela évite d'écrire d'énormes formules avec plusieurs symboles $\sum$. Il faut toujours privilégier la lisibilité.

Question III.D

31. Comme $x \mapsto \frac{1}{x^{p+2q}}$ décroît sur $[2, +\infty[$, quand $|\Lambda_{i,n}| \geq A$, on a

$$|\Lambda_{i,n}|^p = \frac{|\Lambda_{i,n}|^{2p+2q}}{|\Lambda_{i,n}|^{p+2q}} \leq \frac{|\Lambda_{i,n}|^{2p+2q}}{A^{p+2q}}$$

On somme ces inégalités et on majore ensuite en ajoutant des quantités positives :

$$\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \leq \frac{1}{A^{p+2q}} \sum_{i=1}^n |\Lambda_{i,n}|^{2(p+q)}$$

Par croissance et linéarité de l'espérance (l'existence des espérances est acquise depuis la question 21 et de toutes les façons, les quantités sont positives et on pourrait se permettre les calcul dans $\mathbb{R} \cup \{+\infty\}$), on a donc

$$\mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \right) \leq \frac{1}{A^{p+2q}} \mathbb{E} \left(\sum_{i=1}^n |\Lambda_{i,n}|^{2(p+q)} \right)$$

Remarque : penser à justifier l'existence des espérance ou à donner l'argument du signe comme ci-dessus. Ce sera vraiment apprécié du correcteur.

32. On fixe $p \in \mathbb{N}$.

La question 19 donne $m_{2(k+1)} \leq 4m_{2k}$ et donc (récurrence simple) $m_{2k} \leq 4^k$.

On remarque que $0 \leq \frac{1+m_{2(p+q)}}{A^{p+2q}} \leq \frac{1}{A^{p+2q}} + \frac{4^{p+q}}{A^{p+2q}} = \frac{1}{A^{p+2q}} + \frac{4^p}{A^p} \left(\frac{2}{A}\right)^{2q}$ est de limite nulle quand $q \rightarrow +\infty$.

Soit alors $\varepsilon > 0$. Il existe un entier q tel que $\frac{1+m_{2(p+q)}}{A^{p+2q}} \leq \varepsilon$.

La question 30 (ou 29 au choix) donne

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \mathbb{E} \left(\sum_{i=1}^n |\Lambda_{i,n}|^{2(p+q)} \right) = m_{2(p+q)}$$

et il existe un rang n_0 tel que

$$\forall n \geq n_0, \frac{1}{n} \mathbb{E} \left(\sum_{i=1}^n |\Lambda_{i,n}|^{2(p+q)} \right) \leq 1 + m_{2(p+q)}$$

Avec le choix fait pour q , on a alors (avec l'inégalité de la question 31)

$$\forall n \geq n_0, \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \right) \leq \varepsilon$$

On a montré que

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p \right) = 0$$

33. $\frac{f(x)-P(x)}{x^p} = \frac{f(x)}{x^p} - \frac{P(x)}{x^p}$. Comme f est bornée, le premier terme est borné au voisinage des infinis (majoré en module par $|f(x)|$ pour $|x| \geq 1$). Comme P est de degré p , le second terme tend en module vers le module du coefficient dominant de P aux infinis.

Il existe donc $B > A$ et une constante c telle que

$$\forall |x| \geq B, \left| \frac{f(x) - P(x)}{x^p} \right| \leq c$$

Mais comme $x \mapsto \frac{f(x)-P(x)}{x^p}$ est continue sur le compact $[-B, -A] \cup [A, B]$, elle est bornée aussi sur ce compact. Elle l'est donc sur $\mathbb{R} \setminus]-A, A[$.

$$\boxed{\exists K, \forall |x| \geq A, |f(x) - P(x)| \leq K|x|^p \text{ si } P \text{ est un polynôme de degré } p}$$

Remarque : en notant M un majorant de $|f|$ sur $\mathbb{R} \setminus]-A, A[$, et en notant a_i les coefficients de P , on aurait pu remarquer que

$$|f(x) - P(x)| \leq \left(\frac{M}{|x|^p} + \sum_{i=0}^p |a_i| \right) |x|^p \leq \left(\frac{M}{A^p} + \sum_{i=0}^p |a_i| \right) |x|^p$$

pour obtenir une constante plus explicite.

34. On a donc

$$\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |f - P|(\Lambda_{i,n}) \leq K \sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |\Lambda_{i,n}|^p$$

On multiplie par $1/n$, on passe à l'espérance (opération croissante) et on utilise la question 32 et le théorème d'encadrement pour conclure que

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} |f - P|(\Lambda_{i,n}) \right) = 0$$

Question III.E

35. Soit f continue et bornée de $\mathbb{R}$ dans $\mathbb{R}$. On fixe $A > 2$.

Soit $\varepsilon > 0$. Par théorème de Weierstrass (approximation des fonctions continues sur un segment), il existe un polynôme P tel que $\|f - P\|_{\infty, [-A, A]} \leq \varepsilon$.

On écrit que

$$\mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx = I_n + J_n + K_n$$

$$I_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n (f - P)(\Lambda_{i,n}) \right)$$

$$J_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n P(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 P(x) \sqrt{4-x^2} dx$$

$$K_n = \frac{1}{2\pi} \int_{-2}^2 (f - P)(x) \sqrt{4-x^2} dx$$

- Comme $[-2, 2] \subset [-A, A]$, $|K_n| \leq \varepsilon m_0 = \varepsilon$.
- La question 30 donne un rang n_1 à partir duquel $|J_n| \leq \varepsilon$.
- On découpe I_n en deux morceaux :

$$I_n = \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| \geq A}} (f - P)(\Lambda_{i,n}) \right) + \frac{1}{n} \mathbb{E} \left(\sum_{\substack{1 \leq i \leq n \\ |\Lambda_{i,n}| < A}} (f - P)(\Lambda_{i,n}) \right) = I'_n + I''_n$$

La question 34 donne un rang n_1 à partir duquel $|I'_n| \leq \varepsilon$.

On a $|I''_n| \leq \varepsilon$ (le module de chaque terme dans la somme est majoré par ε).

Ainsi, pour $n \geq \max(n_1, n_2)$,

$$\left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx \right| \leq 4\varepsilon$$

et on a prouvé que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) = \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx$$

Remarque : il est beaucoup plus agréable d'utiliser la densité des polynômes "avec des ε " plutôt qu'avec des suites. On risque en effet, si on procède avec des suites, de se retrouver avec des interversions de limites.

IV. Loi du demi-cercle, cas général

Remarque : la suite du problème n'aura certainement été abordée que par un nombre très limité de candidats. Il ne semble pas judicieux de s'y lancer pour grappiller des points.

Question IV.A

36. On suppose X d'espérance finie. Comme $|X\mathbf{1}_{|X|\leq C}| \leq |X|$, $X\mathbf{1}_C$ est aussi d'espérance finie. Par formule de transfert,

$$\mathbb{E}(X\mathbf{1}_{|X|\leq C}) = \sum_{x \in X(\Omega)} x\mathbf{1}_{[0,C]}(|x|)\mathbb{P}(X=x)$$

Comme X est une variable aléatoire discrète elle ne prend qu'un nombre dénombrable ou fini de valeurs. On peut donc trouver une suite $(x_k)_{k \in \mathbb{N}}$ de réels distincts telle que $X(\Omega) \subset \{x_k, k \in \mathbb{N}\}$.

Comme les familles considérées sont sommables, on peut sommer les termes dans l'ordre que l'on veut et on a

$$\mathbb{E}(X\mathbf{1}_C) = \sum_{k=0}^{\infty} x_k \mathbf{1}_{[0,C]}(|x_k|)\mathbb{P}(X=x_k)$$

Notons $f_k : C \mapsto x_k \mathbf{1}_{[0,C]}(|x_k|)\mathbb{P}(X=x_k)$. Pour tout k , $f_k(C) \rightarrow x_k \mathbb{P}(X=x_k)$ quand $C \rightarrow +\infty$. De plus

$$\forall k, \forall C \geq 0, |f_k(C)| \leq |x_k| \mathbb{P}(X=x_k)$$

et le majorant est le terme général d'une série convergente. Ainsi, $\sum (f_k)$ converge normalement sur $\mathbb{R}^+$. Le théorème de double limite s'applique et donne

$$\lim_{C \rightarrow +\infty} \sum_{k=0}^{\infty} f_k(C) = \mathbb{E}(X)$$

et ainsi

$$\boxed{\lim_{C \rightarrow +\infty} \mathbb{E}(X\mathbf{1}_{|X|\leq C}) = \mathbb{E}(X)}$$

37. On a

$$\begin{aligned} \mathbb{V}(X_{i,j}\mathbf{1}_{|X_{i,j}|\leq C}) &= \mathbb{E}((X_{i,j}\mathbf{1}_{|X_{i,j}|\leq C})^2) - \mathbb{E}(X_{i,j}\mathbf{1}_{|X_{i,j}|\leq C})^2 \\ &= \mathbb{E}(X_{i,j}^2\mathbf{1}_{|X_{i,j}|\leq C}) - \mathbb{E}(X_{i,j}\mathbf{1}_{|X_{i,j}|\leq C})^2 \end{aligned}$$

On peut appliquer la question précédente avec $X_{i,j}$ d'une part et $X_{i,j}^2$ d'autre part pour obtenir (quand $C \rightarrow +\infty$, $C^2 \rightarrow +\infty$)

$$\lim_{C \rightarrow +\infty} \mathbb{V}(X_{i,j}\mathbf{1}_{|X_{i,j}|\leq C}) = \mathbb{E}(X_{i,j}^2) - \mathbb{E}(X_{i,j})^2$$

Ainsi (le passage à la racine carrée est continu)

$$\boxed{\lim_{C \rightarrow +\infty} \sigma_{i,j}(C) = \sigma(X_{i,j}) = 1}$$

38. Un terme de limite 1 fini par être plus grand que 1/2 et donc non nul. Ainsi $\sigma_{i,j}(C) \neq 0$ pour C assez grand et

$$\boxed{\widehat{X}_{i,j}(C) \text{ est bien définie pour } C \text{ grand}}$$

Il est immédiat que $\widehat{X}_{i,j}(C)$ est centrée réduite et le lemme des coalitions indique, puisque les $X_{i,j}$ sont indépendantes (pour les $i \leq j$), que

Les $\widehat{X}_{i,j}(C)$ pour $1 \leq i \leq j$ sont indépendantes

On a enfin

$$\forall \omega \in \Omega, |\widehat{X}_{i,j}(C)(\omega)| \leq \frac{|C| + \mathbb{E}(|X_{i,j}| \mathbf{1}_{|X_{i,j}| \leq C})}{\sigma_{i,j}(C)} \leq \frac{2C}{\sigma_{i,j}(C)}$$

et donc

Les $\widehat{X}_{i,j}(C)$ sont bornées

39. On a

$$X_{i,j} \mathbf{1}_{|X_{i,j}| \leq C} = X_{i,j} (1 - \mathbf{1}_{|X_{i,j}| > C})$$

et en passant à l'espérance

$$\mathbb{E}(X_{i,j} \mathbf{1}_{|X_{i,j}| \leq C}) = \mathbb{E}(X_{i,j}) - \mathbb{E}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C}) = -\mathbb{E}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C})$$

On en déduit que

$$\sigma_{i,j}(C) \widehat{X}_{i,j}(C) = X_{i,j} - X_{i,j} \mathbf{1}_{|X_{i,j}| > C} + \mathbb{E}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C})$$

et ainsi

$$X_{i,j} - \widehat{X}_{i,j}(C) = X_{i,j} \left(1 - \frac{1}{\sigma_{i,j}(C)} \right) + \frac{1}{\sigma_{i,j}(C)} (X_{i,j} \mathbf{1}_{|X_{i,j}| > C} - \mathbb{E}(\mathbf{1}_{|X_{i,j}| > C}))$$

40. Posons

$$A(C) = X_{i,j} \left(1 - \frac{1}{\sigma_{i,j}(C)} \right) \quad \text{et} \quad B(C) = \frac{1}{\sigma_{i,j}(C)} (X_{i,j} \mathbf{1}_{|X_{i,j}| > C} - \mathbb{E}(\mathbf{1}_{|X_{i,j}| > C}))$$

On a alors

$$(X_{i,j} - \widehat{X}_{i,j}(C))^2 = (A(C) + B(C))^2 \leq 2A(C)^2 + 2B(C)^2$$

On passe à l'espérance et on veut obtenir le fait que les deux termes du membre de droite tendent vers 0

- $\mathbb{E}(A(C)^2) = \left(1 - \frac{1}{\sigma_{i,j}(C)} \right)^2$ (car $1 = \mathbb{V}(X_{i,j}) = \mathbb{E}(X_{i,j}^2)$) et ce terme est de limite nulle.

- $\mathbb{E}(B(C)) = 0$ et donc $\mathbb{E}(B(C)^2) = \mathbb{V}(B(C)) = \frac{\mathbb{V}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C})}{\sigma_{i,j}(C)^2}$

or $\mathbb{V}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C}) = \mathbb{E}((X_{i,j} \mathbf{1}_{|X_{i,j}| > C})^2) - \mathbb{E}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C})^2 \leq \mathbb{E}(X_{i,j}^2 \mathbf{1}_{|X_{i,j}| > C})$
donc

$$0 \leq \mathbb{V}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C}) \leq \mathbb{E}(X_{i,j}^2) - \mathbb{E}(X_{i,j}^2 \mathbf{1}_{|X_{i,j}|^2 \leq C^2})$$

Ainsi d'après Q36 et le théorème d'encadrement : $\mathbb{V}(X_{i,j} \mathbf{1}_{|X_{i,j}| > C}) \xrightarrow{C \rightarrow +\infty} 0$ et donc aussi $\mathbb{E}(B(C)^2) \xrightarrow{C \rightarrow +\infty} 0$.

On a donc

$$\lim_{C \rightarrow +\infty} \mathbb{E}((X_{i,j} - \widehat{X}_{i,j}(C))^2) = 0$$

Question IV.B

41. Par linéarité et croissance de l'intégrale, on a

$$\left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\hat{\Lambda}_{i,n}) \right) \right| \leq \frac{1}{n} \mathbb{E} \left(\sum_{i=1}^n (|f(\Lambda_{i,n}) - f(\hat{\Lambda}_{i,n})|) \right)$$

Avec le caractère K -lipschitzien de f , ceci donne

$$\left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\hat{\Lambda}_{i,n}) \right) \right| \leq \frac{K}{n} \mathbb{E} \left(\sum_{i=1}^n |\Lambda_{i,n} - \hat{\Lambda}_{i,n}| \right)$$

L'inégalité de Cauchy-Schwarz dans $\mathbb{R}^n$ permet d'affirmer que

$$\sum_{i=1}^n |\Lambda_{i,n} - \hat{\Lambda}_{i,n}| \leq \sqrt{n} \sqrt{\sum_{i=1}^n (\Lambda_{i,n} - \hat{\Lambda}_{i,n})^2}$$

et avec la question 8

$$\sum_{i=1}^n (|\Lambda_{i,n} - \hat{\Lambda}_{i,n}|) \leq \sqrt{n} \left\| \frac{1}{\sqrt{n}} M_n - \frac{1}{\sqrt{n}} \widehat{M}_n(C) \right\|_F$$

On en déduit ainsi que

$$\left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\hat{\Lambda}_{i,n}) \right) \right| \leq \frac{K}{n} \mathbb{E}(\|M_n - \widehat{M}_n(C)\|_F)$$

42. Soit $\varepsilon > 0$. La question 40 montre que pour C assez grand (si les coefficients d'une matrice sont tous de limite quand $C \rightarrow +\infty$, il en est de même de la norme euclidienne)

$$\mathbb{E}(\|M_n - \widehat{M}_n(C)\|_F) \leq n\varepsilon$$

Les $\sigma_{i,j}(C)$ étant tous de limite 1, les quantités sont toutes plus grandes que $1/2$ pour C assez grand et pour de tels C , la question 38 montre que les $\widehat{X}_{i,j}(C)$ sont uniformément bornées (majorées en module par $4C$).

On peut donc choisir un C tel qu'on ait cette propriété de borne uniforme ET simultanément $\mathbb{E}(\|M_n - \widehat{M}_n(C)\|_F) \leq n\varepsilon$.

On fait un tel choix pour C et on a donc

$$\forall n, \left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\hat{\Lambda}_{i,n}) \right) \right| \leq K\varepsilon$$

On peut aussi utiliser la partie III et affirmer qu'il existe un rang n_0 tel que

$$\forall n \geq n_0, \left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\hat{\Lambda}_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx \right| \leq \varepsilon$$

On a donc finalement

$$\forall n \geq n_0, \left| \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx \right| \leq (K+1)\varepsilon$$

On a donc montré que

$$\lim_{n \rightarrow +\infty} \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) = \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx$$

Question IV.C

43. L'idée essentielle est de se ramener au segment $[-2, 2]$, sur lequel on peut approcher f par une fonction de classe C^1 et donc lipschitzienne.
L'objectif est de majorer en module la quantité

$$U_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n f(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 f(x) \sqrt{4-x^2} dx$$

On fixe $\varepsilon > 0$ que l'on choisit tel que $\varepsilon < 2$ (la raison en apparaît ensuite).

Par théorème de Weierstrass, il existe une fonction polynomiale P telle que $\|f - P\|_{\infty, [-2, 2]} \leq \varepsilon$.

On note g la fonction telle que $\forall x < -2$, $g(x) = P(-2)$, $\forall x \in [-2, 2]$, $g(x) = P(x)$ et $\forall x > 2$, $g(x) = P(2)$. On obtient ainsi une fonction g qui est lipschitzienne sur $\mathbb{R}$. De plus, en notant M une borne pour $|f|$, alors pour tout x on a $|f(x) - g(x)| \leq 2M + 2 = B$ (borne indépendante de ε).

On découpe U_n en trois morceaux : $U_n = V_n + W_n + X_n$ avec

$$V_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n (f - g)(\Lambda_{i,n}) \right)$$

$$W_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n g(\Lambda_{i,n}) \right) - \frac{1}{2\pi} \int_{-2}^2 g(x) \sqrt{4-x^2} dx$$

$$X_n = \frac{1}{2\pi} \int_{-2}^2 (g - f)(x) \sqrt{4-x^2} dx$$

- D'après la question précédente, $W_n \rightarrow 0$ et est plus petit en module que ε pour n assez grand, disons $n \geq n_1$.
- $|X_n| \leq \|f - g\|_{\infty, [-2, 2]} m_0 \leq \varepsilon$.
- Pour s'occuper de V_n , on va distinguer selon l'appartenance de $\Lambda_{i,n}$ à $[-2, 2]$. On écrit donc que

$$V_n = \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n (f - g)(\Lambda_{i,n}) \mathbf{1}_{|\Lambda_{i,n}| \leq 2} \right) + \mathbb{E} \left(\frac{1}{n} \sum_{i=1}^n (f - g)(\Lambda_{i,n}) \mathbf{1}_{|\Lambda_{i,n}| > 2} \right) = V'_n + V''_n$$

On a immédiatement

$$|V'_n| \leq \frac{1}{n} \sum_{i=1}^n \mathbb{E} (|(f - g)(\Lambda_{i,n})| \mathbf{1}_{|\Lambda_{i,n}| \leq 2}) \leq \varepsilon$$

On a vu que $f - g$ est bornée sur $\mathbb{R}$, majorée en module par B , et ainsi

$$|V''_n| \leq \frac{B}{n} \sum_{i=1}^n \mathbb{E} (\mathbf{1}_{|\Lambda_{i,n}| > 2}) = \frac{B}{n} \sum_{i=1}^n \mathbb{E} (\mathbf{1}_{\mathbb{R} \setminus [-2, 2]}(\Lambda_{i,n}))$$

Soit h la fonction nulle $[-2 + \varepsilon, 2 - \varepsilon]$, valant 1 hors de $[-2, 2]$ avec raccord affine de sorte à assurer la continuité. C'est une fonction bornée et lipschitzienne (le rapport dépend bien sûr de ε , en l'occurrence il vaut $1/\varepsilon$). De plus $\mathbf{1}_{\mathbb{R} \setminus [-2, 2]} \leq h$. Ainsi

$$|V_n''| \leq \frac{B}{n} \sum_{i=1}^n \mathbb{E}(h(\Lambda_{i,n}))$$

La question précédente montre que

$$\lim_{n \rightarrow +\infty} \frac{1}{n} \sum_{i=1}^n \mathbb{E}(h(\Lambda_{i,n})) = \frac{1}{2\pi} \int_{-2}^2 h(t) \sqrt{4-t^2} dt \leq \frac{2\varepsilon}{\pi}$$

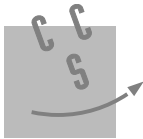
Pour n assez grand, disons $n \geq n_2$, le majorant de $|V_n''|$ est donc plus petit que $\frac{2B\varepsilon}{\pi}$.

On a finalement

$$\forall n \geq \max(n_1, n_2), |U_n| \leq \left(3 + \frac{2B}{\pi}\right) \varepsilon$$

et $U_n \rightarrow 0$.

La loi du demi-cercle est vraie dans le cas général



CONCOURS CENTRALE-SUPÉLEC

Mathématiques 2

4 heures

Calculatrice autorisée

MP

2021

Inégalités de Bernstein

Le but de ce problème est d'étudier les inégalités dites de Bernstein dans deux cadres différents.

La première partie s'intéresse à la démonstration de l'inégalité de Bernstein pour les polynômes et à certaines applications. La deuxième partie introduit la notion de transformée de Fourier et permet d'établir une inégalité de Bernstein pour des fonctions dont la transformée de Fourier vérifie certaines propriétés.

Les deux parties de ce sujet sont complètement indépendantes et peuvent être traitées dans l'ordre désiré.

I Inégalité polynomiale de Bernstein et applications

Dans cette partie,

- si $n \in \mathbb{N}$, on note $\mathbb{C}_n[X]$ le $\mathbb{C}$ -espace vectoriel des polynômes à coefficients complexes de degré inférieur ou égal à n ;
- si $n \in \mathbb{N}^*$, on note $\mathcal{S}_n$ le $\mathbb{C}$ -espace vectoriel des fonctions $f : \mathbb{R} \rightarrow \mathbb{C}$ vérifiant

$$\exists (a_0, \dots, a_n) \in \mathbb{C}^{n+1}, \quad \exists (b_1, \dots, b_n) \in \mathbb{C}^n, \quad \forall t \in \mathbb{R}, \quad f(t) = a_0 + \sum_{k=1}^n (a_k \cos(kt) + b_k \sin(kt)).$$

On remarque que les éléments de $\mathcal{S}_n$ sont des fonctions bornées ;

- si I est un intervalle non vide de $\mathbb{R}$ et si f est une fonction bornée de I dans $\mathbb{C}$, on note

$$\|f\|_{L^\infty(I)} = \sup_{x \in I} |f(x)|.$$

On admet que $f \mapsto \|f\|_{L^\infty(I)}$ définit une norme sur le $\mathbb{C}$ -espace vectoriel des fonctions bornées de I dans $\mathbb{C}$.

I.A – Polynômes de Tchebychev

On définit la suite de polynômes $(T_n)_{n \in \mathbb{N}}$ par $T_0 = 1$, $T_1 = X$ et $\forall n \in \mathbb{N}$, $T_{n+2} = 2XT_{n+1} - T_n$.

- Q 1.** Pour tout n dans $\mathbb{N}$, déterminer le degré de T_n , puis montrer que $(T_k)_{0 \leq k \leq n}$ est une base de $\mathbb{C}_n[X]$.
- Q 2.** Montrer que, pour tous $n \in \mathbb{N}$ et $\theta \in \mathbb{R}$, $T_n(\cos \theta) = \cos(n\theta)$.
- Q 3.** En déduire que, pour tous $n \in \mathbb{N}$ et $P \in \mathbb{C}_n[X]$, la fonction de $\mathbb{R}$ dans $\mathbb{C}$, $\theta \mapsto P(\cos \theta)$ est dans $\mathcal{S}_n$.
- Q 4.** Pour $n \in \mathbb{N}$, calculer $\|T_n\|_{L^\infty([-1,1])}$.
- Q 5.** Montrer que, pour tout $n \in \mathbb{N}$, $\|T'_n\|_{L^\infty([-1,1])} = n^2$.

On pourra commencer par établir que, pour tous $n \in \mathbb{N}$ et $\theta \in \mathbb{R}$, $|\sin(n\theta)| \leq n|\sin \theta|$.

I.B – Inégalité de Bernstein

Soit n un entier naturel non nul.

- Q 6.** Soit $A \in \mathbb{C}_{2n}[X]$, scindé à racines simples, et $(\alpha_1, \dots, \alpha_{2n})$ ses racines. Montrer que

$$\forall B \in \mathbb{C}_{2n-1}[X], \quad B(X) = \sum_{k=1}^{2n} B(\alpha_k) \frac{A(X)}{(X - \alpha_k)A'(\alpha_k)}. \quad (\text{I.1})$$

Soit P dans $\mathbb{C}_{2n}[X]$, et, pour tout $\lambda \in \mathbb{C}$, $P_\lambda(X) = P(\lambda X) - P(\lambda)$.

- Q 7.** Si $\lambda \in \mathbb{C}$, vérifier que $X - 1$ divise P_λ .

Pour tout λ dans $\mathbb{C}$, on note Q_λ le quotient de P_λ par $X - 1$:

$$Q_\lambda(X) = \frac{P(\lambda X) - P(\lambda)}{X - 1} \in \mathbb{C}_{2n-1}[X].$$

Q 8. Montrer que, pour tout λ dans $\mathbb{C}$, $Q_\lambda(1) = \lambda P'(\lambda)$.

On considère le polynôme $R(X) = X^{2n} + 1$. Pour k dans $\llbracket 1, 2n \rrbracket$, on note $\varphi_k = \frac{\pi}{2n} + \frac{k\pi}{n}$ et $\omega_k = e^{i\varphi_k}$.

Q 9. Montrer que

$$R(X) = \prod_{k=1}^{2n} (X - \omega_k).$$

Q 10. À l'aide de la formule (I.1), montrer que

$$\forall \lambda \in \mathbb{C}, \quad Q_\lambda(X) = -\frac{1}{2n} \sum_{k=1}^{2n} \frac{P(\lambda\omega_k) - P(\lambda)}{\omega_k - 1} \frac{X^{2n} + 1}{X - \omega_k} \omega_k$$

puis en déduire que

$$\forall \lambda \in \mathbb{C}, \quad \lambda P'(\lambda) = \frac{1}{2n} \sum_{k=1}^{2n} P(\lambda\omega_k) \frac{2\omega_k}{(1 - \omega_k)^2} - \frac{P(\lambda)}{2n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2}. \quad (\text{I.2})$$

Q 11. Montrer que

$$\forall \lambda \in \mathbb{C}, \quad \lambda P'(\lambda) = \frac{1}{2n} \sum_{k=1}^{2n} P(\lambda\omega_k) \frac{2\omega_k}{(1 - \omega_k)^2} + nP(\lambda).$$

On pourra appliquer l'égalité (I.2) au polynôme X^{2n} .

Soit maintenant f dans $\mathcal{S}_n$.

Q 12. Montrer qu'il existe $U \in \mathbb{C}_{2n}[X]$ tel que, pour tout $\theta \in \mathbb{R}$, $f(\theta) = e^{-in\theta} U(e^{i\theta})$.

Q 13. Vérifier que, pour tout $k \in \llbracket 1, 2n \rrbracket$, $\frac{2\omega_k}{(1 - \omega_k)^2} = \frac{-1}{2 \sin(\varphi_k/2)^2}$ et déduire des questions 11 et 12 que

$$\forall \theta \in \mathbb{R}, \quad f'(\theta) = \frac{1}{2n} \sum_{k=1}^{2n} f(\theta + \varphi_k) \frac{(-1)^k}{2 \sin(\varphi_k/2)^2}. \quad (\text{I.3})$$

Q 14. En déduire que

$$\forall \theta \in \mathbb{R}, \quad |f'(\theta)| \leq n \|f\|_{L^\infty(\mathbb{R})}. \quad (\text{I.4})$$

I.C – Quelques conséquences de l'inégalité (I.4)

Soit n un entier naturel non nul.

Q 15. Déduire des questions 3 et 14 que

$$\forall P \in \mathbb{C}_n[X], \quad \forall x \in [-1, 1], \quad |P'(x)\sqrt{1-x^2}| \leq n \|P\|_{L^\infty([-1,1])}.$$

Q 16. Montrer que

$$\forall Q \in \mathbb{C}_{n-1}[X], \quad |Q(1)| \leq n \sup_{-1 \leq x \leq 1} |Q(x)\sqrt{1-x^2}|.$$

On pourra considérer $f : \theta \mapsto Q(\cos \theta) \sin \theta$ et vérifier que $f \in \mathcal{S}_n$.

Q 17. Soit $R \in \mathbb{C}_{n-1}[X]$ et $t \in [-1, 1]$. Montrer que

$$|R(t)| \leq n \sup_{-1 \leq x \leq 1} |R(x)\sqrt{1-x^2}|.$$

On pourra considérer le polynôme $S_t(X) = R(tX)$.

Q 18. En déduire que, pour tout P dans $\mathbb{C}_n[X]$,

$$\|P'\|_{L^\infty([-1,1])} \leq n^2 \|P\|_{L^\infty([-1,1])}.$$

Q 19. Peut-il y avoir égalité dans l'inégalité précédente ?

II Inégalités de Bernstein et transformée de Fourier

Dans cette partie,

- pour $k \in \mathbb{N}$, on dit qu'une fonction f de $\mathbb{R}$ dans $\mathbb{C}$ est de classe $\mathcal{C}^k$ si elle est k fois dérivable sur $\mathbb{R}$, de dérivée k -ième continue sur $\mathbb{R}$ (si $k = 0$, f est continue) ; on dit que f est $\mathcal{C}^\infty$ si f est $\mathcal{C}^k$ pour tout $k \in \mathbb{N}$. On note $\mathcal{C}^k(\mathbb{R})$ (respectivement $\mathcal{C}^\infty(\mathbb{R})$) l'ensemble des fonctions de classe $\mathcal{C}^k$ (respectivement $\mathcal{C}^\infty$) sur $\mathbb{R}$;
- on note $L^1(\mathbb{R})$ l'ensemble des fonctions de $\mathbb{R}$ dans $\mathbb{C}$ continues et intégrables sur $\mathbb{R}$;
- pour $f \in L^1(\mathbb{R})$, on note $\|f\|_1 = \int_{-\infty}^{+\infty} |f(t)| dt$;
- on note $L^\infty(\mathbb{R})$ l'ensemble des fonctions de $\mathbb{R}$ dans $\mathbb{C}$ continues et bornées sur $\mathbb{R}$;
- pour $f \in L^\infty(\mathbb{R})$, on note $\|f\|_\infty = \sup_{x \in \mathbb{R}} |f(x)|$.

On admet que $L^1(\mathbb{R})$, $L^\infty(\mathbb{R})$ et $\mathcal{C}^k(\mathbb{R})$ ($k \in \mathbb{N}$) sont des sous-espaces vectoriels de $\mathbb{C}^{\mathbb{R}}$. On admet également que $f \mapsto \|f\|_1$ définit une norme sur $L^1(\mathbb{R})$ et que $f \mapsto \|f\|_\infty$ définit une norme sur $L^\infty(\mathbb{R})$. On dispose ainsi des espaces vectoriels normés $(L^1(\mathbb{R}), \|\cdot\|_1)$ et $(L^\infty(\mathbb{R}), \|\cdot\|_\infty)$.

II.A – Transformée de Fourier d'une fonction

Soit $f \in L^1(\mathbb{R})$. On appelle transformée de Fourier de f et on note $\hat{f}$ la fonction de $\mathbb{R}$ dans $\mathbb{C}$ telle que

$$\forall \xi \in \mathbb{R}, \quad \hat{f}(\xi) = \int_{-\infty}^{+\infty} f(x) e^{-ix\xi} dx.$$

Q 20. Montrer que, pour toute fonction $f \in L^1(\mathbb{R})$, $\hat{f}$ est définie et continue sur $\mathbb{R}$.

Q 21. Montrer que l'application $f \mapsto \hat{f}$ est une application linéaire continue de l'espace vectoriel normé $(L^1(\mathbb{R}), \|\cdot\|_1)$ dans l'espace vectoriel normé $(L^\infty(\mathbb{R}), \|\cdot\|_\infty)$.

Q 22. Soit $f \in L^1(\mathbb{R})$, $\lambda \in \mathbb{R}_+^*$ et soit g la fonction de $\mathbb{R}$ dans $\mathbb{C}$ telle que $g(x) = f(\lambda x)$ pour tout réel x . Montrer que $g \in L^1(\mathbb{R})$ et, pour tout réel ξ , exprimer $\hat{g}(\xi)$ à l'aide de $\hat{f}$, de ξ et de λ .

II.B – Produit de convolution

Si f et g sont deux fonctions continues de $\mathbb{R}$ dans $\mathbb{C}$ telles que, pour tout $x \in \mathbb{R}$, la fonction $t \mapsto f(t)g(x-t)$ soit intégrable sur $\mathbb{R}$, on appelle produit de convolution de f et g , et on note $f * g$, la fonction de $\mathbb{R}$ dans $\mathbb{C}$ telle que

$$\forall x \in \mathbb{R}, \quad (f * g)(x) = \int_{-\infty}^{+\infty} f(t)g(x-t) dt.$$

On suppose désormais et jusqu'à la fin de la sous-partie II.B que $f \in L^1(\mathbb{R})$ et $g \in L^\infty(\mathbb{R})$.

Q 23. Montrer que $f * g$ est définie sur $\mathbb{R}$ et que

$$\forall x \in \mathbb{R}, \quad (f * g)(x) = \int_{-\infty}^{+\infty} f(x-t)g(t) dt = (g * f)(x).$$

Q 24. Montrer que $f * g$ est bornée et que $\|f * g\|_\infty \leq \|f\|_1 \|g\|_\infty$.

Q 25. Soit $k \in \mathbb{N}$. Montrer que, si g est de classe $\mathcal{C}^k$ et si les fonctions $g^{(j)}$ sont bornées pour $j \in \llbracket 0, k \rrbracket$, alors $f * g$ est de classe $\mathcal{C}^k$ et $(f * g)^{(k)} = f * (g^{(k)})$.

Q 26. On suppose toujours que $f \in L^1(\mathbb{R})$ et $g \in L^\infty(\mathbb{R})$ et on suppose de plus que $g \in L^1(\mathbb{R})$ et $f * g \in L^1(\mathbb{R})$. En admettant que, pour tout ξ réel,

$$\int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} e^{-ix\xi} f(t)g(x-t) dt \right) dx \quad \text{et} \quad \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} e^{-ix\xi} f(t)g(x-t) dx \right) dt$$

existent et sont égales, montrer que $\widehat{f * g} = \hat{f} \hat{g}$.

II.C – Introduction d’une fonction plateau

On cherche dans cette sous-partie à construire une fonction réelle positive ρ , définie et de classe $\mathcal{C}^\infty$ sur $\mathbb{R}$, telle que $\rho(t) = 1$ pour tout $t \in [-1, 1]$ et $\rho(t) = 0$ pour tout $t \in \mathbb{R} \setminus [-2, 2]$.

Soit φ la fonction définie sur $\mathbb{R}$ par

$$\forall t \in \mathbb{R}, \quad \varphi(t) = \begin{cases} 0 & \text{si } t \leq 0, \\ e^{-1/t} & \text{sinon.} \end{cases}$$

Q 27. Montrer que φ est de classe $\mathcal{C}^\infty$ sur $\mathbb{R}$.

On pourra montrer que : $\forall k \in \mathbb{N}, \exists P_k \in \mathbb{R}[X], \forall t > 0, \varphi^{(k)}(t) = P_k(1/t)e^{-1/t}$.

Soit ψ la fonction définie sur $\mathbb{R}$ par

$$\forall t \in \mathbb{R}, \quad \psi(t) = \begin{cases} 0 & \text{si } t \notin]-1, 1[, \\ e^{1/(t^2-1)} & \text{sinon.} \end{cases}$$

Q 28. Montrer, en l’exprimant à l’aide de φ , que ψ est de classe $\mathcal{C}^\infty$.

Q 29. Soit θ l’unique primitive de ψ s’annulant en 0. Montrer que θ est de classe $\mathcal{C}^\infty$, constante sur $]-\infty, -1]$ (on note A cette constante) et constante sur $[1, +\infty[$ (on note B cette constante). Vérifier que $A \neq B$.

Q 30. Construire alors une fonction $\rho \in \mathcal{C}^\infty(\mathbb{R})$, constante égale à 1 sur $[-1, 1]$ et constante égale à 0 sur $\mathbb{R} \setminus [-2, 2]$.

II.D – Inégalités de Bernstein

On admet les formules suivantes, dites formules d’inversion de Fourier :

$$\text{— si } f \in L^1(\mathbb{R}) \text{ et si } \hat{f} \in L^1(\mathbb{R}), \text{ alors, pour tout } x \in \mathbb{R}, f(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{ix\xi} \hat{f}(\xi) d\xi;$$

$$\text{— si } \alpha \in L^1(\mathbb{R}), \text{ si } a \text{ est la fonction de } \mathbb{R} \text{ dans } \mathbb{C} : x \mapsto \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{ix\xi} \alpha(\xi) d\xi, \text{ et si } a \in L^1(\mathbb{R}), \text{ alors } \alpha = \hat{a}.$$

On remarque que ces résultats permettent d’affirmer que, si f et g sont deux fonctions continues telles que $f, g, \hat{f}$ et $\hat{g}$ sont intégrables et si $\hat{f} = \hat{g}$, alors $f = g$.

On considère toujours la fonction ρ définie à la question 30.

Soit r la fonction de $\mathbb{R}$ dans $\mathbb{C}$ telle que, pour tout réel x ,

$$r(x) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} e^{ix\xi} \rho(\xi) d\xi.$$

Q 31. Montrer que r est dérivable sur $\mathbb{R}$ et donner une expression de sa fonction dérivée (faisant éventuellement intervenir une intégrale).

Q 32. Montrer que $x \mapsto x^2 r(x)$ est bornée sur $\mathbb{R}$ et en déduire que r est intégrable et bornée sur $\mathbb{R}$.

On admet qu’en utilisant la même méthode, on montre que r' est intégrable et bornée sur $\mathbb{R}$.

Soit $\lambda > 0$ et soit $f \in L^1(\mathbb{R}) \cap \mathcal{C}^1(\mathbb{R})$ telle que $\hat{f} \in L^1(\mathbb{R})$ et telle que $\hat{f}$ soit nulle en dehors du segment $[-\lambda, \lambda]$. On note r_λ la fonction de $\mathbb{R}$ dans $\mathbb{C}$ telle que $r_\lambda(x) = r(\lambda x)$ pour tout réel x .

Q 33. On admet que $f * r_\lambda$ est intégrable. Montrer que $f = \lambda f * r_\lambda$.

Q 34. En déduire que, si $f \in L^\infty(\mathbb{R})$, il existe une constante $C \in \mathbb{R}_+^*$, indépendante de λ et de f , telle que

$$\|f'\|_\infty \leq C\lambda \|f\|_\infty.$$

• • • FIN • • •

SOLUTION

I. Inégalité polynomiale de Bernstein et applications.

I.A - Polynômes de Tchebychev

1. On montre par récurrence que $\boxed{\forall n \in \mathbb{N}, \deg(T_n) = n}$ (c'est l'hypothèse de récurrence au rang n).

- C'est vrai aux rang 0 et 1.
- Supposons le résultat vrai jusqu'à un rang $n \geq 1$ (on utilise une récurrence généralisée dite aussi récurrence forte). On a alors $T_{n+1} = 2XT_n - T_{n-1}$ qui est somme de deux polynômes de degrés $n+1$ et $n-1$. Comme ces degrés sont différents, T_{n+1} est de degré $\max(n+1, n-1) = n+1$.

$(T_k)_{0 \leq k \leq n}$ étant échelonnée en degré est libre. Elle contient $n+1$ éléments de $\mathbb{C}_n[X]$ qui est de dimension $n+1$. Ainsi,

$$\boxed{(T_k)_{0 \leq k \leq n} \text{ est une base de } \mathbb{C}_n[X]}$$

Remarque : dans cette première question de problème, la récurrence doit être menée avec précision : énoncé de l'hypothèse et type de récurrence. On aurait aussi pu utiliser une récurrence à deux crans mais il me semble plus simple de systématiquement utiliser une récurrence généralisée. En tous les cas, l'initialisation doit être suffisante pour conclure.

2. Procédons encore par récurrence. Cette fois, on veut montrer que

$$\boxed{\forall n \in \mathbb{N}, \forall \theta \in \mathbb{R}, T_n(\cos(\theta)) = \cos(n\theta)}$$

- C'est vrai aux rang 0 et 1.
- Supposons le résultat vrai jusqu'à un rang $n \geq 1$. On a alors

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

Comme $2\cos(a)\cos(b) = \cos(a-b) + \cos(a+b)$, le résultat au rang $n+1$ s'en déduit.

Remarque : on est encore au début du problème et il est indispensable de préciser les formules élémentaires (ici la formule de trigonométrie) que l'on utilise.

3. Comme $\mathcal{S}_n$ et $\mathbb{C}_n[X]$ sont des espaces vectoriels, il suffit de prouver le résultat pour les éléments d'une base de $\mathbb{C}_n[X]$ (et de conclure par combinaisons linéaires). Or, la question précédente prouve l'appartenance à $\mathcal{S}_n$ de $\theta \mapsto P(\cos(\theta))$ quand P est élément de la base $(T_0, \dots, T_n)$. Ainsi

$$\boxed{\forall n \in \mathbb{N}, \forall P \in \mathbb{C}_n[X], \theta \mapsto P(\cos(\theta)) \in \mathcal{S}_n}$$

Remarque : quand on veut prouver qu'un résultat est vrai pour tout polynôme, il est souvent bienvenu de commencer par montrer le résultat pour les éléments de la base canonique et (si c'est possible) de conclure par linéarité. Cela évite d'écrire le polynôme comme combinaison linéaire et de devoir se coltiner de grosses expressions. Ce qui est vrai avec la base canonique est bien sûr vrai avec une autre base et ici, il en est une qui s'impose. Il est bon d'avoir ces réflexes (on retrouvera cette technique plus loin dans le sujet).

4. Quand θ varie dans $\mathbb{R}$, $\cos(\theta)$ décrit tout $[-1, 1]$.

Ainsi la norme infinie de T_n sur $[-1, 1]$ est celle de $\theta \mapsto T_n(\cos(\theta))$ sur $\mathbb{R}$.

Puisque $|T_n(\cos(\theta))| = |\cos(n\theta)| \leq 1$ cette norme infinie est plus petite que 1.

De plus $T_n(1) = T_n(\cos(0)) = \cos(0) = 1$ donc la norme infinie est égale à 1 (c'est même un maximum).

$$\forall n \in \mathbb{N}, \|T_n\|_{L^\infty([-1,1])} = 1$$

Remarque : pour déterminer la borne supérieure d'un ensemble A , ici $A = \{T_n(x), x \in [-1, 1]\}$, on procède presque toujours en deux étapes. La première consiste à trouver un bon majorant α de A ce qui donne $\sup(A) \leq \alpha$. On montre ensuite que α est la borne supérieure soit en montrant que $\alpha \in A$ (on a alors un maximum comme ici) soit en trouvant une suite d'éléments de A qui converge vers α .

5. Prouvons par récurrence sur n que

$$\forall \theta \in \mathbb{R}, |\sin(n\theta)| \leq n |\sin(\theta)|$$

- C'est immédiat au rang 0.
- Supposons le résultat vrai jusqu'à un rang $n \geq 0$. On a alors, pour tout réel θ , en utilisant la formule donnant $\sin(a+b)$ ainsi que l'inégalité triangulaire,

$$|\sin((n+1)\theta)| \leq |\sin(n\theta)\cos(\theta)| + |\sin(\theta)\cos(n\theta)| \leq |\sin(n\theta)| + |\sin(\theta)| \leq (n+1)|\sin(\theta)|$$
 et le résultat est vrai au rang $n+1$.

Remarques.

- Il faut être très attentif quand on doit effectuer une majoration en module. C'est une traditionnelle source d'erreur.
- On peut aussi conclure sans récurrence en remarquant que $2i \sin(n\theta) = e^{in\theta} - e^{-in\theta}$, terme que l'on peut factoriser avec la formule donnant $x^n - y^n$. Le rapport de l'épreuve détaille cette méthode.

Par ailleurs, en dérivant la relation $T_n(\cos(\theta)) = \cos(n\theta)$, on obtient

$$\forall \theta \in \mathbb{R}, -\sin(\theta)T'_n(\cos(\theta)) = -n \sin(n\theta)$$

En combinant ceci,

$$\forall \theta \in \mathbb{R}, |\sin(\theta)T'_n(\cos(\theta))| \leq n^2 |\sin(\theta)|$$

On en déduit que si $\theta \neq 0[\pi]$ (pour pouvoir diviser par $|\sin(\theta)| > 0$), $|T'_n(\cos(\theta))| \leq n^2$.

Par continuité de $\theta \mapsto T'_n(\cos(\theta))$, ceci reste vrai sur $\mathbb{R}$.

La norme infinie de T'_n sur $[-1, 1]$ est donc plus petite que n^2 .

En utilisant une expression précédente, on a

$$\forall \theta \in]0, \pi/2], |T'_n(\cos(\theta))| = n \frac{|\sin(n\theta)|}{|\sin(\theta)|} \underset{\theta \rightarrow 0^+}{\sim} n \frac{n\theta}{\theta} = n^2$$

et ainsi (continuité) $|T'_n(1)| = n^2$. On a donc (le majorant trouvé est atteint et c'est un maximum et a fortiori une borne supérieure)

$$\forall n \in \mathbb{N}, \|T'_n\|_{L^\infty([-1,1])} = n^2$$

Remarque : Il faut être attentif aux cas particulier comme ci-dessus. Par ailleurs, on retrouve la méthode de détermination de borne supérieure évoquée plus haut.

I.B - Inégalité de Bernstein

Remarque préliminaire : à partir de cette question, l'énoncé fait apparaître des quotients de polynômes, i.e. des éléments de $\mathbb{C}(X)$. En réalité, les fractions écrites, comme après la question 7, se simplifient et les quantités sont des polynômes. L'énoncé ne dit rien de ce "tour de passe-passe". A moins de vraiment travailler dans $\mathbb{C}(X)$, comme dans l'une des méthodes de la question 6, il ne me semble pas indispensable de justifier cela (ce ne sera pas sanctionné). Le faire (par exemple en remarquant plus loin que la quantité Q_λ définie est effectivement un polynôme et pas une fraction rationnelle) sera malgré tout apprécié du correcteur.

6. Par hypothèse, et en notant c le coefficient dominant de A ,

$$A = c \prod_{j=1}^{2n} (X - \alpha_j)$$

On en déduit que

$$A' = c \sum_{k=1}^{2n} \prod_{\substack{1 \leq j \leq 2n \\ j \neq k}} (X - \alpha_j)$$

et en particulier

$$A'(\alpha_k) = c \prod_{\substack{1 \leq j \leq 2n \\ j \neq k}} (\alpha_k - \alpha_j)$$

Posons $L_k = \frac{A(X)}{(X - \alpha_k)A'(\alpha_k)}$. $L_k \in \mathbb{C}_{2n-1}[X]$ (puisque $A(X)$ est divisible par $X - \alpha_k$). Si $j \neq k$, on a $L_k(\alpha_j) = 0$. Le terme $X - \alpha_k$ se simplifiant dans $\frac{A}{X - \alpha_k}$, le calcul précédent donne $L_k(\alpha_k) = 1$. Ainsi

$$L_k(\alpha_j) = \delta_{j,k} \text{ (symbole de Kronecker)}$$

En particulier, pour tout polynôme B on a $B = \sum_{k=1}^{2n} B(\alpha_k)L_k$ qui est nul en tous les α_j . Quand $B \in \mathbb{C}_{2n-1}[X]$, c'est un polynôme de degré $\leq 2n - 1$ qui est donc nul (puisque'il a au moins $2n$ racine).

$$\forall B \in \mathbb{C}_{2n-1}[X], \quad B(X) = \sum_{k=1}^{2n} B(\alpha_k) \frac{A(X)}{(X - \alpha_k)A'(\alpha_k)}$$

Méthode alternative : on se donne $B \in \mathbb{C}_{2n-1}[X]$. $\frac{B}{A}$ est une fraction rationnelle de partie entière nulle (puisque $\deg(B) < \deg(A)$). Les pôles de la fraction sont simples (puisque A est à racines simples). Le cours indique que la décomposition en éléments simples s'écrit

$$\frac{B(X)}{A(X)} = \sum_{k=1}^{2n} \frac{c_k}{X - \alpha_k} \quad \text{avec} \quad c_k = \frac{B(\alpha_k)}{A'(\alpha_k)}$$

On en déduit alors la formule demandée dans $\mathbb{C}(X)$ en multipliant par $A(X)$.

Remarque : la formule donnant le coefficient relatif à un pôle simple dans une décomposition en éléments simples est à connaître. De même que le symbole $\delta_{j,k}$, parfois noté δ_j^k , qui vaut 0 si $j \neq k$ et 1 si $j = k$. Notez que dans un sujet, cette notation devrait toujours vous être rappelée.

7. On a $P_\lambda(1) = 0$ et donc $\boxed{(X - 1) \text{ divise } P_\lambda}$.
8. Comme 1 est racine de $P(\lambda X) - P(\lambda)$, la quantité Q_λ est bien un polynôme. On fixe $\lambda \in \mathbb{C}$. Les deux membres de l'égalité à prouver étant des expressions linéaires vis à vis de P , il suffit de vérifier la formule pour des P formant une base de $\mathbb{C}_{2n}[X]$, par exemple les X^k (deux application linéaires définies sur un espace vectoriel sont égales si et seulement si elles ont la même action sur une base). Or,

$$\frac{(\lambda X)^k - \lambda^k}{X - 1} = \lambda^k (X^{k-1} + X^{k-2} + \dots + 1)$$

et la valeur en 1 est $k\lambda^k$, qui est bien $\lambda(k\lambda^{k-1})$.

$$\boxed{\forall \lambda \in \mathbb{C}, Q_\lambda(1) = \lambda P'(\lambda)}$$

Attention : quand l'énoncé parle de $P'(\lambda)$, il s'agit du polynôme P' dans lequel on substitue λ à l'indéterminée X (ou, si on préfère, l'évaluation en λ de la fonction polynomiale associée au polynôme P). Ici, il ne peut être question de dériver la fonction polynomiale P car la variable considérée est complexe et le cours ne parle de dérivation que dans le cadre d'une variable réelle.

9. On remarque tout d'abord que

$$R(\omega_k) = e^{2in\varphi_k} + 1 = 0$$

et ω_k est racine de R . De plus

$$\varphi_k - \varphi_\ell = (k - \ell) \frac{\pi}{n}$$

Si $k, \ell \in \llbracket 1, 2n \rrbracket$, $-2n < k - \ell < 2n$ et donc $\varphi_k - \varphi_\ell \in]-2\pi, 2\pi[$ n'est nul que si $k = \ell$. On a ainsi $2n$ racines différentes pour R unitaire de degré $2n$ et donc

$$\boxed{R(X) = \prod_{k=1}^{2n} (X - \omega_k)}$$

Remarque : on pourrait, bien sûr, utiliser la méthode de factorisation du cours, ce qui revient à chercher les racines $2n$ -ième de -1 . Cependant, comme la formule est donnée par l'énoncé, il est aussi simple de faire, comme ci-dessus, une simple vérification. Le fait que les ω_k soient deux à deux distincts est crucial.

10. Si on applique (I.1) avec $A = R$ et $\alpha_k = \omega_k$ (qui sont bien distincts), on obtient, compte-tenu de $R'(\omega_k) = 2n\omega_k^{2n-1} = -\frac{2n}{\omega_k}$ (puisque $\omega_k^{2n} = -1$)

$$B(X) = -\frac{1}{2n} \sum_{k=1}^{2n} \frac{B(\omega_k)R(X)}{X - \omega_k} \omega_k$$

Ceci est vrai pour $B \in \mathbb{C}_{2n-1}[X]$ et en particulier pour Q_λ . Comme les ω_k sont différents de 1, l'expression de Q_λ donne alors

$$\forall \lambda \in \mathbb{C}, \quad Q_\lambda(X) = -\frac{1}{2n} \sum_{k=1}^{2n} \frac{P(\lambda\omega_k) - P(\lambda)}{\omega_k - 1} \frac{X^{2n} + 1}{X - \omega_k} \omega_k$$

Appliquons cette formule en $\lambda = 1$. Avec la question 8, on a alors

$$\lambda P'(\lambda) = -\frac{1}{2n} \sum_{k=1}^{2n} \frac{P(\lambda\omega_k) - P(\lambda)}{\omega_k - 1} \frac{2}{1 - \omega_k} \omega_k$$

Il reste à couper la somme en deux pour conclure que

$$\forall \lambda \in \mathbb{C}, \quad \lambda P'(\lambda) = \frac{1}{2n} \sum_{k=1}^{2n} P(\lambda\omega_k) \frac{2\omega_k}{(1 - \omega_k)^2} - \frac{P(\lambda)}{2n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2}$$

11. (I.2) avec $P = X^{2n}$ donne,

$$\forall \lambda \in \mathbb{C}, \quad 2n\lambda^{2n} = -\frac{1}{2n} \sum_{k=1}^{2n} \frac{2\lambda^{2n}\omega_k}{(1 - \omega_k)^2} - \frac{\lambda^{2n}}{2n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2} = -\frac{\lambda^{2n}}{n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2}$$

On en déduit, par exemple en donnant à λ la valeur 1, que

$$\frac{1}{2n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2} = -n$$

ce qui permet, après multiplication par $P(\lambda)$ de réécrire le second terme de (I.2) et de conclure que

$$\forall \lambda \in \mathbb{C}, \quad \lambda P'(\lambda) = \frac{1}{2n} \sum_{k=1}^{2n} P(\lambda\omega_k) \frac{2\omega_k}{(1 - \omega_k)^2} + nP(\lambda)$$

12. Soit $f \in \mathcal{S}_n$. Il lui est associé une suite $(a_k)_{0 \leq k \leq n}$ et une suite $(b_k)_{1 \leq k \leq n}$. Avec les formules d'Euler, on a

$$\begin{aligned} f(t) &= a_0 + \sum_{k=1}^n \left(\frac{a_k - ib_k}{2} e^{ikt} + \frac{a_k + ib_k}{2} e^{-ikt} \right) \\ &= e^{-int} \left(a_0 e^{int} + \sum_{k=1}^n \left(\frac{a_k - ib_k}{2} e^{i(k+n)t} + \frac{a_k + ib_k}{2} e^{i(n-k)t} \right) \right) \end{aligned}$$

Si on pose

$$U(X) = a_0 X^n + \sum_{k=1}^n \left(\frac{a_k - ib_k}{2} X^{k+n} + \frac{a_k + ib_k}{2} X^{n-k} \right)$$

on obtient un élément de $\mathbb{C}_{2n}[X]$ (quand $k \in \llbracket 1, n \rrbracket$, $k+n$ et $n-k$ sont entre 0 et $2n$) tel que $f(t) = e^{-int} U(e^{it})$.

$$\exists U \in \mathbb{C}_{2n}[X], \quad \forall \theta \in \mathbb{R}, f(\theta) = e^{-in\theta} U(e^{i\theta})$$

13. *Remarque préliminaire : l'énoncé donnant les formules à prouver, il faut un minimum détailler les calculs.*

On a $1 - \omega_k = e^{i\varphi_k/2}(e^{-i\varphi_k/2} - e^{i\varphi_k/2}) = -2ie^{i\varphi_k/2} \sin(\varphi_k/2)$ et ainsi

$$\frac{2\omega_k}{(1 - \omega_k)^2} = \frac{2e^{i\varphi_k}}{-4e^{i\varphi_k} \sin(\varphi_k/2)^2} = \frac{-1}{2 \sin(\varphi_k/2)^2}$$

Appliquons la question 11 au polynôme U . Avec l'expression ci-dessus, on obtient

$$\lambda U'(\lambda) = -\frac{1}{2n} \sum_{k=1}^{2n} U(\lambda \omega_k) \frac{1}{2 \sin(\varphi_k/2)^2} + nU(\lambda)$$

En particulier, pour $\lambda = e^{it}$, on obtient (puisque $f'(t) = -inf(t) + ie^{-int} e^{it} U'(e^{it})$)

$$\begin{aligned} -ie^{int}(f'(t) + inf(t)) &= -\frac{1}{2n} \sum_{k=1}^{2n} U(e^{i(t+\varphi_k)}) \frac{1}{2 \sin(\varphi_k/2)^2} + nU(e^{it}) \\ &= -\frac{1}{2n} \sum_{k=1}^{2n} e^{in(t+\varphi_k)} f(t + \varphi_k) \frac{1}{2 \sin(\varphi_k/2)^2} + ne^{int} f(t) \end{aligned}$$

Comme $e^{in\varphi_k} = i(-1)^k$, on conclut que

$$-if'(t) = -\frac{1}{2n} \sum_{k=1}^{2n} i(-1)^k f(t + \varphi_k) \frac{1}{2 \sin(\varphi_k/2)^2}$$

On a montré que

$$\boxed{\forall \theta \in \mathbb{R}, \quad f'(\theta) = \frac{1}{2n} \sum_{k=1}^{2n} f(\theta + \varphi_k) \frac{(-1)^k}{2 \sin(\varphi_k/2)^2}}$$

14. *Remarque préliminaire : ceci est une question qui synthétise/utilise plusieurs autres sans que l'énoncé le dise. Elle est répertoriée comme première question difficile selon l'énoncé. Elle va donc, sans doute, rapporter des points.*

D'après la question 11 avec $P = 1$, on a

$$\frac{1}{2n} \sum_{k=1}^{2n} \frac{2\omega_k}{(1 - \omega_k)^2} = -n$$

et avec la question 13, on en déduit que

$$\frac{1}{2n} \sum_{k=1}^{2n} \frac{1}{2 \sin(\varphi_k/2)^2} = n$$

Par inégalité triangulaire à partir de la question 13,

$$|f'(\theta)| \leq \frac{1}{2n} \sum_{k=1}^{2n} \|f\|_{L^\infty([-1,1])} \frac{1}{2 \sin(\varphi_k/2)^2} = n\|f\|_{L^\infty([-1,1])}$$

$$\boxed{\forall \theta \in \mathbb{R}, \quad |f'(\theta)| \leq n\|f\|_{L^\infty(\mathbb{R})}}$$

I.C - Quelques conséquences de l'inégalité (I.4)

15. Soit $P \in \mathbb{C}_n[X]$. Posons $f : t \mapsto P(\cos(t))$. La question 3 nous indique que c'est un élément de $\mathcal{S}_n$ et on peut donc lui appliquer la question 14. Mais on a

$$\forall t \in \mathbb{R}, f'(t) = -\sin(t)P'(\cos(t))$$

Si $x \in [-1, 1]$, on applique ceci avec $t = \arccos(x)$. Comme $\sin(\arccos(x)) = \sqrt{1-x^2}$ (car $0 \leq \sin(\theta) = \sqrt{1-\cos^2(\theta)}$ quand $\theta \in [0, \pi]$)

$$|-\sqrt{1-x^2}P'(x)| \leq n\|f\|_{L^\infty(\mathbb{R})} \leq nP_{L^\infty([-1,1])}$$

Le majorant est indépendant de x et ainsi

$$\boxed{\forall P \in \mathbb{C}_n[X], \quad \forall x \in [-1, 1], \quad |P'(x)\sqrt{1-x^2}| \leq n\|P\|_{L^\infty([-1,1])}}$$

Remarque : la relation $\sin(\arcsin(x)) = x$ est vraie pour tout $x \in [-1, 1]$. En revanche, $\arcsin(\sin(x)) = x$ n'est valable sur $[-\pi/2, \pi/2]$. A adapter pour les fonctions $\arccos$ ou $\arctan$. Par ailleurs, en règle générale on a $\sqrt{1-\cos(x)^2} = |\sin(x)|$ et il ne faut pas oublier les valeurs absolues ou préciser que l'on connaît le signe.

16. Soit $Q \in \mathbb{C}_{n-1}[X]$. Posons $f : \theta \mapsto Q(\cos(\theta))\sin(\theta)$. On sait déjà que $\theta \mapsto Q(\cos(\theta)) \in \mathcal{S}_{n-1}$ et s'écrit donc comme combinaison de $\cos(k\theta)$ et $\sin(k\theta)$ pour $k \in \llbracket 1, n-1 \rrbracket$ et d'une constante. Or,

$$\cos(k\theta)\sin(\theta) = \frac{1}{2}(\sin((k+1)\theta) - \sin((k-1)\theta))$$

$$\sin(k\theta)\sin(\theta) = -\frac{1}{2}(\cos((k+1)\theta) - \cos((k-1)\theta))$$

et $f(\theta)$ est donc combinaison de $\cos(j\theta)$ et $\sin(j\theta)$ pour $j \in \llbracket 0, n \rrbracket$ et de $\sin(\theta)$ (pour la constante multipliée par $\sin(\theta)$). $\theta \mapsto f(\theta)$ est donc un élément de $\mathcal{S}_n$.

Comme $f'(\theta) = Q(\cos(\theta))\cos(\theta) - \sin^2(\theta)Q'(\cos(\theta))$, on a $f'(1) = Q(1)$ ($\theta = 0$). Avec (I.4), on a donc

$$|Q(1)| \leq n\|f\|_{L^\infty(\mathbb{R})}$$

Remarquons alors que

$$\forall \theta \in \mathbb{R}, |f(\theta)| = |Q(\cos(\theta))\sin(\theta)| = |Q(x)|\sqrt{1-x^2} \quad \text{avec } x = \cos(\theta)$$

et donc $|f(\theta)|$ est plus petit que la borne supérieure des $|Q(x)|\sqrt{1-x^2}$ pour $x \in [-1, 1]$. Ainsi

$$\boxed{|Q(1)| \leq n \sup_{-1 \leq x \leq 1} |Q(x)\sqrt{1-x^2}|}$$

17. Soit $R \in \mathbb{C}_{n-1}[X]$ et $t \in [-1, 1]$. Considérons $S_t(X) = R(tX) \in \mathbb{C}_{n-1}[X]$. La question précédente utilisée avec ce polynôme donne

$$|R(t)| \leq n \sup_{-1 \leq x \leq 1} |R(tx)\sqrt{1-x^2}|$$

Pour tout $x \in [-1, 1]$, on a $1-x^2 \leq 1-t^2x^2$ et donc $\sqrt{1-x^2} \leq \sqrt{1-t^2x^2}$. Ainsi

$$\forall x \in [-1, 1], \quad |R(tx)\sqrt{1-x^2}| \leq |R(tx)\sqrt{1-(tx)^2}| \leq \sup_{-1 \leq y \leq 1} |R(y)\sqrt{1-y^2}|$$

On a ainsi montré que

$$\sup_{-1 \leq x \leq 1} \left| R(tx) \sqrt{1-x^2} \right| \leq \sup_{-1 \leq y \leq 1} |R(y) \sqrt{1-y^2}|$$

et on a donc

$$\boxed{|R(t)| \leq n \sup_{-1 \leq x \leq 1} |R(x) \sqrt{1-x^2}|}$$

Remarque : on notera ici la façon de prouver que $\sup(A) \leq \sup(B)$ et qui suit la démarche de la remarque en question 4. On montre que tout élément de A est plus petit que $\sup(B)$. Retenez de cela qu'il est facile de majorer une borne supérieure. Cette tactique se retrouve quand on veut prouver l'inégalité triangulaire pour la norme infinie (preuve qu'il vous est conseillée de bien connaître).

18. Soit $P \in \mathbb{C}_n[X]$, on peut appliquer ce qui précède à $P' \in \mathbb{C}_{n-1}[X]$:

$$\forall t \in [-1, 1], |P'(t)| \leq n \sup_{-1 \leq x \leq 1} |P'(x) \sqrt{1-x^2}|$$

Avec la question 15, on a donc

$$\forall t \in [-1, 1], |P'(t)| \leq n^2 \|P\|_{L^\infty([-1, 1])}$$

et ainsi

$$\boxed{\|P'\|_{L^\infty([-1, 1])} \leq n^2 \|P\|_{L^\infty([-1, 1])}}$$

19. D'une part, on $\|T_n\|_{L^\infty([-1, 1])} = 1$ (question 4). D'autre part, $\|T'_n\|_{L^\infty([-1, 1])} = n^2$ (question 5). Ainsi,

$$\boxed{\text{L'inégalité est une égalité quand } P = T_n \in \mathbb{C}_n[X]}$$

Remarque : plus simplement, on a égalité quand $P = 0$. Cette réponse était acceptée, même si elle est moins dans l'esprit du problème. Une question intéressante serait de trouver tous les cas d'égalité.

II Inégalités de Bernstein et transformée de Fourier

II.A - Transformée de Fourier d'une fonction

20. On utilise le théorème de continuité des intégrales à paramètres.

- Pour tout $\xi \in \mathbb{R}$, la fonction $x \mapsto f(x)e^{-ix\xi}$ est continue (par morceaux) sur $\mathbb{R}$.
- Pour tout $x \in \mathbb{R}$, la fonction $\xi \mapsto f(x)e^{-ix\xi}$ est continue sur $\mathbb{R}$.
- On a l'hypothèse de domination : $\forall x \in \mathbb{R}, \forall \xi \in \mathbb{R}, |f(x)e^{-ix\xi}| \leq |f(x)|$ et la fonction $|f|$ est continue et intégrable sur $\mathbb{R}$

Ainsi, la fonction $\hat{f}$ est définie continue sur $\mathbb{R}$.

$$\forall f \in L^1(\mathbb{R}), \hat{f} \in \mathcal{C}^0(\mathbb{R})$$

Remarque : attention ne pas majorer par des nombres complexes...

21. L'application $f \mapsto \hat{f}$ est linéaire (linéarité du passage à l'intégrale qui donne $\widehat{\lambda f + g} = \lambda \hat{f} + \hat{g}$). Soit $f \in L^1(\mathbb{R})$. On a

$$\forall \xi \in \mathbb{R}, |\hat{f}(\xi)| \leq \int_{\mathbb{R}} |f(x)e^{-ix\xi}| dx = \|f\|_1$$

et donc $\hat{f} \in L^\infty(\mathbb{R})$ avec

$$\|\hat{f}\|_\infty \leq \|f\|_1$$

Ceci montre que l'application linéaire $f \mapsto \hat{f}$ est continue et même 1 lispchitzienne (pour les normes proposées).

$$f \mapsto \hat{f} \text{ est continue de } (L^1(\mathbb{R}), \|\cdot\|_1) \text{ dans } (L^\infty(\mathbb{R}), \|\cdot\|_\infty)$$

Remarque : on a utilisé ici une caractérisation de la continuité des applications linéaires. On a même montré que la norme d'opérateur de l'application (sa "norme triple" associée aux normes choisies) est plus petite que 1.

22. f étant continue, g l'est aussi. De plus, le changement de variable linéaire $u = \lambda x$ donne

$$\int_0^a |g(x)| dx = \frac{1}{\lambda} \int_0^{\lambda a} |f(u)| du$$

et cette quantité admet une limite finie quand $a \rightarrow +\infty$ et aussi quand $a \rightarrow -\infty$. Il y a donc intégrabilité aux voisinages des infinis et

$$g \in L^1(\mathbb{R})$$

On peut alors écrire $\hat{g}(\xi)$ et le même changement de variable donne

$$\hat{g}(\xi) = \frac{1}{\lambda} \int_{-\infty}^{+\infty} f(u) e^{-i\xi u/\lambda} du$$

et ainsi

$$\hat{g}(\xi) = \frac{1}{\lambda} \hat{f}\left(\frac{\xi}{\lambda}\right)$$

Remarque : le programme stipule que les changements de variables simples, par exemple affines, n'ont pas besoin d'être justifiés. Attention à bien distinguer l'intégrabilité et l'existence d'intégrale. L'intégrabilité de g est exactement l'existence de l'intégrale de $|g|$ et ceci explique le calcul en deux étapes.

II.B - Produit de convolution

23. Soit $x \in \mathbb{R}$. La fonction $t \mapsto f(t)g(x-t)$ est continue sur $\mathbb{R}$. On a

$$\forall t \in \mathbb{R}, |f(t)g(x-t)| \leq |f(t)|\|g\|_\infty$$

f étant intégrable sur $\mathbb{R}$, la fonction $t \mapsto f(t)g(x-t)$ l'est aussi par théorème de comparaison. A fortiori, ceci assure la définition de $f * g$ sur $\mathbb{R}$.

Le changement de variable $u = x - t$ donne immédiatement

$$\forall x \in \mathbb{R}, (f * g)(x) = \int_{-\infty}^{+\infty} f(x-u)g(u) du = (g * f)(x)$$

Remarque : il faut bien distinguer les variables. Ici x est fixé et c'est l'existence de l'intégrale vis à vis de la variable t qui nous intéresse. On a d'ailleurs montré plus que l'existence de l'intégrale définissant $(f * g)(x)$ puisque l'on a une justification de l'intégrabilité, c'est à dire l'existence de l'intégrale du module (et PAS le module de l'intégrale, réfléchissez bien à la différence entre ces phrases).

24. L'inégalité de la question précédente entraîne que

$$\forall x \in \mathbb{R}, |(f * g)(x)| \leq \|g\|_\infty \|f\|_1$$

et ainsi

$$f * g \text{ est bornée et } \|f * g\|_\infty \leq \|f\|_1 \|g\|_\infty$$

25. On utilise le théorème de régularité des intégrales à paramètres.

- $\forall x \in \mathbb{R}, t \mapsto f(t)g(x-t)$ est continue (par morceaux) sur $\mathbb{R}$.
- $\forall t \in \mathbb{R}, x \mapsto f(t)g(x-t)$ est de classe C^k sur $\mathbb{R}$ de dérivée p -ième $x \mapsto f(t)g^{(p)}(x-t)$.
- $\forall x \in \mathbb{R}, \forall p \in [0, k], t \mapsto f(t)g^{(p)}(x-t)$ est continue (par morceaux) sur $\mathbb{R}$. Le théorème requiert l'intégrabilité pour $0 \leq p \leq k-1$ mais celle-ci est conséquence de la domination qui suit.
- $\forall x, t \in \mathbb{R}, \forall p \in [0, k], |f(t)g^{(p)}(x-t)| \leq \|g^{(p)}\|_\infty |f(t)|$ et ce majorant est intégrable sur $\mathbb{R}$.

Le théorème s'applique et donne que $f * g$ est de classe C^k avec

$$(f * g)^{(k)} = f * (g^{(k)})$$

Remarque : le programme permet d'utiliser directement le théorème pour montrer que la fonction est de classe C^k . Il impose uniquement la domination du module de la dérivée k -ième et l'intégrabilité des dérivées précédentes (y compris celle d'ordre 0 correspondant à la fonction de départ) ce qui est parfois oublié. Attention à ne pas utiliser de notations fantaisistes avec des dérivées partielles comme $\frac{\partial g(x-t)}{\partial x}$ ou même $\frac{\partial g}{\partial x}(x-t)$ qui n'a pas de sens mathématique (en particulier car g est une fonction d'une seule variable). Si vous voulez absolument introduire une fonction des deux variables, il faut lui donner un nom comme $h : (x, t) \mapsto f(t)g(x-t)$.

26. Par définition

$$\widehat{f * g}(\xi) = \int_{-\infty}^{+\infty} (f * g)(x) e^{-ix\xi} dx = \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} e^{-ix\xi} f(t)g(x-t) dt \right) dx$$

Avec le résultat admis,

$$\widehat{f * g}(\xi) = \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} e^{-ix\xi} f(t)g(x-t) dx \right) dt$$

On pose $u = x - t$ dans l'intégrale intérieure :

$$\widehat{f * g}(\xi) = \int_{-\infty}^{+\infty} \left(\int_{-\infty}^{+\infty} e^{-i(u+t)\xi} f(t)g(u) du \right) dt$$

et on peut "faire sortir" de l'intégrale les termes indépendants de la variable u

$$\widehat{f * g}(\xi) = \int_{-\infty}^{+\infty} \left(f(t) e^{-it\xi} \int_{-\infty}^{+\infty} e^{-iu\xi} g(u) du \right) dt = \int_{-\infty}^{+\infty} f(t) e^{-it\xi} \hat{g}(\xi) dt$$

Là encore $\hat{g}(\xi)$ peut sortir de l'intégrale et on obtient $\hat{f}(x)\hat{g}(x)$.

$$\boxed{\widehat{f * g} = \hat{f} \hat{g}}$$

II. C - Introduction d'une fonction plateau

27. φ est de classe $\mathcal{C}^\infty$ sur $\mathbb{R}^{+*}$ par théorèmes d'opération. On prouve par récurrence que

$$\forall k \in \mathbb{N}, \exists P_k \in \mathbb{R}[X], \forall t > 0, \varphi^{(k)}(t) = P_k(1/t) e^{-1/t}$$

- C'est vrai au rang 0 avec $P_0 = 1$.
- Supposons le résultat vrai jusqu'à un rang $k \geq 0$. On peut alors redériver et obtenir

$$\forall t > 0, \varphi^{(k)}(t) = \left(-\frac{1}{t^2} P'_k(1/t) + \frac{1}{t^2} P_k(1/t) \right) e^{-1/t}$$

$P_{k+1} = X^2(-P'_k + P_k)$ est un polynôme convenable au rang $k+1$.

On peut alors montrer que φ est de classe $\mathcal{C}^k$ sur $\mathbb{R}$ pour tout k en procédant par récurrence.

- Comme $e^{-1/t} \rightarrow 0$ quand $t \rightarrow 0^+$, φ est continue en 0. Elle est continue en tout point de $\mathbb{R}^*$ par théorèmes d'opération (sur un voisinage de $t_0 < 0$, elle est nulle et sur un voisinage de $t_0 > 0$, elle coïncide avec $t \mapsto e^{-1/t}$). φ est donc de classe $\mathcal{C}^0$ sur $\mathbb{R}$.
- Supposons le résultat vrai jusqu'à un rang $k \geq 0$. $\psi = \varphi^{(k)}$ est donc continue sur $\mathbb{R}$. Par théorèmes d'opération, φ est de classe $\mathcal{C}^{k+1}$ sur $\mathbb{R}^*$ et ψ est donc de classe $\mathcal{C}^1$ sur $\mathbb{R}^*$.

Avec la première partie de la question, $\forall t > 0, \psi'(t) = \varphi^{(k+1)}(t) \rightarrow 0$ quand $t \rightarrow 0^+$ (croissances comparées). Par théorème de limite de la dérivée, ψ est dérivable à droite en 0 et sa dérivée à droite est nulle.

ψ est nulle sur $\mathbb{R}^{*-}$ et donc sur $\mathbb{R}^-$ par continuité. Elle est donc dérivable à gauche en 0 et sa dérivée à gauche est nulle.

Les dérivées à droite et gauche en 0 étant égales, ψ est dérivable en 0 et $\psi'(0) = 0$. De plus, ce qui précède montre que ψ' est continue en 0 (sa limite à droite et gauche en 0 vaut $0 = \psi'(0)$).

Finalement ψ est de classe $\mathcal{C}^1$ sur $\mathbb{R}$ et ceci donne le résultat au rang $k+1$.

On a ainsi montré que

$$\boxed{\varphi \text{ est de classe } \mathcal{C}^\infty \text{ sur } \mathbb{R}}$$

Remarque : le programme ne donne pas de généralisation à la classe $\mathcal{C}^k$ pour le théorème de limite de la dérivée. C'est pourquoi nous avons mené une récurrence. Il nous semble qu'il n'est pas déraisonnable de conclure un peu plus rapidement en évoquant l'existence de limites en 0 pour toutes les dérivées de φ . Mais il faut cependant clairement décrire le processus à l'oeuvre dans le théorème de limite de la dérivée : dans un premier temps, il donne la dérivabilité au point puis le fait que la dérivée est continue en ce point. Il n'est pas faux de dire que "les dérivées sont prolongeables par continuité en 0" mais cela ne répond pas à la question car cela n'inclut a priori pas le fait que la fonction est dérivable. C'est d'ailleurs tout l'objet du théorème de limite de la dérivée que de dire que la prolongeabilité de la dérivée donne la dérivabilité (ce qui n'est pas une trivialité).

28. On vérifie que

$$\forall t, \psi(t) = \varphi(1 - t^2)$$

En effet, si $|t| \geq 1$, $1 - t^2 \leq 0$ et $\varphi(1 - t^2) = 0 = \psi(t)$ et si $|t| < 1$, $1 - t^2 > 0$ et $\varphi(1 - t^2) = e^{1/(t^2-1)} = \psi(t)$. Par théorèmes d'opération,

$$\boxed{\psi \in \mathcal{C}^\infty}$$

29. $\boxed{\theta \in \mathcal{C}^\infty}$ comme primitive d'une telle fonction. De plus θ' est nulle sur chaque intervalle $] -\infty, -1]$ et $[1, +\infty[$ et donc θ est constante sur chacun de ces intervalles.

$$\boxed{\theta \text{ est constante sur }] -\infty, -1] \text{ et sur } [1, +\infty[}$$

Par théorème fondamental,

$$\theta(x) = \int_0^x \psi(t) dt$$

et les constantes sont

$$A = - \int_{-1}^0 e^{\frac{1}{t^2-1}} dt \quad \text{et} \quad B = \int_0^1 e^{\frac{1}{t^2-1}} dt$$

Dans les deux cas, on intègre une fonction continue positive non nulle et les intégrales sont > 0 . Ainsi

$$\boxed{A < 0 < B}$$

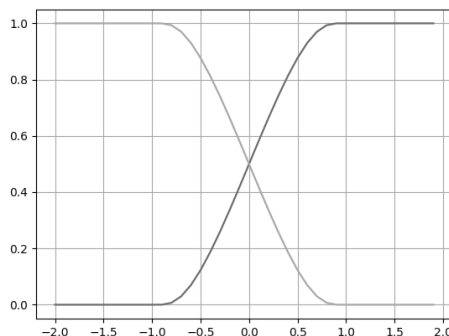
et les constantes sont en particulier différentes.

30. La question précédente nous donne une fonction avec deux plateaux. On peut la normaliser pour se ramener au cas $A = 0$ et $B = 1$ ou $A = 1$ et $B = 0$.

En notant $h_1 = \frac{\theta-A}{B-A}$, on obtient une fonction de classe $\mathcal{C}^\infty$ sur $\mathbb{R}$, nulle sur $] -\infty, -1]$ et valant 1 sur $[1, +\infty[$.

De même $h_2 = \frac{\theta-B}{A-B}$ est une fonction de classe $\mathcal{C}^\infty$ sur $\mathbb{R}$, nulle sur $[1, +\infty[$ et qui vaut 1 sur $] -\infty, -1]$.

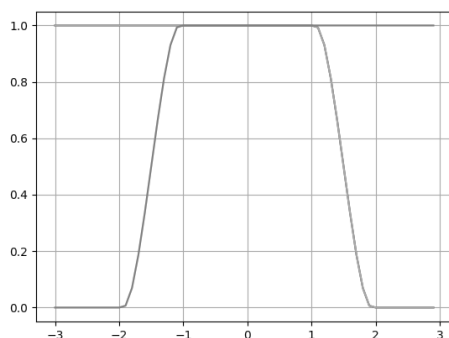
La représentation graphique de ces fonctions est la suivante.



L'idée est de décaler une courbe sur la gauche et l'autre sur la droite, c'est à dire de considérer $x \mapsto h_1(x+1)h_2(x-1)$. On obtiendra bien ainsi une fonction nulle hors de $[-2, 2]$. Mais cela ne suffit pas car on veut le caractère constant sur $[-1, 1]$: il faut que les plateaux à hauteur 1 se chevauchent.

On cherche une transformation $x \mapsto ax+b$ qui envoie $]-\infty, -2]$ sur $]-\infty, -1]$ et $[-1, +\infty[$ sur $[1, +\infty[$ afin que $h(ax+b)$ soit nul si $x \leq -2$ et égal à 1 si $x \geq -1$. On trouve que $a = 2$ et $b = 3$ conviennent.

On pose donc $\rho_1 : x \mapsto h_1(2x+3)$ et de même $\rho_2 : x \mapsto h_2(2x-3)$ dont les graphes sont



La fonction $\rho_1\rho_2$ est de classe C^∞ , nulle hors de $]-2, 2[$ et vaut 1 sur $[-1, 1]$.

Il existe $\rho \in C^\infty(\mathbb{R})$, constante égale à 1 sur $[-1, 1]$ et constante égale à 0 sur $\mathbb{R} \setminus [-2, 2]$

Remarque : il me semble indispensable de faire un dessin pour illustrer la construction. On notera que le rapport indique qu'un effort de représentation graphique était valorisé.

II.D -Inégalités de Bernstein

31. On remarque tout d'abord que

$$r(x) = \frac{1}{2\pi} \int_{-2}^{+2} e^{ix\xi} \rho(\xi) d\xi$$

On utilise le théorème de régularité des intégrales à paramètres.

- Pour tout x , $\xi \mapsto e^{ix\xi}\rho(\xi)$ est continue (par morceaux) sur le segment et donc intégrable sur ce segment.
- Pour tout $\xi \in [-2, 2]$, $x \mapsto e^{ix\xi}\rho(\xi)$ est de classe C^1 de dérivée $x \mapsto i\xi e^{ix\xi}\rho(\xi)$.
- Pour tout x , $\xi \mapsto i\xi e^{ix\xi}\rho(\xi)$ est continue (par morceaux).
- On a enfin, pour tout $x \in \mathbb{R}$ et $\xi \in [-2, 2]$, $|i\xi e^{ix\xi}\rho(\xi)| \leq 2\|\rho\|_{\infty, [-2, 2]}$. Le majorant est une fonction constante qui est intégrable sur le segment $[-2, 2]$.

$$r \in C^1(\mathbb{R}) \text{ et } r'(x) = \frac{i}{2\pi} \int_{-\infty}^{+\infty} \xi e^{ix\xi} \rho(\xi) d\xi$$

Remarque : dans ce théorème, une domination pour x dans un segment de $\mathbb{R}$ arbitraire, ou dans $[-a, a]$ pour un a quelconque, aurait été suffisante. Cette possibilité n'apporte rien dans ce cas mais il faut toujours l'avoir en tête.

32. Utilisons à nouveau l'expression ci-dessus de r . On a une intégrale de fonction de classe C^∞ et on est sur un segment. On va pouvoir opérer des intégrations par parties. On primitive $\xi \mapsto x^2 e^{ix\xi}$ en $\xi \mapsto -ix e^{ix\xi}$ et on dérive ρ :

$$2\pi x^2 r(x) = [-ix e^{ix\xi} \rho(\xi)]_{-2}^{+2} + i \int_{-2}^{+2} x e^{ix\xi} \rho'(\xi) d\xi$$

ρ étant nulle en 2 et -2 , le crochet est nul. Une nouvelle intégration par parties donne alors

$$2\pi x^2 r(x) = [e^{ix\xi} \rho'(\xi)]_{-2}^{+2} - \int_{-2}^{+2} e^{ix\xi} \rho''(\xi) d\xi$$

Le crochet est là encore nul. Par positivité de l'intégrale (majoration grossière) on trouve alors que

$$|2\pi x^2 r(x)| \leq \int_{-2}^{+2} |e^{ix\xi} \rho''(\xi)| d\xi \leq 4\|\rho''\|_{\infty, [-2, 2]}$$

Le majorant étant indépendant de x , on a montré que

$$x \mapsto x^2 r(x) \text{ est bornée sur } \mathbb{R}$$

r est continue sur $\mathbb{R}$ et les seuls problèmes d'intégrabilité sont au voisinage des infinis. En notant M un majorant de $|x^2 r(x)|$, on a $|r(x)| \leq M/x^2$ qui prouve cette intégrabilité par comparaison aux fonctions de Riemann.

$$r \text{ est intégrable sur } \mathbb{R}$$

Enfin, on a (avec l'expression déjà utilisée de r et une majoration grossière) $|r(x)| \leq \frac{2}{\pi} \|\rho\|_{\infty, [-2, 2]}$ et

$$r \text{ est bornée sur } \mathbb{R}$$

33. Commençons par le cas $\lambda = 1$. Les fonctions f et r vérifient les hypothèses de la question 26 et on a donc $\widehat{f * r} = \hat{f}\hat{r}$.

Par ailleurs, le second résultat d'inversion de Fourier donne $\rho = \hat{r}$ et $\hat{r}$ est donc égale à 1 sur $[-1, 1]$.

Ainsi, $\hat{f}\hat{r}$ est égale à $\hat{f}$ sur $[-1, 1]$ mais cela est aussi vrai ailleurs (où il y a nullité).

On a donc $\widehat{f * r} = \hat{f}\hat{r} = \hat{f}$ et donc $f * r = f$ par formule d'inversion de Fourier.

Pour un $\lambda > 0$ quelconque, on remarque que (changement de variable $u = \lambda t$)

$$f * r_\lambda(x) = \int_{-\infty}^{+\infty} f(x-t)r(\lambda t) dt = \frac{1}{\lambda} \int_{-\infty}^{+\infty} f\left(x - \frac{u}{\lambda}\right)r(u) du = \frac{1}{\lambda}(f_{1/\lambda} * r)(\lambda x)$$

Or, $\widehat{f_{1/\lambda}}(x) = \lambda \hat{f}(\lambda x)$ est nulle en dehors du segment $[-1, 1]$, intégrable sur $\mathbb{R}$ et $f_{1/\lambda} \in L^1(\mathbb{R}) \cap \mathcal{C}^1(\mathbb{R})$. On peut donc, avec le premier cas, affirmer que $f_{1/\lambda} * r = f_{1/\lambda}$. Ainsi

$$f * r_\lambda(x) = \frac{1}{\lambda} f_{1/\lambda}(\lambda x) = \frac{1}{\lambda} f(x)$$

$$\boxed{f = \lambda f * r_\lambda}$$

34. La question 25 donne alors en dérivant (r et sa dérivée sont bornées) $f' = \lambda f * (r_\lambda)'$ et la question 24 indique que

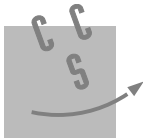
$$\|f'\|_\infty \leq \lambda \|f\|_\infty \|(r_\lambda)'\|_1$$

Il suffit alors de remarquer que

$$\int_{-\infty}^{\infty} |(r_\lambda)'(x)| dx = \int_{-\infty}^{\infty} |\lambda r'(\lambda x)| dx = \int_{-\infty}^{+\infty} |r'(u)| du$$

pour conclure que

$$\boxed{\|f'\|_\infty \leq \lambda \|f\|_\infty \|r'\|_1}$$

*Arbres couvrants et pavages*

Le seul langage de programmation autorisé dans cette épreuve est Caml.

Toutes les fonctions des modules `Array` et `List`, ainsi que les fonctions de la bibliothèque standard (celles qui s'écrivent sans nom de module, comme `max` ou `incr` ainsi que les opérateurs comme `/` ou `mod`) peuvent être librement utilisés.

On utilisera également le générateur pseudo-aléatoire `int` du module `Random`. Quand n est un entier supérieur ou égal à 1, l'appel `Random.int n` renvoie un entier dans l'intervalle $\llbracket 0, n \rrbracket$ (compris entre 0 inclus et n exclu) en simulant une loi uniforme. Les appels successifs à cette fonction fournissent des résultats indépendants.

Les candidats ne devront faire appel à aucun autre module.

En Caml, les tableaux d'objets d'un certain type `'a` sont représentés par le type `'a array`. L'expression `tab.(i)` permet d'accéder à l'élément situé en i -ème position du tableau `tab`. Dans le texte, en dehors du code Caml, cet élément sera noté $Tab[i]$. Plus généralement, les objets mathématiques seront notés G , s , $Adj\dots$ et représentés en Caml par `g`, `s`, `adj...` sans que cela soit systématiquement explicité.

Dans ce problème, un graphe G est un couple (S, A) où :

- S est un ensemble fini dont les éléments sont les *sommets* de G ;
- $A = (a_0, a_1, \dots, a_{m-1})$ est la suite des *arêtes* de G , une arête étant une partie $a = \{s, t\}$ de S de cardinal 2. Les sommets s et t sont appelés les *extrémités* de l'arête a et on dira que a *relie* s et t . Si s et t sont reliés par une arête, on dit qu'ils sont *voisins* ou *adjacents*.

Ainsi, les graphes sont non orientés et il n'y a pas d'arête reliant un sommet à lui-même. Par contre, à partir de la partie V, il est possible que plusieurs arêtes aient les mêmes extrémités.

Par convention, nous noterons n (respectivement m) le nombre de sommets (respectivement d'arêtes) du graphe et nous supposons que $S = \{0, 1, \dots, n-1\} = S_n$.

Un graphe sera représenté par le type

```
type graphe = int list array
```

Si G est un graphe représenté par `g`, alors le nombre de sommets n du graphe est donné par la longueur du tableau `g`. De plus, si $s \in S$, alors `g.(s)` est une liste contenant les indices des sommets voisins de s , dans un ordre quelconque, chaque sommet apparaissant autant de fois qu'il existe d'arêtes vers s .

Les complexités temporelles demandées sont des complexités *dans le pire des cas* et seront exprimées sous la forme $\mathcal{O}(f(n, m))$, où f est une fonction la plus simple possible.

Nous utiliserons la représentation graphique habituelle des graphes. Le graphe G , défini par l'instruction

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

peut être représenté par le graphique de la figure 1.

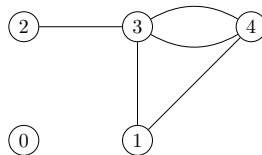
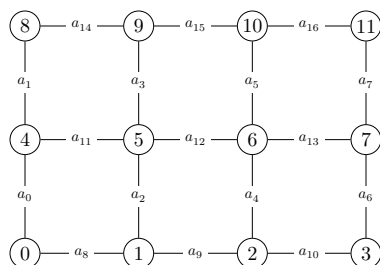


Figure 1 Exemple de graphe

Pour p, q entiers naturels non nuls, nous noterons $G_{p,q}$ le graphe, appelé *graphe quadrillage*, dont les sommets sont placés sur p colonnes et q lignes, chaque sommet étant relié à ses voisins directs. On convient de numéroté les sommets de gauche à droite et de bas en haut.

Par ailleurs, l'ordre des arêtes ayant une importance en partie V, on convient de numéroté les $p(q-1) + q(p-1)$ arêtes en commençant par les arêtes verticales, de bas en haut et de gauche à droite, puis les arêtes horizontales, de gauche à droite et de bas en haut, comme le montre la figure 2, où $p = 4$ et $q = 3$. Le *rang* d'une arête a de $G_{p,q}$ est l'indice i tel que $a = a_i$.

Figure 2 Le graphe $G_{4,3}$

Soit $G = (S_n, A)$ un graphe.

- Si $s \in S_n$, le *tableau d'adjacence* de s est le tableau, dans un ordre quelconque, des voisins de s , chaque sommet apparaissant autant de fois qu'il existe d'arêtes vers s . On note Adj le tableau de taille n tel que, pour tout $s \in \{0, 1, \dots, n-1\}$, $Adj[s]$ est le tableau d'adjacence du sommet s . Adj est donc représenté par une variable `adj` de type `int array array`.
- Un *chemin* dans G est une suite $c = (s_0, s_1, \dots, s_{j-1}, s_j, \dots, s_{k-1}, s_k)$ où pour tout j compris entre 1 et k , s_{j-1} et s_j sont des sommets voisins. On dira que c est un chemin de s_0 à s_k de longueur k . Par convention, pour s sommet de G , il existe un chemin de longueur nulle de s à s .
- La *composante connexe* d'un sommet s de G , notée C_s , est l'ensemble des sommets t de G tels qu'il existe un chemin de s à t .
- On dit que G est *connexe* si pour tous sommets s et t de G , il existe un chemin de s à t .
- Un *cycle* dans G est un chemin de longueur $k \geq 2$ d'un sommet à lui-même et dont les arêtes sont deux à deux distinctes. On dit que G est *acyclique* s'il ne contient aucun cycle.
- Un *arbre* est un graphe connexe acyclique.

I Quelques fonctions auxiliaires

Q 1. Écrire une fonction

```
nombre_aretes : graphe -> int
```

qui, appliquée à `g` représentant un graphe $G = (S, A)$, renvoie le cardinal de A .

Q 2. Écrire une commande Caml permettant de créer le tableau des tableaux d'adjacence Adj associé au graphe $G_{3,2}$.

On rappelle que la fonction `Array.of_list : 'a list -> 'a array` prend en argument une liste $[x_0; x_1; \dots; x_{k-1}]$ d'éléments de type `'a` et renvoie le tableau $[x_0; x_1; \dots; x_{k-1}]$.

Q 3. Écrire une fonction

```
adjacence : graphe -> int array array
```

qui, appliquée à `g` représentant un graphe G , renvoie `adj` représentant le tableau des tableaux d'adjacence Adj associé à G . La fonction `adjacence` devra être de complexité $\mathcal{O}(m + n)$ et on demande de justifier cette complexité.

Q 4. Écrire une fonction

```
rang : int * int -> int * int -> int
```

telle que `rang (p, q) (s, t)` renvoie le rang de l'arête $\{s, t\}$ dans le graphe quadrillage $G_{p,q}$. Par exemple, `rang (4, 3) (6, 7)` renvoie 13. On suppose que s et t sont adjacents et $s < t$ et on ne demande pas de le vérifier.

Q 5. Écrire de même sa fonction réciproque

```
sommets : int * int -> int -> int * int
```

telle que `sommets (p, q) i` renvoie le couple (s, t) tel que $a_i = \{s, t\}$ et $s < t$ dans le graphe $G_{p,q}$.

Q 6. Écrire une fonction

```
quadrillage : int -> int -> graphe
```

telle que l'instruction `quadrillage p q` renvoie le graphe représentant $G_{p,q}$.

II Caractérisation des arbres

Soit $G = (S_n, A)$ un graphe à n sommets et m arêtes, représenté par $\mathbf{g}$. Les arêtes de G sont donc numérotées $A = (a_0, a_1, \dots, a_{m-1})$.

II.A – Propriétés sur les arbres

Q 7. Montrer que les composantes connexes de G forment une partition de S_n , c'est-à-dire que :

- $\forall s \in S_n, C_s \neq \emptyset$;
- $S_n = \bigcup_{s \in S_n} C_s$;
- pour tous sommets s et t , soit $C_s = C_t$, soit $C_s \cap C_t = \emptyset$.

Q 8. Montrer que si s et t sont deux sommets de G tels que $t \in C_s$, il existe un plus court chemin de s à t et que les sommets d'un plus court chemin sont deux à deux distincts.

Pour $k \in \{0, 1, \dots, m\}$, G_k désigne le graphe $(S_n, (a_0, a_1, \dots, a_{k-1}))$ obtenu en ne conservant que les k premières arêtes de G .

Q 9. On suppose que G est un arbre. Montrer que pour tout $k \in \{0, 1, \dots, m-1\}$, les extrémités de a_k appartiennent à deux composantes connexes différentes de G_k . En déduire que $m = n - 1$.

Q 10. Montrer que les trois propriétés suivantes sont équivalentes :

- (i) G est un arbre ;
- (ii) G est connexe et $m = n - 1$;
- (iii) G est acyclique et $m = n - 1$.

II.B – Manipulation de partitions

Nous souhaitons écrire une fonction qui teste si un graphe est un arbre. Nous allons pour cela utiliser une structure de données permettant de manipuler les partitions de l'ensemble S_n : si $\mathcal{P} = \{X_1, X_2, \dots, X_k\}$ est une partition de S_n , on choisit dans chaque partie X_i , un élément particulier r_i , appelé *représentant* de X_i . Notre structure de données doit nous permettre :

- de calculer, pour $s \in S_n$, le représentant de la partie X_i contenant s ; cet élément sera également appelé *représentant* de s ;
- pour deux entiers s et t représentant des parties distinctes X_i et X_j , de transformer $\mathcal{P}$ en réunissant X_i et X_j , s ou t devenant le représentant de la partie $X_i \cup X_j$.

Nous représenterons une partition $\mathcal{P} = \{X_1, \dots, X_k\}$ de S_n par une *forêt* : chaque X_i est représenté par un arbre dont les noeuds sont étiquetés par les éléments de X_i et de racine le représentant r_i de X_i , les arcs étant orientés vers la racine. Nous noterons $h(r_i)$ la hauteur de l'arbre X_i , c'est-à-dire la longueur de sa plus longue branche. Ainsi, $\mathcal{P}_9 = \{\{0, 2\}, \{1, 5, 6, 8\}, \{3, 4, 7\}\}$ est une partition de S_9 et peut par exemple être représentée par la forêt de la figure 3.

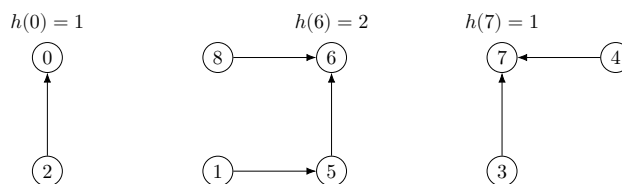


Figure 3 Une représentation de $\mathcal{P}_9 = \{\{0, 2\}, \{1, 5, 6, 8\}, \{3, 4, 7\}\}$

Le calcul du représentant d'un entier $s \in S_n$ se fait donc en remontant jusqu'à la racine de l'arbre (ce qui justifie l'orientation des arcs). Dans l'exemple, les représentants de 2, 1 et 7 sont respectivement 0, 6 et 7.

Une partition $\mathcal{P}$ de $S_n = \{0, 1, \dots, n-1\}$ sera représentée par un tableau d'entiers P de taille n de sorte que pour tout $s \in \{0, 1, \dots, n-1\}$, $P[s]$ est le père de s si s n'est pas un représentant, et est égal à $-1 - h(s)$ si s est un représentant. La partition $\mathcal{P}_9$ peut ainsi être représentée par le tableau

[| -2; 5; 0; 7; 7; 6; -3; -2; 6 |]

La réunion de deux parties X_i et X_j de représentants s et t distincts se fait selon la méthode suivante :

- si $h(s) > h(t)$, s est choisi pour représentant de la partie $X_i \cup X_j$ et devient le père de t ;
- si $h(s) \leq h(t)$, t est choisi pour représentant de la partie $X_i \cup X_j$ et devient le père de s .

Par exemple, la réunion des parties X_1 et X_3 de la partition $\mathcal{P}_9$ peut donner la nouvelle forêt représentée figure 4.

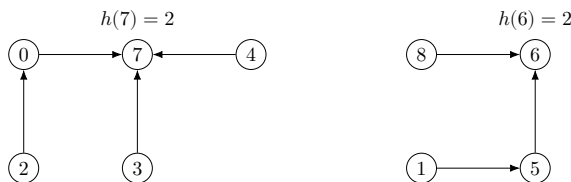


Figure 4 Une représentation de $\{\{0, 2\} \cup \{3, 4, 7\}, \{1, 5, 6, 8\}\}$

Q 11. Écrire une fonction

`representant : int array -> int -> int`

qui, appliquée à un tableau représentant une partition de S_n et à $s \in S_n$, renvoie le représentant de s .

Q 12. Écrire une fonction

`union : int array -> int -> int -> unit`

qui, appliquée à un tableau représentant une partition de S_n et à deux représentants s et t distincts, modifie la partition en réunissant les arbres associés à s et t , selon la méthode expliquée ci-dessus, sans oublier, si nécessaire, de modifier $h(s)$ ou $h(t)$.

On note $\mathcal{P}_n^{(0)}$ la partition de S_n où toutes les parties sont des singletons, c'est-à-dire $\mathcal{P}_n^{(0)} = \{\{0\}, \{1\}, \dots, \{n-1\}\}$.

Q 13. Soit $\mathcal{P}$ une partition de S_n construite à partir de $\mathcal{P}_n^{(0)}$ par des réunions successives selon la méthode précédente. Montrer que si s est le représentant d'une partie $X \in \mathcal{P}$, alors le cardinal de X vérifie $|X| \geq 2^{h(s)}$.

Q 14. En déduire les complexités des deux fonctions précédentes dans le pire des cas en fonction de n pour une partition $\mathcal{P}$ construite à partir de $\mathcal{P}_n^{(0)}$.

Q 15. Écrire une fonction

`est_un_arbre : graphe -> bool`

qui, appliquée à un graphe g représentant un graphe G , renvoie `true` si G est un arbre et `false` sinon.

III Algorithme de Wilson : arbre couvrant aléatoire

Si $G = (S_n, A)$ est un graphe connexe et si $r \in S_n$, un *arbre couvrant de G enraciné en r* est un arbre $\mathcal{T} = (S_n, B)$ tel que $B \subset A$ et dont r a été choisi pour racine. On convient alors d'orienter les arêtes de l'arbre en direction de la racine r et de représenter $\mathcal{T}$ par un tableau `parent` de taille n , `parent.(s)` étant égal au père de s dans $\mathcal{T}$ si $s \neq r$, et à -1 si $s = r$.

La figure 5 représente deux arbres couvrants du graphe $G_{3,2}$, enracinés respectivement en 0 et 2, et leurs représentations.

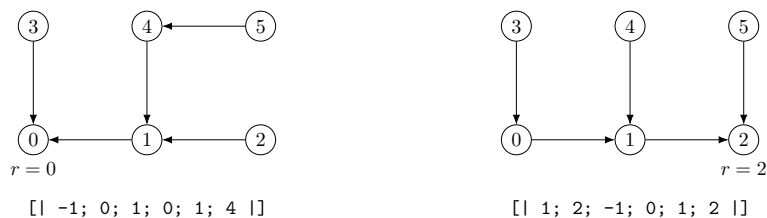


Figure 5 Deux arbres couvrants enracinés de $G_{3,2}$ et leurs représentations

Dans cette partie, nous supposons que $G = (S_n, A)$ est un graphe connexe, représenté par g , que r est un sommet de G et nous cherchons à construire un arbre couvrant aléatoire de G enraciné en r .

Nous allons pour cela faire évoluer dynamiquement un arbre $\mathcal{T}$, représenté par un tableau `parent` de longueur n vérifiant :

- `parent.(r) = -1` ;
- si $s \in S_n$ n'est pas un sommet de $\mathcal{T}$, `parent.(s) = -2` ;
- si $s \in S_n$ est un sommet de $\mathcal{T}$ autre que la racine, `parent.(s)` est le père de s dans l'arbre $\mathcal{T}$.

Nous partons de l'arbre réduit à la racine r , en posant $\text{parent}.(r) = -1$ et $\text{parent}.(s) = -2$ pour tout sommet $s \neq r$. Pour tout sommet s , quand s n'est pas déjà dans l'arbre, on construit un chemin aléatoire c sans boucle qui part de s et qui s'arrête dès qu'il a atteint un sommet de $\mathcal{T}$. La méthode de construction est la suivante :

- au début du calcul, c est réduit à s ;
- à tout moment, c est de la forme $(s = s_0, s_1, \dots, s_{k-1}, s_k)$ où les sommets $s_0, \dots, s_k$ sont deux à deux distincts et les sommets $s_0, \dots, s_{k-1}$ ne sont pas des sommets de $\mathcal{T}$;
- tant que l'extrémité s_k n'est pas un sommet de $\mathcal{T}$, on choisit aléatoirement et uniformément un voisin u de s_k et on distingue,
 - si $u \in \{s_0, s_1, \dots, s_k\}$, on supprime le cycle qui vient d'être formé et u devient le nouveau point d'arrivée du chemin c ;
 - sinon, c devient le chemin $(s = s_0, s_1, \dots, s_{k-1}, s_k, u)$;
- on renvoie le chemin $(s = s_0, s_1, \dots, s_{k-1}, s_k)$ une fois le calcul terminé.

On représentera un chemin en Caml par le type :

```
type chemin = {debut : int; mutable fin : int; suivant : int array}
```

de telle sorte que si le chemin $c = (s_0, s_1, \dots, s_{k-1}, s_k)$ est représenté par c , alors $c.\text{debut}$ est égal à s_0 , $c.\text{fin}$ (qui est un champ mutable) est égal à s_k , et $c.\text{suivant}$ est un tableau de taille n tel que pour $j \in \{0, \dots, k-1\}$, la case d'indice s_j de $c.\text{suivant}$ contient le sommet s_{j+1} . Pour tout autre sommet $t \notin \{s_0, s_1, \dots, s_{k-1}\}$, $c.\text{suivant}.(t)$ pourra prendre une valeur arbitraire.

On rappelle que si c est un objet de type chemin et u un entier représentant un sommet, la modification du champ mutable $c.\text{fin}$ pour y mettre u peut se faire avec la commande :

```
c.fin <- u
```

Q 16. Déterminer, dans le graphe $G_{3,2}$, le chemin représenté par l'objet

```
{debut = 1; fin = 4; suivant = [-5; 2; 5; 3; -1; 4]}
```

Q 17. Que peut-on dire de la terminaison de cet algorithme ?

Q 18. Écrire une fonction

```
marche_aleatoire : int array array -> int array -> int -> chemin
```

telle que l'appel `marche_aleatoire adj parent s` renvoie l'objet c représentant un chemin de s à un sommet $t \in \mathcal{T}$ obtenu comme décrit précédemment. On rappelle que `adj` représente le tableau des tableaux d'adjacence Adj du graphe G , que `parent` représente l'arbre (en construction) $\mathcal{T}$ et on supposera que $s \in S_n$ n'appartient pas à $\mathcal{T}$.

L'algorithme d'évolution de $\mathcal{T}$, appelé algorithme de Wilson, est alors le suivant : à partir de l'arbre $\mathcal{T}$ réduit à r , pour s variant de 0 à $n-1$, si s n'est pas dans $\mathcal{T}$, on calcule un chemin de s à un sommet $t \in \mathcal{T}$ codé par c grâce à la fonction `marche_aleatoire` et on greffe c à $\mathcal{T}$, en ajoutant à $\mathcal{T}$ les arêtes du chemin c , orientées de s vers u , u étant le dernier sommet du chemin.

Q 19. Écrire une fonction

```
greffe : int array -> chemin -> unit
```

telle que l'instruction `greffe parent c` modifie `parent` de sorte à représenter l'arbre obtenu après la greffe du chemin c sur $\mathcal{T}$.

Q 20. Écrire une fonction

```
wilson : graphe -> int -> int array
```

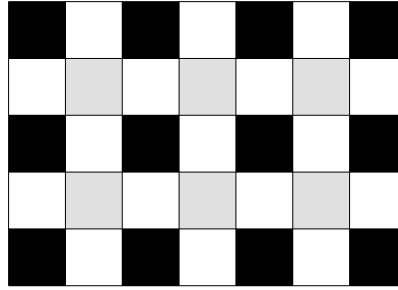
tel que `wilson g r` renvoie un arbre couvrant aléatoire du graphe G de racine r .

IV Arbres couvrants et pavages par des dominos

Soient p et q deux entiers strictement supérieurs à 1. Le but des deux dernières parties est de construire des pavages aléatoires par des dominos de taille 2×1 d'un échiquier à $2p-1$ colonnes et $2q-1$ lignes privé de son coin inférieur gauche (il reste bien un nombre pair de cases à recouvrir). Nous noterons $E_{p,q}$ cet échiquier, dont les cases sont repérées par des couples de coordonnées entières (x, y) , avec $0 \leq x \leq 2(p-1)$ et $0 \leq y \leq 2(q-1)$, et $E'_{p,q}$ l'échiquier $E_{p,q}$ privé de son coin inférieur gauche, c'est-à-dire de la case $(0, 0)$. Les cases de $E_{p,q}$ sont colorées en noir, gris et blanc, comme dans l'exemple de l'échiquier $E_{4,3}$ présenté figure 6.

Les cases dont les deux coordonnées sont paires sont colorées en noir, celles dont les deux coordonnées sont impaires sont colorées en gris, les autres sont colorées en blanc.

Si $\mathcal{P}$ est un pavage de $E'_{p,q}$, chaque case noire ou grise de $E'_{p,q}$ est recouverte par un domino, qui est orienté, à partir de la case noire ou grise considérée, vers le sud, l'ouest, le nord ou l'est.

Figure 6 L'échiquier $E_{4,3}$

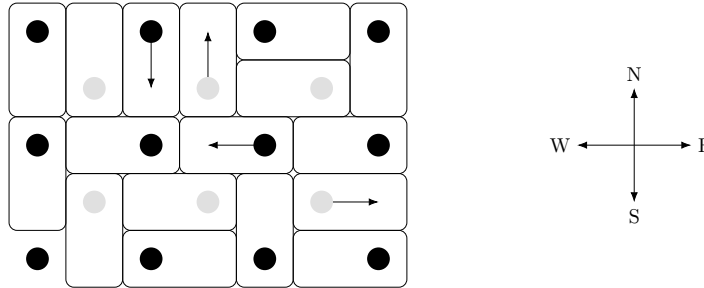
Nous définissons le type

```
type direction = S | W | N | E
```

et nous codons $\mathcal{P}$ par une matrice **pavage** de taille $(2p-1) \times (2q-1)$, avec, pour $i \in \{0, \dots, 2p-2\}$ et $j \in \{0, \dots, 2q-1\}$:

- si i et j ont la même parité et si $(i, j) \neq (0, 0)$, $\mathbf{p} \cdot (i) \cdot (j)$ est la direction du domino qui recouvre la case (i, j) (cette case est soit noire, soit grise) ;
- sinon, $\mathbf{pavage} \cdot (i) \cdot (j)$ prend la valeur N (ces valeurs ne jouent aucun rôle).

Ainsi, si $\mathbf{p1}$ est la matrice associée au pavage $\mathcal{P}_1$ de $E'_{4,3}$ représenté figure 7 (pour mieux distinguer les dominos, les cases ont été remplacées par des ronds colorés) on a par exemple $\mathbf{p1} \cdot (2) \cdot (4) = \mathbf{S}$, $\mathbf{p1} \cdot (4) \cdot (2) = \mathbf{W}$, $\mathbf{p1} \cdot (3) \cdot (3) = \mathbf{N}$, $\mathbf{p1} \cdot (5) \cdot (1) = \mathbf{E}$, comme indiqué par les sens des flèches.

Figure 7 Le pavage $\mathcal{P}_1$ de $E'_{4,3}$

Les cases noires de l'échiquier $E_{p,q}$ seront vues comme les sommets du graphe $G_{p,q}$ (la case noire située en bas à gauche est identifiée au sommet 0 du graphe $G_{p,q}$). Un résultat de Neville Temperley explicite une bijection canonique φ entre les pavages de $E'_{p,q}$ et les arbres couvrants de $G_{p,q}$. La construction de $\varphi(\mathcal{P})$ à partir d'un pavage $\mathcal{P}$ s'obtient en ne conservant que les dominos recouvrant une case noire.

Pour toute la suite du sujet, les valeurs de p et q seront supposées contenues dans les variables globales $\mathbf{p}$ et $\mathbf{q}$. On suppose de plus définie une variable globale $\mathbf{g}$, représentant $G_{p,q}$, par l'instruction

```
let g = quadrillage p q ;;
```

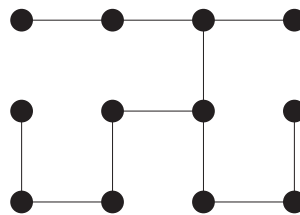
IV.A – Exemples

Q 21. Dessiner l'arbre couvrant *non enraciné* $\mathcal{T}_1 = \varphi(\mathcal{P}_1)$ de $G_{4,3}$.

Q 22. Considérons inversement l'arbre couvrant $\mathcal{T}_2$ de $G_{4,3}$ décrit figure 8. Par un procédé réciproque à celui utilisé à la question précédente, dessiner le pavage $\mathcal{P}_2 = \varphi^{-1}(\mathcal{T}_2)$ de $E'_{4,3}$.

IV.B – Calcul de l'arbre couvrant associé à un pavage

Q 23. Soit $\mathcal{P}$ un pavage de $E'_{p,q}$, associé à l'arbre couvrant $\mathcal{T} = \varphi(\mathcal{P})$ de $G_{p,q}$. Décrire un procédé permettant de calculer le père d'un sommet $s \in \{1, \dots, pq-1\}$ dans l'arbre $\mathcal{T}$ enraciné en 0. On ne demande pas de démontrer que la structure ainsi obtenue est bien un arbre de racine 0.

Figure 8 L'arbre couvrant $\mathcal{T}_2$

Q 24. Écrire une fonction

```
coord_noire : int -> int * int
```

telle que l'instruction `coord_noire s` renvoie le couple des coordonnées de la case correspondant au sommet s du graphe $G_{p,q}$. Par exemple, dans le graphe $G_{4,3}$ représenté figure 2, `coord_noire 6` renverra $(4, 2)$.

Q 25. Écrire une fonction

```
sommet_direction : int -> direction -> int
```

telle que `sommet_direction s d` renvoie le sommet t qui se trouve directement dans la direction d du sommet s de $G_{p,q}$. Cette fonction renverra -1 si un tel sommet n'existe pas. Par exemple, dans le graphe $G_{4,3}$ représenté figure 2, `sommet_direction 5 W` renverra 4.

Q 26. Écrire une fonction

```
phi : direction array -> int array
```

qui, appliquée à une matrice `pavage` représentant un pavage $\mathcal{P}$ de $E'_{p,q}$, renvoie le tableau `parent` représentant l'arbre $\mathcal{T} = \varphi(\mathcal{P})$ enraciné en 0 (cette représentation est définie au début de la partie III).

V Utilisation du dual pour la construction d'un pavage

V.A – Graphe dual de $G_{p,q}$

Le graphe $G_{p,q}$ est un *graphe planaire*, c'est-à-dire qu'il possède une représentation graphique dans laquelle deux arêtes distinctes ne se coupent pas, sauf peut-être en leurs extrémités. Cette *représentation planaire* sépare le plan en $(p-1)(q-1)+1$ faces, que l'on va numéroté de gauche à droite et de bas en haut, la face non bornée portant le numéro 0. Le *graphe dual* de $G_{p,q}$, noté $G_{p,q}^*$, est alors le graphe à $(p-1)(q-1)+1$ sommets (chaque sommet correspond à une des faces délimitées par la représentation de $G_{p,q}$) et qui possède autant d'arêtes que $G_{p,q}$: chaque arête a de $G_{p,q}$ donne une arête a^* entre les deux faces du plan qu'elle sépare. Ce graphe est également planaire. Attention, contrairement au graphe $G_{p,q}$, le graphe $G_{p,q}^*$ *peut posséder plusieurs arêtes entre les mêmes sommets*. Les sommets de $G_{p,q}$ ont déjà été identifiés aux cases noires de $E_{p,q}$; les cases grises seront identifiées aux sommets de $G_{p,q}^*$ distincts de 0 et les cases blanches aux arêtes de $G_{p,q}$ (et donc également aux arêtes de $G_{p,q}^*$, puisque $a \mapsto a^*$ est une bijection). La figure 9 explicite ces identifications dans le cas du graphe $G_{4,3}$. Les sommets de $G_{4,3}$ sont identifiés par des sommets gris foncés, ceux de $G_{4,3}^*$ par des sommets gris clairs, les arêtes de $G_{4,3}$ par des traits pleins et celles de $G_{4,3}^*$ par des traits hachurés. L'intersection de chaque (a, a^*) est marquée par un carré symbolisant une case blanche de $E_{4,3}$.

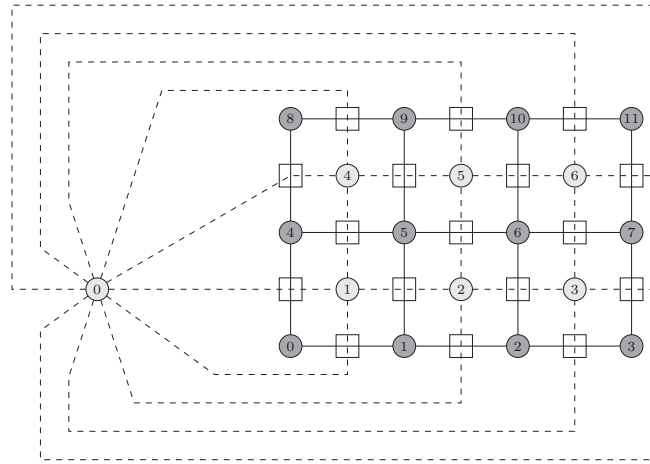
Ainsi, le sommet 6 de $G_{4,3}$ correspond à la case $(4, 2)$ de $E_{4,3}$, le sommet 4 de $G_{4,3}^*$ correspond à la case $(1, 3)$ et les arêtes a_9 de $G_{4,3}$ et a_9^* de $G_{4,3}^*$ correspondent à la case $(3, 0)$.

Dans la suite du problème, nous notons :

- $n = pq$ le nombre de sommets de $G_{p,q}$;
- $n^* = (p-1)(q-1)+1$ le nombre de sommets de $G_{p,q}^*$;
- $m = p(q-1) + q(p-1)$ le nombre d'arêtes de $G_{p,q}$ et de $G_{p,q}^*$;
- A l'ensemble des arêtes de $G_{p,q}$;
- A^* l'ensemble des arêtes de $G_{p,q}^*$.

Nous supposons définies pour la suite :

- une fonction `coord_grise : int -> int * int` qui, appliquée à un sommet de $G_{p,q}^*$ autre que 0, renvoie les coordonnées de la case grise qui correspond à ce sommet ;
- une fonction `numero : int * int -> int` qui, appliquée à un couple (x, y) représentant une case noire ou grise de l'échiquier $E_{p,q}$, renvoie le sommet du graphe $G_{p,q}$ ou $G_{p,q}^*$ associé à cette case. On supposera également que dans tous les autres cas, `numero` renvoie la valeur 0, y compris si les coordonnées sont en dehors de l'échiquier. Quand $p = 4$ et $q = 3$, nous avons par exemple : `numero (4, 2) = 6`, `numero (1, 3) = 4`, et `numero (3, 0) = 0`.

Figure 9 $G_{4,3}$ et $G_{4,3}^*$

Q 27. En utilisant les fonctions précédentes, écrire une fonction

```
dual : unit -> graphe
```

telle que l'instruction `dual ()` renvoie le graphe représentant $G_{p,q}^*$. On respectera la numérotation des sommets de $G_{p,q}^*$ décrite ci-dessus (figure 9). Par ailleurs, les listes d'adjacence du graphe dual contiendront des sommets en doublon s'il existe plusieurs arêtes entre des mêmes sommets.

On suppose désormais définie une variable globale `g_etoile` par l'instruction

```
let g_etoile = dual () ;;
```

V.B – Dual d'un arbre couvrant

Q 28. Montrer que, quitte à renuméroter les sommets, le dual de $G_{p,q}^*$ est le graphe $G_{p,q}$.

Soit B une partie de A . On note B^* la partie de A^* définie par

$$\forall i \in \{0, 1, \dots, m-1\}, a_i^* \in B^* \Leftrightarrow a_i \notin B.$$

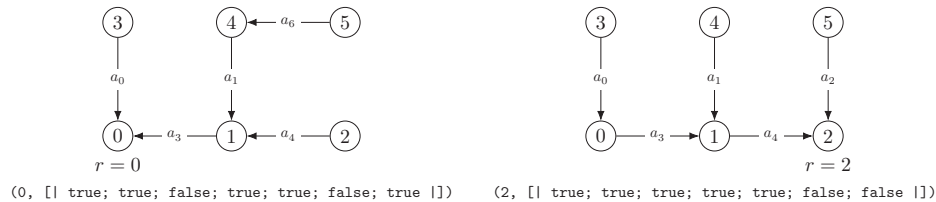
Q 29. Montrer que si le graphe (S_n, B) possède un cycle, le graphe (S_n, B^*) n'est pas connexe.

Q 30. En déduire que (S_n, B) est un arbre couvrant de $G_{p,q}$ si et seulement si (S_n, B^*) est un arbre couvrant de $G_{p,q}^*$.

Si $\mathcal{T} = (S_n, B)$ est un arbre couvrant de $G_{p,q}$, nous noterons $\mathcal{T}^*$ l'arbre couvrant (S_n, B^*) de $G_{p,q}^*$.

Pour construire l'arbre $\mathcal{T}^*$, nous utiliserons une représentation différente des arbres que celle introduite en partie III : on pourra représenter un arbre couvrant $\mathcal{T} = (S, B)$ enraciné en r par un couple $(\mathbf{r}, \mathbf{b})$ où $\mathbf{b}$ est un tableau de booléens de longueur m représentant B , c'est-à-dire tel que $\mathbf{b}(i)$ prend la valeur `true` si et seulement si $a_i \in B$.

La figure 10 représente les deux arbres couvrants du graphe $G_{3,2}$, présenté en figure 5, enracinés respectivement en 0 et 2, et leurs représentations par un couple.

Figure 10 Deux arbres couvrants enracinés de $G_{3,2}$ et leurs représentations

Q 31. Écrire une fonction

```
vers_couple : int array -> int * bool array
```

telle que l'instruction `vers_couple parent`, où `parent` est un tableau représentant un arbre enraciné de $G_{p,q}$, renvoie le couple $(\mathbf{r}, \mathbf{b})$ décrit ci-dessus.

Q 32. Détailler en français un algorithme permettant de construire **parent** à partir du couple (r, B) et de la représentation du graphe $G_{p,q}$ par listes d'adjacences. Cet algorithme est-il applicable à un arbre couvrant du graphe $G_{p,q}^*$? Justifier.

Q 33. Écrire une fonction

```
vers_parent : int * bool array -> int array
```

telle que l'instruction **vers_parent** (**r**, **b**), où **b** désigne un ensemble B tel que (S_n, B) est un arbre de $G_{p,q}$ enraciné en r , renvoie le tableau **parent** représentant cet arbre.

Q 34. Déterminer les complexités de ces deux fonctions de conversion en fonction de n et m .

On supposera écrite une fonction **vers_parent_etoile** : `int * bool array -> int array`, ayant un fonctionnement similaire à **vers_parent**, qui prend en argument un couple (r, b_etoile) correspondant à un arbre couvrant $\mathcal{T}^* = (S_n, B^*)$ de $G_{p,q}^*$ et renvoie un tableau **parent_etoile** décrivant l'arbre en utilisant la représentation décrite partie III.

Q 35. Écrire une fonction

```
arbre_dual : int array -> int array
```

qui, appliquée au tableau **parent** représentant un arbre couvrant $\mathcal{T}$ de $G_{p,q}$ enraciné en 0, renvoie le tableau **parent_etoile** représentant l'arbre couvrant $\mathcal{T}^*$ de $G_{p,q}^*$ enraciné en 0.

V.C – Calcul du pavage associé à un arbre couvrant

Soit $\mathcal{T} = (S_n, B)$ un arbre couvrant de $G_{p,q}$.

Q 36. Décrire un procédé de construction, à partir des arbres $\mathcal{T}$ et $\mathcal{T}^*$ enracinés en 0, du pavage $\mathcal{P} = \varphi^{-1}(\mathcal{T})$ de $E'_{p,q}$. On expliquera en détail comment traiter les arêtes d'un sommet de $G_{p,q}^*$ vers le sommet 0. On justifiera brièvement que l'on obtient bien un pavage.

Q 37. Écrire une fonction

```
pavage_aleatoire : unit -> direction array array
```

telle que l'instruction **pavage_aleatoire** () renvoie une matrice de taille $(2p-1)(2q-1)$ représentant un pavage aléatoire de $E'_{p,q}$ par des dominos.

Q 38. Comment adapter cette méthode à la construction de pavages aléatoires d'un échiquier à $2p$ colonnes et $2q-1$ lignes (respectivement à $2p$ colonnes et $2q$ lignes) ?

• • • FIN • • •

SOLUTION

Q1.

```
let nombre_aretes g =
  let n = Array.length g in
  let nb = ref 0 in
  for i=0 to n-1 do
    nb:=!nb + List.length g.(i) done;
  !nb/2;;
```

Commentaires : *Il s'agit d'une question élémentaire de cours.*

Q2.

```
let g32 = [| [1;3]; [0;4;2]; [1;5]; [0;4]; [3;1;5]; [4;2] |];;
```

Commentaires : *Même remarque que précédemment.*

Q3.

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

Commentaires : *Même remarque que précédemment. Si vous avez des difficultés sur l'une de ces trois questions, reconsolider votre cours sur la programmation des graphes.*

Q4.

```
let rang (p,q) (s,t) = match t-s with
| 1 -> p*(q-1)+(s/p)*(p-1)+s mod p
| _ -> (q-1)*(s mod p) + s/p;;
```

Commentaires : *Question sélective car elle exige du candidat d'imaginer la formule mathématique donnant le rang. Des candidats plus pragmatiques pourraient écrire une fonction suivant p q s t m renvoyant le numéro suivant m' ainsi que sa nouvelle position s' , t' du point de coordonnée s , t et de numéro m (pour mimer un mouvement élémentaire dans le graphe en suivant le parcours proposé). Il suffit d'itérer cette fonction depuis $s = 0$, $t = 0$ et $m = 0$ jusqu'à obtenir les coordonnées s et t . Dans ce cas, on renvoie le numéro m associé. Le code est plus long et a une plus grande complexité mais il mérite de produire le numéro sans connaître la formule mathématique !*

Q5.

```
let sommets (p,q) i = match i < p*(q-1) with
| true -> let s = i/(q-1) +(i mod (q-1))*p in s,s+p
| _ -> let i' = i-p*(q-1) in let s = p*(i'/(p-1))+(i' mod (p-1)) in s,s+1;;
```

Commentaires : Même remarque que précédemment.

Q6.

```
let quadrillage p q =
  let a = Array.make (p*q) [] in
  for i=0 to (p*q) do
    let s,t= sommets (p,q) i in
    a.(s)<-t::a.(s) ; a.(t)<-s::a.(t)
  done;
  a;;
```

Commentaires : Il s'agit d'une question élémentaire de cours. Si vous avez des difficultés sur cette question, reconsolider votre cours sur la programmation des graphes.

Q7. Par définition, tout sommet de S_n appartient à sa composante connexe C_s donc C_s est non vide. En outre, pour tout sommet s , on a

$$s \in C_s \subset \bigcup_{t \in S_n} C_t \Rightarrow S_n \subset \bigcup_{t \in S_n} C_t.$$

L'inclusion réciproque est immédiate car chaque composante connexe C_t appartient à S_n d'où l'égalité $\bigcup_{t \in S_n} C_t = S_n$.

Considérons deux composantes connexes C_s et C_t . Si $C_s \cap C_t$ est non vide, notons z l'un de ses éléments. Soit $u \in C_s$, il existe un chemin c de u à z . Il existe un chemin c' de z à t (car $z \in C_t$) donc en concaténant ces deux chemins cc' , on obtient un chemin de u à t donc $u \in C_t$. On vient de prouver l'inclusion $C_s \subset C_t$. L'inclusion réciproque en échangeant les rôles de s et t d'où l'égalité $C_s = C_t$. Ainsi, soit C_s et C_t sont disjoints, soit ils sont égaux.

Commentaires : Il s'agit d'une question de cours. Malheureusement, elle s'avère sélective. Les correcteurs attendent un argumentaire précis fondé sur les définitions précises et non de vagues affirmations non fondées. Le point clé étant que deux sommets qui possèdent un chemin de l'un à l'autre sont dans la même composante connexe.

Q8. Comme $t \in C_s$, il existe un chemin de s à t . L'ensemble L des longueurs des chemins de s à t est donc non vide et inclus dans $\mathbb{N}$ donc il possède un minimum, ce qui assure l'existence d'un chemin de s à t de longueur minimale. Soit

$$c : s_0 = s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_m = t$$

un chemin de longueur minimal. Supposons qu'il existe deux sommets s_i et s_j identiques avec $i < j$ alors le chemin

$$c' : s_0 = s \rightarrow s_1 \dots \rightarrow s_i = s_j \rightarrow s_{j+1} \rightarrow \dots \rightarrow s_m = t$$

est un chemin de s à t de longueur strictement inférieur à celui de c (puisque l'on a enlevé au moins une arête, celle de s_i à s_{i+1} par exemple). Ceci est absurde donc tous les sommets sont deux à deux distincts.

Commentaires : *Il s'agit quasiment d'une question de cours (exploiter notamment pour justifier qu'un graphe connexe sans cycle est un arbre et l'unicité du chemin de la racine à tout noeud de cette arbre). N'hésitez pas à vous entraîner sur les preuves élémentaires du cours, ces raisonnements sont fréquemment exploités dans les sujets (et cela pose toujours beaucoup de problèmes aux candidats).*

Q9. On procède par l'absurde en supposant qu'il existe un entier $k \in \{0, \dots, m-1\}$ de sorte que les deux extrémités de a_k sont dans la même composante connexe C_s . Si on pose $a_k = \{u, v\}$ alors il existe un chemin c de s à u (car $u \in C_s$). On lui ajoute l'arête $a_k : u \rightarrow v$ puis un chemin c' de v à s (qui existe car $v \in C_s$) donc on obtient un cycle $ca_k c'$ de s vers s dans G_k donc dans G . Ceci est absurde puisque G est un arbre donc il est acyclique. Par conséquent, les extrémités d'une même arête sont dans deux composantes connexes distinctes.

Par définition, $G_0 = (S_n, a_0)$ possède $n-1$ composantes connexes (tous les sommets hormis ceux de a_0 forment les seuls éléments de leur propre composante connexe, les deux extrémités de a_0 forment une seule composante connexe). En ajoutant l'arête a_1 , les composantes connexes des deux extrémités de a_1 sont fusionnées donc $n-2$ composantes connexes dans G_1 (les autres composantes connexes ne peuvent pas être reliées car il n'y a aucune arête les reliant). En itérant, on observe que G_k possède $n-k$ composantes connexes (les deux composantes connexes des extrémités de a_k sont fusionnées, les autres ne sont toujours pas connectées par aucune arête). En particulier, $G_m = G$ possède $n-m$ composantes connexes. Comme G est connexe, il n'a qu'une seule composante connexe donc $n-m = 1$ d'où $m = n-1$.

Commentaires : *Question relativement difficile et destinée aux bonnes copies (à minima, probablement aux meilleures copies).*

Q10. (i) $\Rightarrow$ (ii) et (i) $\Rightarrow$ (iii) : Résulte de la définition d'un arbre et que la question précédente. (ii) $\Rightarrow$ (i) : G étant connexe et $m = n-1$, pour montrer que G est un arbre, il suffit de montrer que G est acyclique. Supposons que G possède un cycle

$$s_0 = s \rightarrow s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_{i-1} \rightarrow s_i = s$$

alors, en retirant l'arête $a'_0 = (s, s_1)$, le graphe reste connexe (tous les sommets autres que les s_i sont reliés à s , et les sommets $s_1, \dots, s_{i-1}$ sont encore reliés à s grâce au chemin $s_1 \rightarrow s_2 \rightarrow \dots \rightarrow s_{i-1} \rightarrow s_i = s$) avec une arête de moins. Si G devient un arbre alors il possède n sommets et $m-1$ arêtes. D'après la question précédente, on obtient $m-1 = n-1$, ce qui est absurde car $m = n-1$. Par conséquent, le graphe G' obtenu à partir de G en lui enlevant l'arête a'_0 reste connexe et n'est pas acyclique. Il possède un cycle, ce qui permet d'enlever encore une arête a'_1 , etc. Ce processus doit s'arrêter car le nombre d'arêtes est fini. On aboutit à un graphe qui reste connexe, sans cycle à la fin, donc un arbre, avec strictement moins de m arêtes et n sommets, ce qui contredit l'égalité $m = n-1$ de la question précédente. On aboutit à une absurdité donc G est acyclique et comme il est connexe, G est un arbre d'où (i).

(iii) $\Rightarrow$ (i) : G étant acyclique et $m = n-1$, pour montrer que G est un arbre, il suffit de montrer que G est connexe. Supposons que G ne soit pas connexe donc il existe deux sommets s et t qui ne sont pas dans la même composante connexe. En ajoutant l'arête $\{s, t\}$ à ce graphe, il possède $m+1$ arêtes et n sommets. S'il est connexe alors c'est un arbre donc $m+1 = n-1$ d'après la question précédente, ce qui est absurde. Ainsi, ce nouveau graphe G'

n'est toujours pas connexe (mais il possède une composante connexe de moins que G) donc il existe deux sommets s' et t' qui ne sont pas dans la même composante connexe. On lui ajoute l'arête $\{s', t'\}$ ce qui nous donne un nouveau graphe G'' ayant une composante connexe de moins et une arête de plus. Ce processus s'arrête un jour (le nombre de composantes connexes décroît strictement et le nombre d'arêtes possibles est fini). On aboutit alors à un graphe connexe, sans cycle donc un arbre mais le nombre d'arêtes ne peut pas être égal à $n - 1$ (le nombre de sommets reste inchangé). On aboutit à une contradiction donc G est connexe d'où G est un arbre et (i) est démontrée.

Commentaires : Il s'agit d'un résultat classique de la théorie des graphes (mais il n'est pas au programme officiel). Cette question sera réussie uniquement par les meilleures copies.

Q11.

```
let rec representant p s =
  if p.(s) < 0 then s else representant p (p.(s));;
```

Commentaires : Question de niveau standard avec aucune difficulté de programmation mais cela nécessite du candidat une compréhension de la navigation dans l'arbre (notamment que les arêtes sont orientées des fils à la racine).

Q12.

```
(*si p.(s)<p.(t) alors h(t)<h(s) donc s devient le père de t
si p(s)=p(t) alors s devient le père de t mais la hauteur est
augmenté de 1 donc p(s) est diminué de 1 par construction p(s)
=-1-h(s)*)

let union p s t =
  if p.(s) < p.(t) then p.(t) <- s
  else
    if p.(t) < p.(s) then p.(s) <- t
    else (p.(t) <- s; p.(s) <- p.(s)-1);;
```

Commentaires : Question de difficulté standard. La seule difficulté est de bien mettre à jour la hauteur. Les candidats ne le faisant pas seront lourdement sanctionnés (c'est le cœur de la question) en n'ayant probablement aucun point à cette question. Nous verrons dans les questions suivantes que cela permet de minimiser la complexité des opérations futures.

Q13. On procède par récurrence forte sur le nombre k d'union nécessaires pour obtenir X depuis un élément de $\mathcal{P}_n^{(0)}$. Si $k = 0$ alors X est un élément de $\mathcal{P}_n^{(0)}$ c'est-à-dire X est de la forme $\{s\}$ pour un certain entier s donc s est un représentant de X . L'arbre associé est de hauteur $h(s) = 0$ et

$$|X| = 1 = 2^0 = 2^{h(s)} \geq 2^{h(s)}$$

donc la propriété est vraie au rang $k = 0$. Considérons un entier $k \geq 1$ et supposons la propriété vérifiée pour tout entier $q < k$ (i.e. pour toute partie obtenue en strictement moins

de k unions depuis $\mathcal{P}_n^{(0)}$). Considérons une partie X obtenue en k unions successives. Comme $k \geq 1$, X est l'union de deux ensembles X' et X'' , i.e. $X = X' \cup X''$, chacun d'eux étant obtenu en strictement moins de k unions. Soient s' un représentant de X' et s'' un représentant de X'' . Par hypothèse de récurrence, on a

$$|X'| \geq 2^{h(s')} \text{ et } |X''| \geq 2^{h(s')}.$$

Par définition de l'union (dans le processus du sujet), le représentant choisit entre s' et s'' pour $X' \cup X'' = X$ sera celui dont la hauteur est la plus petite. Soit r le représentant choisit, on a $h(r) \leq h(s')$ et $h(r) \leq h(s'')$ donc

$$2^{h(r)} \leq |X'| \leq |X' \cup X''| = |X|.$$

L'hérédité vient d'être démontrée ce qui achève la récurrence.

Commentaires : Question de difficulté standard qui est quasiment une question de cours (sur les arbres binaires, le lien entre le nombre de noeuds et la hauteur de l'arbre). Si vous avez eu des difficultés sur cette question (ou n'avez pas eu d'idée), consolider votre apprentissage des preuves du cours. Elles servent souvent aux concours.

Q14. Lors de l'appel *representant* p s , la fonction *representant* est appelé autant de fois que la hauteur de l'arbre où se trouve s . Or, d'après la question précédente, pour tout représentant s d'une partie X , on a

$$2^{h(s)} \leq |X| \leq |S_n| = n \Rightarrow h(s) \leq \log_2(n)$$

donc la hauteur de tout arbre associé à une partie est en $O(\log_2(n))$. Lors de chaque appel de *representant*, les opérations effectuées sont en nombre borné et chacune a une complexité constante donc une complexité en $O(1)$. Par conséquent, l'appel *representant* p s a une complexité en $O(\log_2(n)) O(1) = O(\log_2(n))$.

La complexité de *union* p s t est manifestement en $O(1)$ (nombre borné d'opérations, chacune de coût constant).

Commentaires : Question de niveau standard mais elle s'avère néanmoins sélective (il faut avoir programmer correctement *representant*, analyser la hauteur de l'arbre, etc).

Q15. Le point clé est d'appliquer la question 10 en choisissant la bonne caractérisation. La présence d'un cycle est au programme (dans la construction d'un arbre couvrant par l'algorithme de Kruskal) mais il est plus simple ici d'exploiter la connexité en utilisant l'indication du sujet avec l'union des arbres associés. Pour cela, il faut commencer par écrire une fonction fusion deux arbres associés à s et t . Ensuite, il faut écrire une fonction déterminant les composantes connexes par fusion successives. Voici le code.

```
(*fusion deux parties où se trouvent s et t parmi la partition
p*)
let fusion p s t =
  let s' = representant p s and t' = representant p t in
  if s' != t' then union p s' t';;
```

```

(*on veut calculer les composantes connexes du graphe g a n
   sommets.
on créé la partition des composantes connexes du graphe à n
sommets,
chaque sommet n'est initialement connecté qu'à lui-même donc
la hauteur
de l'arbre associé est -1.
Ensuite, on prend un sommet s du graphe g et on ajoute à sa
composante connexe
où il se trouve un à un ses voisins immédiats (liste appelée
voisins)
c'est-à-dire que l'on fusionne les parties auxquelles ils
appartiennent
ainsi que toutes les composantes connexes auxquels ils
appartiennent).
La partition formée par les composantes connexes est notée
comp *)

let composantes_connexes g n =
  let rec ajout comp s voisins = match voisins with
    | [] -> ()
    | t::q -> fusion comp s t; ajout comp s q
  in let comp = Array.make n (-1) in
  for s = 0 to n-1 do
    ajout comp s (g.(s)) done;
  comp;;

(*on teste si la partition comp ne contient qu'un représentant
,
si c'est le cas, la partition n'a qu'un représentant i.e. le
graphe
sera connexe, sinon il est non connexe. On compare le
représentant
de chaque sommet à celui d'un particulier, j'ai choisi 0*)

let connexe g n =
  let rec aux comp s = match s with
    | s when s == n -> true
    | s -> (representant comp s) == 0 && (aux comp (s+1))
  in aux (composantes_connexes g n) 0;;

let est_un_arbre g =
  let n = Array.length g in
  (nombre_aretes g == n-1) && (connexe g n);;

```

Commentaires : Question difficile qui s'adressent aux candidats ayant une aisance en programmation en Ocaml et qui ont choisi la bonne stratégie parmi celles de la question 10. Retenir que sur un code complexe, définir un cahier des charges des actions à mener et construire des fonctions effectuant ces tâches complexes. Commenter votre code ou expliquer en français ce qu'il signifie (et non ce qu'il fait ligne à ligne !). Si vous ne parvenez pas à tout faire, mais si vous êtes clair dans votre explication (sans en faire un roman) et si vous traitez une ou plusieurs tâches complexes, vous serez valorisés aux concours. N'hésitez pas à recoder vous-même la détermination des composantes connexes, c'est un exercice accessible de programmation pour une tâche complexe et importante en théorie des graphes.

Q16. Voici le chemin : $1 \rightarrow 2 \rightarrow 5 \rightarrow 4$.

Commentaires : Question sans difficulté visant à acclimater le candidat à la notion proposée.

Q17. Cet algorithme ne termine pas forcément (il peut boucler indéfiniment en choisissant toujours des sommets qui ne sont jamais dans $\mathcal{T}$). Néanmoins, soit m le nombre d'éléments de $\mathcal{T}$, la probabilité de ne pas choisir un sommet dans $\mathcal{T}$ vaut $p = 1 - \frac{m}{n} < 1$ donc ne pas choisir k sommets consécutifs qui ne sont pas dans $\mathcal{T}$ vaut p^k . En faisant tendre k vers $+\infty$, cette probabilité tend vers 0. L'algorithme ne termine pas nécessairement mais la probabilité qu'il ne s'arrête pas tend vers 0 au fur et à mesure du temps. On dit en mathématique que c'est un événement presque sûr (au sens des probabilités).

Commentaires : Question sélective (mais relativement simple) car elle demande au candidat d'analyser finement l'algorithme pour détecter des failles. L'étude mathématique n'est pas attendue. Par contre, il est attendu que le candidat observe l'existence potentielle d'une boucle infinie.

Q18. Le piège à cette question est de vouloir tester si u est l'une des valeurs précédemment calculées $s_0, \dots, s_k$ pour supprimer le cycle éventuel. En fait, il suffit de déclarer arbitrairement que u est la fin du chemin ! S'il est différent des $s_0, \dots, s_k$, c'est vrai et s'il vaut par exemple s_3 , en le déclarant fin du cycle, on déclare s_3 comme fin du cycle donc on a bien éliminé le cycle ! Un nombre important de candidats se sont perdus en tentant d'écrire un code de détection des cycles (problème difficile sans aide pour la grande majorité des candidats) alors que cela ne servait à rien. Il aurait été judicieux que le sujet le mentionne car il a fait perdre du temps et surtout de l'énergie aux candidats sur une question qui ne se pose pas en fait !

```
(*on choisit aléatoirement un sommet voisin au sommet i
parmi ses voisins*)

let rec choix adj parent suit sk =
  let nk = Array.length adj.(sk) in
  let u = adj.(sk).(Random.int nk) in
  suit.(sk) <- u;
  if parent.(u) == (-2) then choix adj parent suit u else u;;

let marche_aleatoire adj parent s =
  let n = Array.length adj in
```

```
let suit = Array.make n (-1) in
let u = choix adj parent suit s in
{debut = s ; fin = u ; suivant = suit} ;;
```

Commentaires : *Hormis la détection de cycle mentionné au début de cette réponse, le code est aisé. Si vous êtes bloqué par la détection, écrivez tout le reste du code, laissez un trou au bon endroit et expliquez au correcteur que vous ne parvenez pas à coder cette tâche. Selon la question, le correcteur pourra vous attribuer une part significative des points (parfois rien si le cœur de la question est ce que vous n'avez pas codé :-), pour le cas de notre question, il est probable qu'il vous mette ... tous les points !)*

Q19.

```
let greffe parent c =
  let sk = ref c.debut in
  while c.suivant.(!sk) != c.fin do
    parent.(c.suivant.(!sk)) <- !sk;
    sk := c.suivant.(!sk)
  done;;
```

Commentaires : *Question de difficulté standard. Pensez que si voulez modifier des variables en Ocaml, il faut passer par les références `s = ref x` et l'accès "aux valeurs" de `s` se font par déréférencement `!s`.*

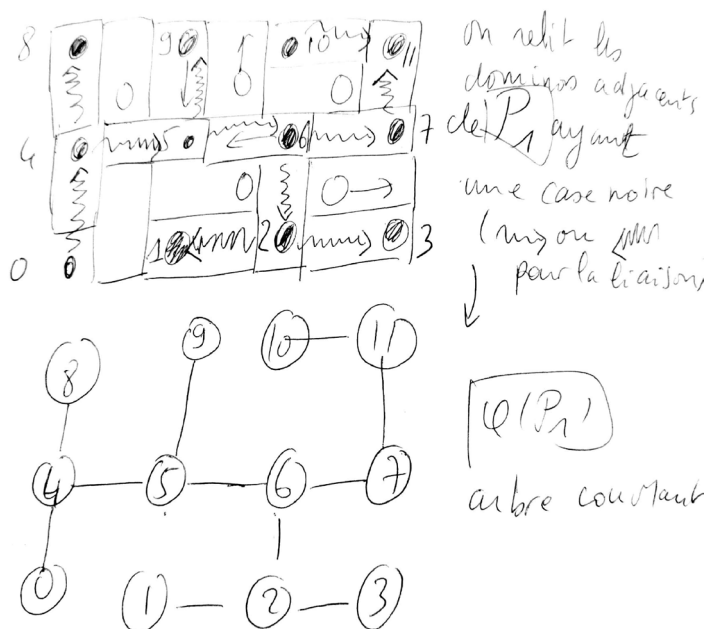
Q20.

```
let init_parent s n =
  let p = Array.make n (-2) in
  p.(s) <- (-1) ; p;;

let wilson g r =
  let adj = adjacence g in
  let n = Array.length g in
  let parent = init_parent r n in
  for s = 0 to n-1 do
    if parent.(s) == (-2) then
      let c = marche_aleatoire adj parent s in
      greffe parent c done;
  parent;;
```

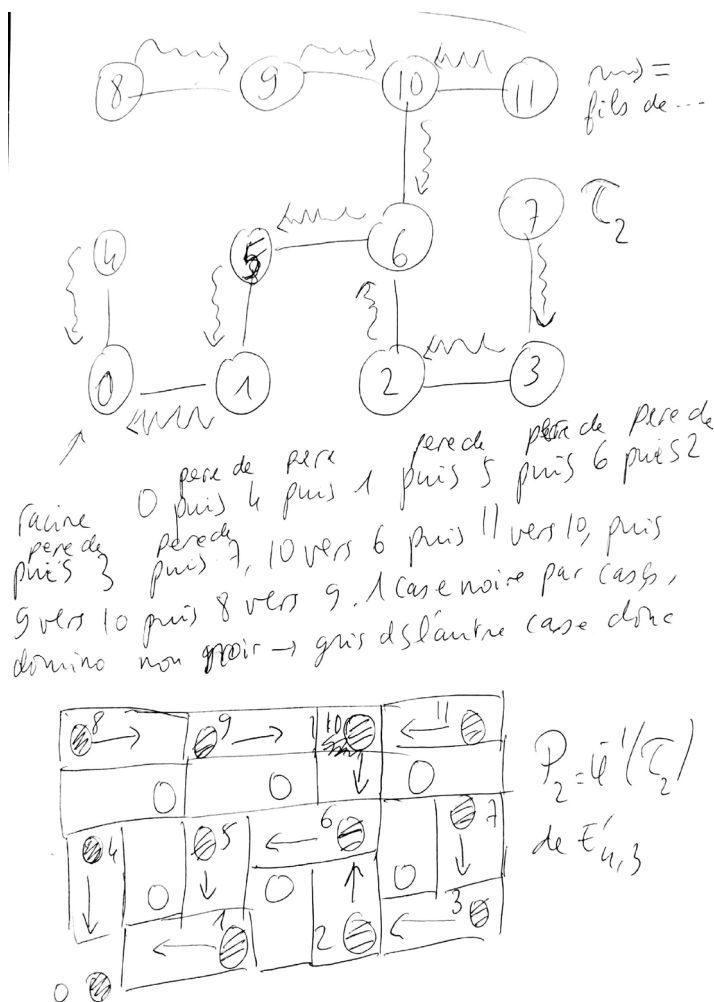
Commentaires : *Question de difficulté standard.*

Q21.



Commentaires : La seule difficulté est comprendre la fonction φ (conservation des dominos ayant un point noir !). Aucune explication n'est attendue par le correcteur. La réponse est tout ou rien (rien = tout sauf la totalité de juste).

Q22. Il faut commencer par rajouter les sommets gris, puis construire par étape l'arbre à partir des données fournies. On place les flèches qui relient les enfants aux parents. Voici le dessin correspondant à la page suivante.



Commentaires : Question très sélective car elle demande beaucoup plus de réflexion de la part des candidats.

Q23. Le père d'un sommet de numéro s est le sommet vers lequel pointe la flèche qui part de s donc il s'agit des sommets $s+1$ (est) ou $s-1$ (ouest) ou $s+p$ (nord) ou $s-p$ (sud).

Commentaires :

Question sélective car elle demande du candidat une compréhension de ces graphes et pavages. Pour ceux qui l'ont saisi, la réponse est simple.

Q24. Il y a p sommets par ligne donc on effectue la division euclidienne de s par p pour obtenir les coordonnées i et j que l'on doit doubler car un domino est associé à deux sommets initiaux.


```
let coord_noire s =
  let x = s/p and y = s mod p in 2*y,2*x;;
```

Commentaires : Si la programmation est aisée, la difficulté pour le candidat est de concevoir la formule mathématique. Ne pas hésitez à la deviner en observant les numéros des différents points sur les figures proposées par le sujet.

Q25.

```
let sommet_direction s d = match d with
|N -> if s+p <= p*q-1 then s+p else (-1)
|E -> if s+1 <= p*q-1 then s+1 else (-1)
|S -> if s >= p then s-p else (-1)
|W -> if s >= 1 then s-1 else (-1);;
```

Commentaires : Question simple ... pour les candidats ayant répondu à la question 23.

Q26.

```
let phi pavage =
  let n = p*q in
  let parent = Array.make n (-1) in
  for i = 0 to n-1 do
    let x , y = coord_noire i in
    parent.(i) <- sommet_direction i (pavage.(y).(x))
  done; parent;;
```

Commentaires : Question de difficulté standard qui peut être traitées sans avoir réussi les questions précédentes ! Pensez-y, cela arrive régulièrement aux écrits.

Q27.

```
let ajout_arete g s =
  let aux num =
    g.(s) <- num::g.(s); if num = 0 then g.(num) <- s::g.(num)
  in
  let x,y=coord_grise s in
  aux (numero (x-2,y)); aux (numero (x+2,y));aux (numero (x,y
-2));aux (numero (x,y+2));;

let dual()=
  let n =(p-1)*(q-1) in let g = Array.make (n+1) [] in
  for s=1 to n do
    ajout_arete g s done;
  g;;
```

Commentaires : Question difficile, non pour la programmation, mais par le fait que le candidat soit apte à comprendre toutes ces notations et notions abstraites (graphe dual par exemple !) et coordonne tout cela. Au final, question destinée uniquement aux bonnes copies.

Q28. Commençons par un exemple. Dans le graphe $G_{4,3}^*$, toute face contient exactement un sommet de $G_{4,3}$ (qui est son milieu géométrique si on est dans le quadrillage). Toute arête a^* de $G_{4,3}^*$ est un côté commun à deux faces de $G_{4,3}^*$ et chacun contient un unique sommet de $G_{4,3}$ donc en reliant ces deux sommets, on obtient une unique arête a de $G_{4,3}$. On étend cela à tous les graphes $G_{p,q}$ immédiatement.

Commentaires : Question difficile car elle demande une bonne compréhension par le candidat de la construction du graphe dual.

Q29. Commençons par un exemple. Considérons le cycle

$$c = 1 \rightarrow 5 \rightarrow 6 \rightarrow 2 \rightarrow 1$$

dans le graphe $G_{4,3}$ en posant

$$B = \{\{1, 5\}, \{5, 6\}, \{6, 2\}, \{2, 1\}\}$$

Son parcours enserre le sommet 2 de $G_{4,3}^*$. Si 0 dans $G_{4,3}^*$ est dans la même composante connexe que 2 dans $(G_{4,3}^*, B^*)$ alors il existe une succession d'arêtes qui amène de 0 à 2 dans $(G_{4,3}^*, B^*)$. Or chacune de ces arêtes de B^* sont toutes les arêtes de $G_{4,3}^*$ sauf celles de

$$\{\{1, 5\}^*, \{5, 6\}^*, \{6, 2\}^*, \{2, 1\}^*\} = \{\{1, 2\}, \{2, 5\}, \{2, 3\}^*, \{0, 2\}^*\}$$

Il est impossible de passer de 0 à 2 dans $G_{4,3}^*$ sans jamais passer par ces arêtes dans $G_{4,3}^*$ car ces arêtes sont les seules qui mènent à 2! Ainsi, $(G_{4,3}^*, B^*)$ n'est pas connexe. Etendons ce raisonnement au cas général.

Considérons un cycle

$$c : s = s_0 \rightarrow s_1 \rightarrow \dots \rightarrow s_{p-1} \rightarrow s_p = s$$

dans (S_n, B) donc chaque arête (s_i, s_{i+1}) appartient à B . Les arêtes $(s_{p-1}, s_p) = (s_{p-1}, s)$ et $(s_0, s_1) = (s, s_1)$ appartiennent à B et définissent un carré dans le graphe S_n qui est contenu dans $G_{p,q}$. Soit t l'unique sommet de $G_{p,q}^*$ qui soit son centre (grisée). Toute arête de $G_{p,q}^*$ dont une extrémité est t est le dual d'une des arêtes de ce carré ou une multi-arête $\{0, t\}$ dans $G_{p,q}^*$ (donc n'appartient pas à B^* puisque les arêtes a^* de B^* sont les arêtes a de $G_{p,q}$ qui n'appartiennent pas à B !) ce qui empêche de créer un chemin de 0 à t donc (S_n^*, B^*) est non connexe.

Commentaires : Question très difficile éventuellement réussie uniquement par les meilleures copies.

Q30. Par contraposée de la question précédente, on peut affirmer que (S_n^*, B^*) est connexe alors (S_n, B) est acyclique. Par conséquent, si (S_n^*, B^*) est un arbre couvrant de $G_{p,q}^*$ alors (S_n^*, B^*) est connexe donc (S_n, B) est acyclique. Notons :

- $n = pq$ le nombre de sommets de $G_{p,q}$ (donc de S_n) et $n^* = (p-1)(q-1)+1$ le nombre de sommets de $G_{p,q}^*$;

- $m = p(q-1) + q(p-1)$ est le nombre d'arêtes de $G_{p,q}$, m_B celui de B et m_B^* celui de B^* .

On a :

$$m_B^* = m - m_B$$

(puisque $a^* \in B^* \Leftrightarrow a \notin B$). Comme (S_n^*, B^*) est un arbre, d'après la question 10, on a

$$\begin{aligned} m_B^* &= n^* - 1 \Leftrightarrow m - m_B = n^* - 1 \Leftrightarrow m_B = m - n^* + 1 \\ &= p(q-1) + q(p-1) - (p-1)(q-1) - 1 + 1 \\ &= pq - 1 \text{ (après développement)} = n - 1 \end{aligned}$$

donc, comme (S_n, B) est connexe et $m_B = n - 1$, la question 10 montre que (S_n, B) est un arbre dont les sommets sont tous ceux de $G_{p,q}$, c'est bien un arbre couvrant.

Comme $G_{p,q}^*$ a pour dual $G_{p,q}$ (ou s'identifie donc on identifie S_n^* à S_n) le dual de B^* est B (ou s'identifie), $(n^*)^* = n$, S_n^* a pour dual, en appliquant le résultat que nous venons de montrer à $S_n = S_n^*$, on obtient que (S_n, B) est un arbre couvrant alors (S_n^*, B^*) l'est d'où l'équivalence.

Commentaires : *Même remarque que précédemment.*

Q31.

```
let vers_couple parent =
  let m = p*(q-1)+q*(p-1) and n = Array.length parent in
  let b = Array.make m false in
  for i=0 to m-1 do
    let a,c = sommets (p,q) i in
    if parent.(a) = c || parent.(c) = a then b.(i) <- true
  done;
  let rec dernier i acc= match i with
    | i when i=n -> acc
    | i -> if parent.(i) = -1 then dernier (i+1) i else dernier
          (i+1) acc
  in let r = dernier 0 0
  in r,b;;
```

Commentaires : *Bravo aux candidats ayant eu le temps et la capacité de traiter cette question. Elle est accessible ... mais il faut être présent pour l'écrire :-).*

Q32. On effectue un parcours en profondeur à partir de la racine r . On marque tout sommet s rencontré (donc on marque r au début) puis, pour tous ses voisins t dans la liste $g.(s)$ de ses voisins, si t est non marqué, on le déclare comme parent de s puis on relance parcours depuis t .

Cela n'est pas possible pour le graphe dual car les arêtes ne sont plus uniques entre deux sommets (l'algorithme convient pour les arêtes usuelles c'est-à-dire celles duales d'une arête initiale, pas pour celle qui relie 0 à un sommet non nul).

Commentaires : *Félicitations aux rares candidats ayant correctement traité cette question.*

Q33.

```

let vers_parent (r,b) =
  let n = p*q in let parent = Array.make n (-2) in
  parent.(r) <- -1;
  let rec parcourt s =
    let aux t =
      if parent.(t) = -2 then
        let a = rang (p,q) (min s t, max s t) in
        if b.(a) then (parent.(t) <- s ; parcourt t)
    in
    let rec iter liste_voisins = match liste_voisins with
      | [] -> ()
      | t::q -> aux t ; iter q
    in iter (g.(s))
  in parcourt r ; parent;;

```

Commentaires : *Même remarque que précédemment.*

Q34. La fonction *vers_couple* a une complexité en $O(n + m)$ (la boucle s'itère m fois, chaque opération dans la boucle ayant un coût constant et elles sont en nombre bornée, la fonction *dernier* s'exécute au plus n fois, chacun de ses appels entraîne un nombre borné d'opérations à coût constant) donc en $O(m)$ (car $n \leq m$).

La fonction *vers_parent* (r, b) a une complexité en $O(m)$. En effet, elle parcourt toutes les arêtes une seule fois (si les deux extrémités sont marquées, l'arête n'est pas traitée, tout sommet rencontré est marqué) et toutes les opérations ont un coût borné. On peut aussi faire un raisonnement plus local : la fonction *parcours* s a une complexité en $O(d_s)$, où d_s est le nombre de voisins de s (la longueur de $g.(s)$), car elle s'appelle au plus d_s fois et les opérations hormis ces appels sont en nombre borné et chaque opération a un coût constant. On itère cela sur tous les sommets et comme $\sum_s d_s = 2m$ (chaque arête $\{s, t\}$ est vu deux fois, une fois dans d_s en visitant le sommet t et une fois dans d_t en visitant le sommet s). Le coût total est donc en $\sum_s O(d_s) = O\left(\sum_s d_s\right) = O(m)$.

Commentaires : *Même remarque que précédemment*

Q35.

```

let arbre_dual parent =
  let (r,b) = vers_couple parent in
  let m = Array.length b in
  let b_etoile = Array.make m false in
  for i=0 to m-1 do
    b_etoile.(i) <- not (b.(i))
  done;
  vers_parent_etoile (r,b_etoile);;

```

Commentaires : *Même remarque que précédemment*

Q36, 37 et 38.

Commentaires : *Ces questions sont inaccessibles à un candidat de CPGE (sauf cas largement exceptionnel) à la fois en terme de temps (mais certain(e)s sont rapides et efficaces) et surtout en terme de niveau. Il est déraisonnable de travailler ces questions à ce niveau donc je préfère ne pas donner un corrigé qui est illusoire.*

A2021 – INFO



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2021

ÉPREUVE D'INFORMATIQUE COMMUNE

Durée de l'épreuve : 2 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

Cette épreuve est commune aux candidats des filières MP, PC et PSI.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE COMMUNE

L'énoncé de cette épreuve comporte 10 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Marchons, marchons, marchons...

Ce sujet propose d'appliquer des techniques informatiques à l'étude de trois marches de nature différente :

- une marche concrète (partie I - Randonnée)
- une marche stochastique (partie II - Mouvement brownien d'une petite particule)
- une marche auto-évitante (partie III - Marche auto-évitante)

Les trois parties sont indépendantes. L'annexe à la fin du sujet (pages 8 à 10) fournit les canevas de code en Python que vous devrez reprendre dans votre copie pour répondre aux questions de programmation.

Partie I. Randonnée

Lors de la préparation d'une randonnée, une accompagnatrice doit prendre en compte les exigences des participants. Elle dispose d'informations rassemblées dans deux tables d'une base de données :

- la table **Rando** décrit les randonnées possibles – la clef primaire entière **rid**, son nom, le niveau de difficulté du parcours (entier entre 1 et 5), le dénivelé (en mètres), la durée moyenne (en minutes) :

rid	rnom	diff	deniv	duree
1	La belle des champs	1	20	30
2	Lac de Castellane	4	650	150
3	Le tour du mont	2	200	120
4	Les crêtes de la mort	5	1200	360
5	Yukon Ho!	3	700	210
...	...	...	...	...

- la table **Participant** décrit les randonneurs – la clef primaire entière **pid**, le nom du randonneur, son année de naissance, le niveau de difficulté maximum de ses randonnées :

pid	pnom	ne	diff_max
1	Calvin	2014	2
2	Hobbes	2015	2
3	Susie	2014	2
4	Rosalyn	2001	4
...	...	...	...

Donner une requête SQL sur cette base pour :

- ❑ **Q1** – Compter le nombre de participants nés entre 1999 et 2003 inclus.
- ❑ **Q2** – Calculer la durée moyenne des randonnées pour chaque niveau de difficulté.
- ❑ **Q3** – Extraire le nom des participants pour lesquels la randonnée n°42 est trop difficile.
- ❑ **Q4** – Extraire les clés primaires des randonnées qui ont un ou des homonymes (nom identique et clé primaire distincte), sans redondance.

L'accompagnatrice a activé le suivi d'une randonnée par géolocalisation satellitaire et souhaite obtenir quelques propriétés de cette randonnée une fois celle-ci effectuée. Elle a exporté les données au format texte CSV (*comma-separated values* – valeurs séparées par des virgules) dans un fichier nommé **suivi_rando.csv** : la première ligne annonce le format, les suivantes donnent les positions dans l'ordre chronologique.

Voici le début de ce fichier pour une randonnée partant de Valmorel, en Savoie, un bel après-midi d'été :

```
lat(°),long(°),height(m),time(s)
45.461516,6.44461,1315.221,1597496965
45.461448,6.444426,1315.702,1597496980
45.461383,6.444239,1316.182,1597496995
45.461641,6.444035,1316.663,1597496710
45.461534,6.443879,1317.144,1597496725
45.461595,6.4437,1317.634,1597496740
45.461562,6.443521,1318.105,1597496755
...
```

Le module `math` de Python fournit les fonctions `asin`, `sin`, `cos`, `sqrt` et `radians`. Cette dernière convertit des degrés en radians. La documentation donne aussi des éléments de manipulation de fichiers textuels :

`fichier = open(nom_fichier, mode)` ouvre le fichier, en lecture si `mode` est `"r"`.
`ligne = fichier.readline()` récupère la ligne suivante de `fichier` ouvert en lecture avec `open`.
`lignes = fichier.readlines()` donne la liste des lignes suivantes.
`fichier.close()` ferme `fichier`, ouvert avec `open`, après son utilisation.
`ligne.split(sep)` découpe la chaîne de caractères `ligne` selon le séparateur `sep` : si `ligne` vaut `"42,43,44"`, alors `ligne.split(",")` renvoie la liste `["42", "43", "44"]`.

On souhaite exploiter le fichier de suivi d'une randonnée – supposé préalablement placé dans le répertoire de travail – pour obtenir une liste `coords` des listes de 4 flottants (latitude, longitude, altitude, temps) représentant les points de passage collectés lors de la randonnée.

À partir du canevas fourni en annexe et en ajoutant les `import` nécessaires :

❑ **Q5** – Implémenter la fonction `importe_rando(nom_fichier)` qui réalise cette importation en retournant la liste souhaitée, par exemple en utilisant certaines des fonctions ci-dessus.

On suppose maintenant l'importation effectuée dans `coords`, avec au moins deux points d'instant distincts.

❑ **Q6** – Implémenter la fonction `plus_haut(coords)` qui renvoie la liste (latitude, longitude) du point le plus haut de la randonnée.

❑ **Q7** – Implémenter la fonction `deniveles(coords)` qui calcule les dénivelés cumulés positif et négatif (en mètres) de la randonnée, sous forme d'une liste de deux flottants. Le dénivelé positif est la somme des variations d'altitude positives sur le chemin, et inversement pour le dénivelé négatif.

On souhaite évaluer de manière approchée la distance parcourue lors d'une randonnée. On suppose la Terre parfaitement sphérique de rayon $R_T = 6371$ km au niveau de la mer. On utilise la formule de haversine pour calculer la distance d du grand cercle sur une sphère de rayon r entre deux points de coordonnées (latitude, longitude) (φ_1, λ_1) et (φ_2, λ_2) :

$$d = 2r \arcsin \left(\sqrt{\sin^2 \left(\frac{\varphi_2 - \varphi_1}{2} \right) + \cos(\varphi_1) \cos(\varphi_2) \sin^2 \left(\frac{\lambda_2 - \lambda_1}{2} \right)} \right)$$

On prendra en compte l'altitude moyenne de l'arc, que l'on complètera pour la variation d'altitude par la formule de Pythagore, en assimilant la portion de cercle à un segment droit perpendiculaire à la verticale.

❑ **Q8** – Implémenter la fonction `distance(c1, c2)` qui calcule avec cette approche la distance en mètres entre deux points de passage. On décomposera obligatoirement les formules pour en améliorer la lisibilité.

❑ **Q9** – Implémenter la fonction `distance_totale(coords)` qui calcule la distance en mètres parcourue au cours d'une randonnée.

Partie II. Mouvement brownien d'une petite particule

De petites particules en suspension dans un liquide se déplacent spontanément sous l'effet de l'agitation thermique du milieu environnant, donnant ainsi naissance à des trajectoires apparemment chaotiques et peu régulières. Ce phénomène est à la base de la vie : l'agitation incessante des protéines permet aux réactions biochimiques de se produire à l'intérieur de nos cellules. Il a été observé expérimentalement en 1827 sur des grains de pollen en suspension dans l'eau par le botaniste ROBERT BROWN, d'où le nom de mouvement brownien. Son étude théorique par ALBERT EINSTEIN en 1905 a permis au physicien JEAN PERRIN d'estimer la valeur du nombre d'Avogadro dans une série d'expériences menées entre 1907 et 1909.

Dans cette partie, on s'intéresse à un modèle du mouvement brownien proposé par PAUL LANGEVIN en 1908. Dans ce modèle, la particule étudiée est supposée soumise à deux actions de la part du fluide :

- une force de frottement fluide $\vec{f}_F = -\alpha \vec{v}$;
- une force $\vec{f}_B$ aléatoire simulant l'action désordonnée des molécules d'eau sur la particule.

Le mouvement de cette particule est donc régi par l'équation différentielle :

$$\frac{d\vec{v}}{dt} = -\frac{\alpha}{m} \vec{v} + \frac{\vec{f}_B}{m}$$

Pour faire une simulation en deux dimensions, on prend une particule de masse $m = 10^{-6}$ kg, on attribue à α la valeur 10^{-5} kg/s, on suppose enfin que $\vec{f}_B$ change à chaque pas d'intégration, avec une direction isotrope aléatoire (l'angle suit une loi de probabilité uniforme) et une norme qui est la valeur absolue d'une variable aléatoire suivant une loi de probabilité gaussienne (loi normale) d'espérance μ nulle et d'écart-type $\sigma = 10^{-8}$ N.

On simule le vecteur d'état $E = (x, y, \dot{x}, \dot{y})$ de la particule en intégrant $\dot{E} = (\dot{x}, \dot{y}, \ddot{x}, \ddot{y})$ selon la méthode d'Euler. E et $\dot{E}$ seront chacun représentés par une liste de 4 flottants.

L'instruction `assert expression` de Python vérifie la véracité d'une expression booléenne et interrompt brutalement l'exécution du programme si ce n'est pas le cas. Elle permet de vérifier très simplement une précondition ou un invariant, comme illustré à la ligne 10 du canevas.

Le module `random` fournit la fonction `uniform(bi, bs)` qui renvoie un flottant aléatoire entre les valeurs `bi` et `bs` incluses en utilisant une densité de probabilité uniforme ainsi que la fonction `gauss(mu, sigma)` qui renvoie un flottant aléatoire en utilisant une densité de probabilité gaussienne d'espérance `mu` et d'écart-type `sigma`.

Le module `math` fournit enfin les fonctions `cos` et `sin` (en radians), la fonction `sqrt` et la constante `pi`. La fonction `abs` est directement disponible.

En utilisant le canevas fourni en annexe et en ajoutant les `import` nécessaires :

□ **Q10** – Implémenter la fonction `vma(v1, a, v2)` (multiplication-addition sur des vecteurs) avec `v1` et `v2` des listes de flottants et `a` un scalaire flottant, qui renvoie une nouvelle liste de flottants correspondant à :

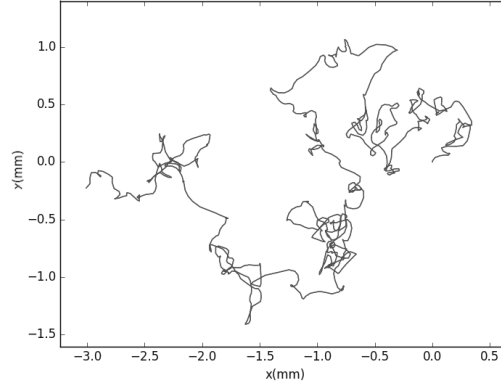
$$\vec{v} = \vec{v}_1 + a \vec{v}_2$$

La fonction vérifiera que les deux listes en entrée sont de même longueur avec `assert`.

□ **Q11** – Implémenter la fonction `derive(E)` qui renvoie la dérivée du vecteur d'état passé en paramètre d'après l'équation différentielle décrite en introduction de la partie.

□ **Q12** – Implémenter la fonction `euler(E0, dt, n)` pour résoudre numériquement l'équation différentielle par la méthode d'Euler avec `E0` le vecteur d'état initial, `dt` le pas d'intégration et `n` le nombre de pas de la simulation. La fonction renvoie la liste des $n + 1$ vecteurs d'état.

À titre d'exemple, voici le tracé d'une trajectoire obtenue par cette méthode :



Partie III. Marche auto-évitante

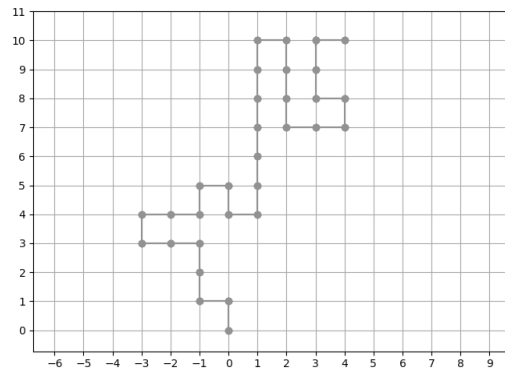
Une marche auto-évitante est un processus au cours duquel un point décrit un chemin auto-évitant, c'est-à-dire qui ne se recoupe jamais lui-même. Ce modèle peut servir lors de la modélisation d'une chaîne polymère. En effet, deux monomères distincts de la chaîne ne peuvent pas se trouver à deux positions identiques pour des raisons d'encombrement stérique.

Dans ce sujet, on appellera chemin auto-évitant (CAE) de longueur n tout ensemble de $n + 1$ points $P_i \in \mathbb{Z}^2$ pour $0 \leq i \leq n$ tels que :

- $\forall i \quad \|P_{i+1} - P_i\| = 1$
- $\forall (i, j) \quad i \neq j \Rightarrow P_i \neq P_j$

Chaque point du chemin sera représenté par une liste à deux éléments entiers précisant les deux coordonnées (par exemple $P_i = (5, -4)$ est représenté par la liste `[5, -4]`). Le chemin lui-même est constitué de la liste des points, dans l'ordre dans lequel ils sont atteints. Les codes Python produits dans cette partie devront manipuler exclusivement des coordonnées entières.

Voici un exemple de représentation graphique de CAE de longueur 30 à partir de $(0, 0)$:



On s'intéresse dans un premier temps à une méthode naïve pour générer un chemin auto-évitant de longueur n sur une grille carrée. La méthode adoptée est une approche gloutonne :

1. Le premier point est choisi à l'origine : $P_0 = (0, 0)$.
2. En chaque position atteinte par le chemin, on recense les positions voisines accessibles pour le pas suivant et on en sélectionne une au hasard. En l'absence de positions accessibles l'algorithme échoue.
3. On itère l'étape 2 jusqu'à ce que le chemin possède la longueur désirée ou échoue.

Le module `random` fournit la fonction `randrange(a, b)` qui renvoie un entier compris entre a et $b - 1$ inclus, ainsi que la fonction `choice(L)` qui renvoie l'un des éléments de la liste L , dans les deux cas choisis aléatoirement avec une probabilité uniforme.

L'expression `x in L` est une expression booléenne qui vaut `True` si x est l'un des éléments de L et `False` dans le cas contraire. On supposera que la méthode employée pour évaluer cette expression sur une liste est une recherche séquentielle.

La valeur spéciale `None` est utilisée en Python pour représenter une valeur invalide, inconnue ou indéfinie. L'expression booléenne `v is None` indique si la valeur de v est cette valeur spéciale.

En utilisant le canevas fourni en annexe et en ajoutant les `import` nécessaires :

□ **Q13** – Implémenter la fonction `positions_possibles(p, atteints)` qui construit la liste des positions suivantes possibles à partir du point p . La liste `atteints` contient les points déjà atteints par le chemin.

□ **Q14** – Mettre en évidence graphiquement un exemple de CAE le plus court possible pour lequel, à une étape donnée, la fonction `positions_possibles` va renvoyer une liste vide. En prenant en compte les symétries, combien de tels chemins distincts existent pour cette longueur minimale ?

□ **Q15** – Implémenter la fonction `genere_chemin_naif(n)` qui construit la liste des points représentant le chemin auto-évitant de longueur n et renvoie le résultat, ou bien renvoie la valeur spéciale `None` si à une étape `positions_possibles` renvoie une liste vide.

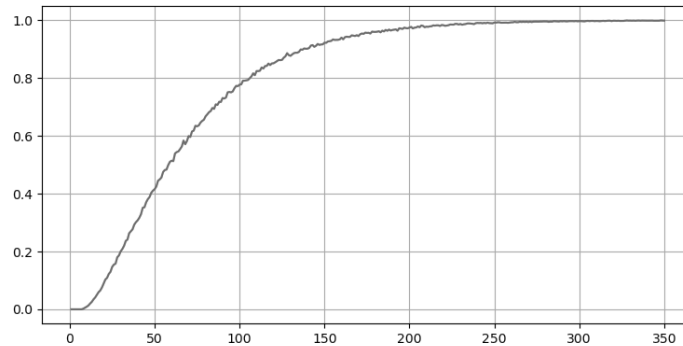
□ **Q16** – Évaluer avec soin la complexité temporelle asymptotique dans le pire des cas de la fonction `genere_chemin_naif(n)` en fonction de n , en supposant que la fonction ne renvoie pas `None`.

Une personne curieuse et patiente a écrit le code suivant :

```

1  from chemin import genere_chemin_naif
2
3  N, M, L, P = 10000, 351, [], []
4
5  for n in range(1, M):
6      nb = 0
7      for i in range(N):
8          chemin = genere_chemin_naif(n)
9          if chemin is None:
10             nb += 1
11         L.append(n)
12         P.append(nb / N)
13
14  import matplotlib.pyplot as plt
15  plt.plot(L, P)
16  plt.grid()
17  plt.show()
```

Après un long moment, elle a obtenu le graphique suivant, volontairement laissé sans étiquettes d'axes :

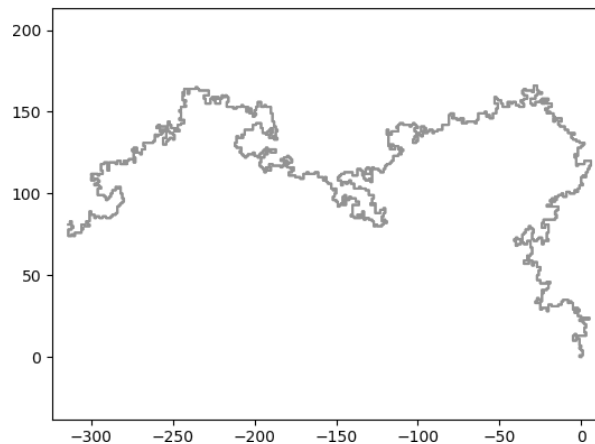


□ **Q17** – Décrire ce que représente ce graphique et interpréter sa signification pour la méthode naïve.

Afin d'éviter les inconvénients de la méthode précédente, on s'intéresse à une solution différente nommée *méthode du pivot*, proposée par Moti Lal en 1969. Son principe est le suivant :

1. On part d'un chemin auto-évitant arbitraire de longueur n . Ici, on choisira une initialisation très simple, le chemin droit $[[0, 0], [1, 0], [2, 0], \dots, [n, 0]]$.
2. On sélectionne au hasard un point, nommé pivot, entre le second et l'avant-dernier point du chemin, et un angle aléatoire de rotation parmi π , $\frac{\pi}{2}$ et $-\frac{\pi}{2}$.
3. On laisse les points avant le pivot inchangés et on fait subir à l'ensemble des points situés strictement après le pivot une rotation ayant pour centre le pivot et pour angle, l'angle choisi à l'étape 2 ci-dessus.
4. Si le chemin ainsi obtenu est auto-évitant, on le garde. Sinon, on reprend à l'étape 2 la sélection d'un pivot et d'un angle, jusqu'à en trouver une paire qui conviennent.
5. On répète les étapes 2 à 4 un certain nombre de fois. Le choix du nombre minimal de rotations à effectuer pour obtenir un chemin non corrélé au chemin initial est laissé de côté dans ce sujet.

Cette méthode permet de générer de manière efficace des marches auto-évitantes de plusieurs milliers de points, comme la marche ci-dessous de longueur 2000 :



Dans cet algorithme, une étape importante est la vérification qu'un chemin donné est auto-évitant. Pour vérifier cela on pourrait bien sûr s'y prendre comme dans la fonction `positions_possibles`, mais on adopte une méthode différente avec une fonction `est_CAE(chemin)` qui trie les points du chemin puis met à profit ce tri pour vérifier efficacement que le chemin est auto-évitant.

On utilise pour cela la fonction `sorted` qui renvoie une nouvelle liste triée par ordre croissant. Elle fonctionne sur tout type d'éléments disposant d'une relation d'ordre, y compris des listes pour lesquelles l'ordre lexicographique (ordre du premier élément, en cas d'égalité du second, etc.) est appliqué. On suppose de plus que la complexité temporelle asymptotique dans le pire des cas de cette fonction est la meilleure possible.

□ **Q18** – Rappeler, sans la justifier, la complexité temporelle asymptotique dans le pire des cas attendue de `sorted` en fonction de la longueur de la liste à trier. Donner le nom d'un algorithme possible pour son implémentation.

En utilisant le canevas fourni en annexe et en ajoutant les `import` nécessaires :

□ **Q19** – Implémenter la fonction `est_CAE(chemin)` qui vérifie si un chemin est auto-évitant en se basant sur `sorted` et renvoie un résultat booléen. Elle devra ne pas être de complexité temporelle asymptotique dans le pire des cas supérieure à la fonction `sorted` : vous prouvez ce dernier point.

□ **Q20** – Implémenter la fonction `rot(p, q, a)` qui renvoie le point image du point `q` par la rotation de centre `p` et d'angle défini par la valeur de `a` : 0 pour π , 1 pour $\frac{\pi}{2}$, 2 pour $-\frac{\pi}{2}$.

□ **Q21** – Implémenter la fonction `rotation(chemin, i_piv, a)` qui renvoie un nouveau chemin identique à `chemin` jusqu'au pivot d'indice `i_piv`, et ayant subi une rotation de `a` autour du pivot (même codage que la fonction précédente) sur la partie strictement après le pivot.

□ **Q22** – Implémenter la fonction `genere_chemin_pivot(n, n_rot)` permettant de générer un chemin auto-évitant de longueur `n` en appliquant `n_rot` rotations. L'application d'une rotation peut nécessiter plusieurs tentatives.

□ **Q23** – On considère un pivot, son point précédent et son point suivant. Quel est l'impact prévisible sur les rotations admissibles ? Suggérer un moyen de prendre en compte cette propriété pour améliorer l'algorithme.

Fin de l'énoncé

Annexe canevas de codes Python sur les pages suivantes

Annexe : canevas de codes Python

— Partie I : Randonnée

```

1  # import Python à compléter...
2
3  # importation du fichier d'une randonnée
4  def importe_rando(nom_fichier):
5      # À compléter...
6
7  coords = importe_rando("suivi_rando.csv")
8
9  # donne le point (latitude, longitude) le plus haut de la randonnée
10 def plus_haut(coords):
11     # À compléter...
12
13 print("point le plus haut", plus_haut(coords))
14 # exemple : point le plus haut [45.461451, 6.443064]
15
16 # calcul des dénivelés positif et négatif de la randonnée
17 def deniveles(coords):
18     # À compléter...
19
20 print("dénivelés", deniveles(coords))
21 # exemple : denivelés [4.059999999999945, -1.175999999999309]
22
23 RT = 6371 # rayon moyen volumétrique de la Terre en km
24
25 # distance entre deux points
26 def distance(c1, c2):
27     # À compléter...
28
29 print("premier intervalle", distance(coords[0], coords[1]), "m")
30 # exemple : premier intervalle 16.230964254992816 m
31
32 # distance totale de la randonnée
33 def distance_totale(coords):
34     # À compléter...
35
36 print("distance parcourue", distance_totale(coords), "m")
37 # exemple : distance parcourue 187.9700904658368 m

```

— Partie II : Mouvement brownien

```

1  # import Python à compléter...
2
3  # paramètres physiques
4  MU = 0.0      # N
5  SIGMA = 1E-8  # N
6  M = 1E-6      # kg
7  ALPHA = 1E-5  # kg/s
8
9  # vérification des hypothèses sur les paramètres
10 assert MU >= 0 and SIGMA > 0 and M > 0 and ALPHA > 0
11
12 # multiplication-addition vectorielle
13 def vma(v1, a, v2):
14     # À compléter...
15
16 # dérivée du vecteur d'état
17 def derive(E):
18     # À compléter...
19
20 # intégration par la méthode d'Euler
21 def euler(E0, dt, n):
22     Es = [ E0 ]
23     # À compléter...
24     return Es
25
26 # simulation
27 DT = 0.002    # durée du pas en secondes
28 N = 5000      # nombre de pas
29
30 Es = euler([ 0.0, 0.0, 0.0, 0.0 ], DT, N)
31
32 print("trajectoire", Es)

```

— Partie III : Chemin auto-évitant - méthode naïve

```

1  # import Python à compléter...
2
3  # positions auto-évitantes suivantes possibles
4  def positions_possibles(p, atteints):
5      possibles = []
6      # À compléter...
7      return possibles
8
9  # génération gloutonne d'un chemin de longueur n
10 # renvoie None en cas d'échec
11 def genere_chemin_naif(n):
12     chemin = [ [ 0, 0 ] ] # on part de l'origine
13     # À compléter...
14     return chemin
15
16 N = 10
17 print("chemin", genere_chemin_naif(N))

```

— Partie III : Chemin auto-évitant - méthode du pivot

```

1  # import Python à compléter...
2
3  # vérifie si un chemin est CAE
4  def est_CAE(chemin):
5      # À compléter...
6
7  # calcule la rotation de q autour de p selon a :
8  # Pi si a vaut 0, Pi/2 si a vaut 1, -Pi/2 si a vaut 2
9  def rot(p, q, a):
10     # À compléter...
11
12 # renvoie le chemin dont les points après i_pivot
13 # ont subi une rotation a codé comme précédemment
14 def rotation(chemin, i_pivot, a):
15     # À compléter...
16
17 # génère un chemin de longueur n (donc n+1 points)
18 def genere_chemin_pivot(n, n_rot):
19     # À compléter...
20
21 N, A = 1000, 2.3
22 print("chemin", genere_chemin_pivot(N, int( A * N )))

```


SOLUTION

Q1.

```
SELECT COUNT(*) FROM Participant WHERE ne >= 1999 AND ne <= 2003;
```

Commentaires : Question élémentaire sans difficulté particulière même s'il semble qu'un nombre significatif de candidats oublient le SELECT ! Il ne faut pas commettre l'erreur d'écrire " 1999 <= ne <= 2003 " au lieu de " ne >= 1999 AND ne <= 2003 ".

Q2.

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

Commentaires : Question de niveau standard même si elle s'avère en pratique sélective. En effet, un nombre très significatif de candidats maîtrisent mal (ou pas) la fonction d'agrégation AVG (moyenne). À travailler impérativement (car grand classique et sans difficulté insurmontable).

Q3. Une version sans jointure

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

Une version avec jointure.

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

Commentaires : Question beaucoup plus délicate qui nécessite soit une bonne maîtrise des requêtes et sous-requêtes ou de la notion de jointure. Dans ce dernier cas, un nombre important de candidats n'ont pas su gérer correctement la condition de jointure (ils se sont perdus dans les différents attributs des tables)

Q4.

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

Commentaires : Question difficile à destination des candidats les plus à l'aise en SQL.

Q5.

```
def importe_rando_1(nom_fichier):
    temp = []
    fichier = open(nom_fichier, "r")
    fichier.readline()
    for ligne in fichier.readlines():
        temp.append( [ float(c) for c in ligne.split(",") ] )
    fichier.close()
    return temp
```

Commentaires : Question très sélective. Les candidats ayant déjà travaillé les commandes de manipulation de fichiers (open, readline, etc) ont généralement bien réussi cette question (à des détails près comme l'utilisation des commandes float et split). Tous les autres ont écrit des horreurs car ils ne comprenaient pas la philosophie de ces commandes. N'hésitez à travailler un peu ces notions (qui sont hors programme !) car Mines-Ponts est coutumier de ce type de questions (une à trois au maximum par sujet).

Q6.

```
def plus_haut_1(coords):
    lat,long,h,t = coords[0]
    for i in range(len(coords)):
        lat1,long1,h1,t1 = coords[i]
        if h1 > h:
            lat,long,h = lat1,long1,h1
    return [lat,long]
```

Commentaires : Question sans grande difficulté hormis qu'il faut songer à se rappeler qu'un élément de coords n'est pas un nombre (donc pas de fonction max !) donc il faut dépaqueter ses éléments (lat,long,h,t = coords[i]) ou extraire les bonnes données (par exemple coords[i][1]) ET SURTOUT ne pas initialiser le maximum courant (ici h) à 0 (tous les éléments peuvent être négatifs par exemple !) mais à un des éléments à comparer.

Q7.

```
def deniveles_1(coords):
    dplus,dmoins = 0, 0
    for i in range(len(coords) - 1):
        deniv = coords[i+1][2] - coords[i][2]
        if deniv > 0:
            dplus = dplus + deniv
        else:
            dmoins = dmoins + deniv
    return [ dplus, dmoins ]
```

Commentaires : *Question sans aucun difficulté (hormis distinguer la montée de la descente !).*

Q8.

```
import math as m

def distance(coord1, coord2):
    lat1, lon1, alt1, t1 = coord1
    lat2, lon2, alt2, t2 = coord2
    # conversion en radians
    phi1, lambda1 = m.radians(lat1), m.radians(lon1)
    phi2, lambda2 = m.radians(lat2), m.radians(lon2)
    alt = 0.5 * (alt1 + alt2)
    # calcule du grand cercle en convertissant en mètre le
    rayon
    dgc = 2.0 * (1000 * RT + alt) * m.asin( m.sqrt( m.sin(0.5
        * (phi2 - phi1)) ** 2 + m.cos(phi1) * m.cos(phi2) * m.
        sin(0.5 * (lambda2 - lambda1)) ** 2 ) )
    # variation d'altitude avec Pythagore
    return m.sqrt( dgc ** 2 + (alt2 - alt1) ** 2 )
```

Commentaires : *Question où la seule difficulté est de :*

- ne pas se perdre dans la formule ;
- penser aux diverses conversions ;
- penser à la variation d'altitude avec Pythagore.

Au final, pas mal de chausse-trappes. Il fallait être vigilant.

Q9.

```
def distance_totale_1(coords):
    d = 0
    for i in range(len(coords) - 1):
        d = d + distance(coords[i], coords[i+1])
    return d
```

Commentaires : *Question élémentaire.*

Q10.

```
def vma_1(v1, a, v2):
    return [ v1[i] + a * v2[i] for i in range(len(v1)) ]
```

Commentaires : Question qui posa des difficultés aux candidats ayant une maîtrise insuffisante en Python. Les listes manipulées sont des listes Python et non numpy. Par conséquent, la notation `l1+l2` correspond à la concaténation des deux listes `l1` et `l2`, non à l'addition des vecteurs associés ! Quant à la notation `a*l1`, elle ne veut rien dire en Python si `a` est un flottant et `l1` une liste. Pour les autres candidats, question élémentaire.

Q11.

```
import random as rd
import math as m
# version directe
def derive_1(E):
    vx, vy = E[2], E[3]
    angle = rd.uniform(0.0, m.pi)
    norme = rd.gauss(MU, SIGMA) / M
    C = - ALPHA / M
    return [ vx, vy, C*vx + m.cos(angle)*norme, C*vy + m.sin(
        angle)*norme ]
```

Commentaires : Question sélective. Un nombre important de candidats eurent du mal à manipuler correctement les générateurs aléatoires `uniforme` et `gauss`. L'écriture `" x = rd.gauss(m,s)*cos(rd.uniform(0,p), y=rd.gauss(m,s)*sin(rd.uniform(0,p)) "` ne produit pas le même résultat que `" x = rd.gauss(m,s)*cos(uniforme(0,p)), y = rd.gauss(m,s)*cos(uniforme(0,p)) "`. En effet, le second crée deux nombres aléatoires `r` et `a` puis les nombres `x = r cos(a)` et `y = r sin(a)` alors que le premier crée 4 nombres aléatoires `r1, r2, a1, a2` puis les nombres `x = r1 cos(a1)`, `y = r2 cos(a2)`. Chaque appel de ces fonctions aléatoires génère un nouveau nombre aléatoire (sans lien, théoriquement, avec les précédents). Cette erreur est rédibitoire (dans l'implémentation et ... dans la copie :-)).

Q12.

```
def euler(E0, dt, n):
    Es = [E0]
    for i in range(n):
        Es.append( vma(Es[-1], dt, derive(Es[-1])) )
    return Es
```

Commentaires : Question qui s'est avérée difficile par beaucoup de candidats. Normal, le schéma d'Euler venait de disparaître du programme d'informatique ! Pour ceux qui connaissait l'implémentation de ce schéma, un nombre important pensait manipuler des nombres et non des listes d'où les oublis des fonctions `vma` et `derive` ! Au final, peu d'heureux élus à cette question.

Q13.

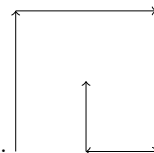
```
def positions_possibles(p, atteints):
    possibles = []
```

```

x, y = p
for q in [ [ x + 1, y ], [ x, y + 1 ], [ x - 1, y ], [ x,
            y - 1 ] ]:
    if not q in atteints:
        possibles.append(q)
return possibles

```

Commentaires : Question sans difficulté particulière et plutôt bien réussie par les candidats.



Q14. Il suffit de tourner en faisant un escargot le plus court possible.

Il y a 8 tels chemins (faire toutes les symétries possibles de ce dessin, 4 en tout puis les composées par les symétries centrales par rapport au centre de la figure).

Commentaires : La recherche d'un exemple fut classante (un nombre important y parvient). Par contre, le comptage, comme d'habitude, est très sélectif et peu de candidats parviennent à une réponse juste.

Q15.

```

import random as rnd
def genere_chemin_naif_1(n):
    chemin = [ [ 0, 0 ] ] # on part de l'origine
    for i in range(n):
        suivants = positions_possibles(chemin[-1], chemin)
        if len(suivants) == 0:
            return None
        chemin.append(rnd.choice(suivants))
    return chemin

```

Commentaires : Question assez bien réussie qui mérite un peu de réflexions (est-il possible de poursuivre, gestion de None et non renvoie d'une liste vide, sélection aléatoire d'un choix possible)

Q16. La boucle *for* sur *i* est exécutée *n* fois au maximum (sauf si le *if* est vérifié). A chaque itération, les instructions ont un coût constant hormis la fonction *choix_possibles*. Chaque appel de *choix_possibles(p,chemin)* a une complexité linéaire en la taille de la liste *chemin* (car il faut tester l'appartenance à cette liste d'un nombre d'au maximum 4 éléments). Tant que le *if* n'est pas réalisé, cette liste est de longueur *i* (la valeur de l'itérateur) donc la complexité maximale de *genere_chemin_naif(n)* est en

$$O(1 + 2 + \dots + n) = O(n^2).$$

Commentaires : Question sans difficulté particulière hormis ... que le sujet demande une justification et pas simplement la réponse. Il est attendu de parler de la complexité de choix possible, etc en additionnant correctement les complexités (ou en comptant le nombre d'itérations au maximum et la complexité maximale de chaque itération). Au final, la question s'est avérée très sélective.

Q17. A part de $n = 300$, la probabilité que la fonction `genere_chemin_naif(n)` échoue est quasiment égal à 1. Autrement dit, il est quasiment impossible de générer des chemins de longueur au moins 300.

Commentaires : Comme souvent pour les questions d'interprétation, la question s'avère sélective. Un nombre important parle de pourcentage, ce n'est pas le cas. Il s'agit plutôt de fréquence (qui approxime une probabilité) ou d'une approximation de probabilité.

Q18. D'après le cours, l'algorithme de tri de meilleure complexité en $O(n \ln(n))$ où n est la taille de la liste en entrée (par exemple, le tri fusion).

Commentaires : Il s'agit d'une question de cours. Il est faux de penser que le tri rapide a une complexité en $O(n \ln(n))$, il est en $O(n^2)$.

Q19.

```
def est_CAE_1(chemin):
    tri = sorted(chemin)
    for i in range(len(tri)-1):
        if tri[i] == tri[i+1]:
            return False
    return True
```

Commentaires : Question sans difficulté ... si on n'oublie de trier préalablement la liste ! Elle a été bien réussie par la plupart des candidats.

Q20.

```
def rot(p, q, a):
    x, y = p
    dx, dy = q[0] - x, q[1] - y
    if a == 0:
        return [x - dx, y - dy]
    if a == 1:
        return [x - dy, y + dx]
    if a == 2:
        return [x + dy, y - dx]
```

Commentaires : Question sélective. Non pas du point de vue informatique, mais peu de candidats parviennent à déterminer les coordonnées d'un point $(x; y)$ après une rotation d'angle π (bien fait en général) ou $\pm \frac{\pi}{2}$ (rarement réussi). Il suffit pourtant de faire un dessin (comme pour les formules de symétries des fonctions trigonométriques sin et cos). Au final, peu de réponses correctes.

Q21. Une version avec boucles

```
def rotation_2(chemin, i_pivot, a):
    nouveau = []
    for i in range(i_pivot+1):
        nouveau.append(chemin[i])
    for i in range(i_pivot+1, len(chemin)):
        nouveau.append(rot(chemin[i_pivot], chemin[i], a))
    return nouveau
```

Une version avec liste en compréhension

```
def rotation_1(chemin, i_pivot, a):
    temp = chemin[:i_pivot+1]
    for p in chemin[i_pivot+1:]:
        temp.append(rot(chemin[i_pivot], p, a))
    return temp
```

Commentaires : Question sans difficulté particulière.

Q22.

```
def genere_chemin_pivot_1(n, n_rot):
    chemin = [ [ i, 0 ] for i in range(n+1) ]
    while n_rot > 0:
        i_pivot = rd.randrange(1, n)
        a = rd.randrange(3)
        essai = rotation(chemin, i_pivot, a)
        if est_CAE(essai):
            chemin = essai
            n_rot -= 1
    return chemin
```

Commentaires : Question qui demande un peu de recul du candidat et sa capacité à transcrire informatiquement la méthode proposée. Correctement traitée par les bonnes copies.

Q23. On dispose de trois rotations mais l'une d'elle ramène le point suivant sur le précédent ce qui amène un échec automatique. Pour contourner le problème, on choisit au hasard (uniforme) l'une des deux rotations possibles restantes.

Commentaires : Question à destination des meilleures copies qui ont bien compris les tenants et aboutissant de la stratégie proposée par le sujet.

A2021 – MATH I MP



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2021

PREMIÈRE ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 3 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES I - MP

L'énoncé de cette épreuve comporte 4 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Théorème de De Moivre-Laplace

Notations

Dans tout le problème :

- Par convention $0^0 = 1$.
- Si i et j sont des entiers naturels tels que $i \leq j$, on note $\llbracket i, j \rrbracket$ l'ensemble des entiers k tels que $i \leq k \leq j$.
- a et b sont des réels tels que $a < b$.
- Si x est un réel, on définit :

$$\lfloor x \rfloor = \max\{k \in \mathbf{Z}, k \leq x\} \quad \text{et} \quad \lceil x \rceil = \min\{k \in \mathbf{Z}, x \leq k\}$$

- p est un réel de $]0, 1[$ et $q = 1 - p$.
- ζ est la fonction de $] - 1, +\infty[$ dans $\mathbf{R}$ définie par :

$$\zeta(x) = (x + 1) \ln(x + 1).$$

- Φ est la fonction de $\mathbf{R}$ dans $\mathbf{R}$ définie par :

$$\Phi(t) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}}.$$

- $(\Omega, \mathcal{A}, P)$ est un espace probabilisé.
- $(X_n)_{n \in \mathbf{N}^*}$ est une suite de variables aléatoires définies sur $(\Omega, \mathcal{A}, P)$ telle que, pour tout $n \in \mathbf{N}^*$, X_n suit la loi binomiale de paramètres n et p , ce que l'on note $X_n \hookrightarrow \mathcal{B}(n, p)$.

Résultats préliminaires

- 1 ▷ Rappeler la formule de Stirling. En déduire l'existence d'une suite réelle $(\epsilon_n)_{n \in \mathbf{N}^*}$ convergeant vers 0 telle que :

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

- 2 ▷ Soit $\lambda \in \mathbf{R}_+^*$ et $\mu \in \mathbf{R}$. Démontrer que :

$$\lfloor \lambda x + \mu \rfloor \underset{x \rightarrow +\infty}{\sim} \lambda x \quad \text{et} \quad \lceil \lambda x + \mu \rceil \underset{x \rightarrow +\infty}{\sim} \lambda x.$$

3 ▷ Prouver que l'intégrale $\int_{-\infty}^{+\infty} \Phi(t) dt$ converge.

4 ▷ Démontrer que :

$$\zeta(x) \underset{x \rightarrow 0}{=} x + \frac{x^2}{2} + o(x^2).$$

Étude asymptotique d'une suite

Dans cette partie, si $n \in \mathbf{N}^*$, on note x_n le nombre entier $\lceil np - q \rceil$ et p_n le réel $P(X_n = x_n)$.

5 ▷ Justifier que p_n est le plus grand élément de $\{P(X_n = k), k \in \llbracket 0, n \rrbracket\}$.

6 ▷ Vérifier que $\lim_{n \rightarrow +\infty} x_n = +\infty$ et $\lim_{n \rightarrow +\infty} (n - x_n) = +\infty$.

Établir alors :

$$\sqrt{n p q} p_n \underset{n \rightarrow +\infty}{\sim} \frac{n^n p^{x_n} q^{n-x_n}}{\sqrt{2\pi} x_n^{x_n} (n - x_n)^{n-x_n}}.$$

7 ▷ Montrer que, pour tout entier $n > \max\left\{\frac{p}{q}, \frac{q}{p}\right\}$:

$$\frac{n^n p^{x_n} q^{n-x_n}}{x_n^{x_n} (n - x_n)^{n-x_n}} = e^{-np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right)}.$$

8 ▷ Montrer que la suite $(\sqrt{n p q} p_n)_{n \in \mathbf{N}^*}$ converge.

Convergence en loi

Dans toute la suite, pour tout $n \in \mathbf{N}^*$, on note $Y_n = \frac{1}{\sqrt{n p q}}(X_n - np)$ et on définit les réels $\tau_{n,k}$ par la relation :

$$\forall k \in \mathbf{Z}, \tau_{n,k} = \frac{k - np}{\sqrt{n p q}}.$$

9 ▷ Soit $n \in \mathbf{N}^*$. Déterminer la loi de Y_n et vérifier que Y_n est une variable aléatoire centrée réduite.

10 ▷ Justifier l'existence d'un élément $N \in \mathbf{N}^*$ tel que :

$$\text{pour tout entier } n \geq N, \quad [a, b] \subset [\tau_{n,0}, \tau_{n,n}] \text{ et } \frac{1}{\sqrt{n p q}} \leq b - a.$$

On définit les suites $(k_n)_{n \in \mathbf{N}^*}$, $(e_n)_{n \in \mathbf{N}^*}$ et $(f_n)_{n \in \mathbf{N}^*}$, de fonctions de $\mathbf{R}$ dans $\mathbf{R}$ de la façon suivante : pour tout $n \in \mathbf{N}^*$, pour tout $t \in \mathbf{R}$,

$$k_n(t) = \lfloor \sqrt{npq} t + np \rfloor, \quad e_n(t) = \tau_{n, k_n(t)}, \quad f_n(t) = \sqrt{npq} P(Y_n = e_n(t)).$$

11 ▷ Démontrer que pour tout $n \in \mathbf{N}^*$, e_n est une fonction en escalier croissante vérifiant :

$$\forall t \in \mathbf{R}, \quad e_n(t) \leq t < e_n(t) + \frac{1}{\sqrt{npq}}.$$

Démontrer que $(e_n)_{n \in \mathbf{N}^*}$ converge simplement vers une fonction e que l'on précisera.

12 ▷ Montrer que :

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

puis vérifier que

$$P(e_n(a) \leq Y_n \leq e_n(b)) = \int_{\tau_{n, k_n(a)}}^{\tau_{n, k_n(b)}+1} f_n(t) dt.$$

13 ▷ Prouver que, pour tout $n \in \mathbf{N}^*$, pour tout $k \in \llbracket 1, n-1 \rrbracket$:

$$f_n(\tau_{n,k}) = \frac{1}{\sqrt{2\pi}} \sqrt{\frac{pq n^2}{k(n-k)}} \frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} \frac{1 + \epsilon_n}{(1 + \epsilon_k)(1 + \epsilon_{n-k})},$$

où $(\epsilon_n)_{n \in \mathbf{N}^*}$ est la suite définie à la question 1.

14 ▷ Justifier que, pour tout $t \in [a, b]$:

$$\sqrt{\frac{pq n^2}{k_n(t)(n - k_n(t))}} \xrightarrow{n \rightarrow +\infty} 1 \quad \text{et} \quad \frac{1 + \epsilon_n}{(1 + \epsilon_{k_n(t)})(1 + \epsilon_{n - k_n(t)})} \xrightarrow{n \rightarrow +\infty} 1.$$

15 ▷ Montrer que pour tout $n \in \mathbf{N}^*$ et pour tout $k \in \llbracket 0, n \rrbracket$ tels que $\max \left\{ \sqrt{\frac{q}{np}}, \sqrt{\frac{p}{nq}} \right\} \times |\tau_{n,k}| < 1$:

$$\frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} = e^{-np\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) - nq\zeta\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right)}.$$

16 ▷ Démontrer que :

$$\frac{p^{k_n(t)} q^{n - k_n(t)}}{\left(\frac{k_n(t)}{n}\right)^{k_n(t)} \left(\frac{n - k_n(t)}{n}\right)^{n - k_n(t)}} \xrightarrow{n \rightarrow +\infty} e^{-\frac{t^2}{2}}.$$

17 ▷ En conclure que :

$$\forall t \in [a, b], f_n(t) \xrightarrow{n \rightarrow +\infty} \Phi(t),$$

puis que :

$$\int_a^b f_n(t) dt \xrightarrow{n \rightarrow +\infty} \int_a^b \Phi(t) dt.$$

18 ▷ Dédurre de tout ce qui précède que :

$$P(e_n(a) \leq Y_n \leq e_n(b)) \xrightarrow{n \rightarrow +\infty} \int_a^b \Phi(t) dt,$$

puis que :

$$P(a \leq Y_n \leq b) \xrightarrow{n \rightarrow +\infty} \int_a^b \Phi(t) dt.$$

Applications

19 ▷ Montrer que :

$$\forall T \in \mathbf{R}_+^*, \int_{-T}^T \Phi(t) dt \geq 1 - \frac{1}{T^2},$$

puis en déduire la valeur de $\int_{-\infty}^{+\infty} \Phi(t) dt$.

20 ▷ Les suites $(P(Y_n \leq b))_{n \in \mathbf{N}^*}$ et $(P(Y_n \geq a))_{n \in \mathbf{N}^*}$ sont-elles convergentes ? En préciser les limites éventuelles.

Généralisation

Soit φ une fonction de $\mathbf{R}$ dans $\mathbf{R}$, de classe $\mathcal{C}^1$ et telle que φ' ne s'annule pas sur $\mathbf{R}$. Pour tout $n \in \mathbf{N}^*$, on note $Z_n = \varphi \circ Y_n$.

21 ▷ Montrer que, si $\varphi(\mathbf{R}) = \mathbf{R}$, il existe une unique fonction Ψ continue sur $\mathbf{R}$ telle que :

$$\text{pour tout } (\alpha, \beta) \in \overline{\mathbf{R}}^2, \text{ si } \alpha \leq \beta, \text{ alors } P(\alpha \leq Z_n \leq \beta) \xrightarrow{n \rightarrow +\infty} \int_{\alpha}^{\beta} \Psi(t) dt,$$

où $\overline{\mathbf{R}}$ désigne l'ensemble constitué des réels, de $-\infty$ et de $+\infty$.

Que dire si l'on ne suppose plus $\varphi(\mathbf{R}) = \mathbf{R}$?

FIN DU PROBLÈME

SOLUTION

Résultats préliminaires

1. La formule de Stirling dit que $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$. Posons alors

$$\varepsilon_n = \frac{n!}{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n} - 1$$

C'est une suite bien définie et (par définition d'un équivalent, si $u_n \sim v_n$ et v_n ne s'annule pas alors $\frac{u_n}{v_n} \rightarrow 1$) on a

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + \varepsilon_n) \text{ et } \varepsilon_n \rightarrow 0$$

Remarque : pour certains, la formule demandée pourrait être la définition de l'équivalent. Si c'est le cas, il faut s'efforcer de trouver malgré tout des arguments. La caractérisation comme quotient de limite nulle (qui n'est valable que si le quotient existe asymptotiquement) est sans doute la plus facile à manipuler.

2. Par définition, on a

$$\forall y \in \mathbb{R}, y - 1 < \lfloor y \rfloor \leq y \text{ et } y \leq \lceil y \rceil < y + 1$$

En particulier, pour $y = \lambda x + \mu$ et en divisant par $\lambda x > 0$,

$$\forall x > 0, 1 + \frac{\mu - 1}{\lambda x} \leq \frac{\lfloor \lambda x + \mu \rfloor}{\lambda x} \leq 1 \text{ et } 1 \leq \frac{\lceil \lambda x + \mu \rceil}{\lambda x} < 1 + \frac{\mu + 1}{\lambda x}$$

Dans chaque cas, le théorème d'encadrement montre que le quotient tend vers 1 quand $x \rightarrow +\infty$. Ainsi

$$\lfloor \lambda x + \mu \rfloor \underset{x \rightarrow +\infty}{\sim} \lambda x \text{ et } \lceil \lambda x + \mu \rceil \underset{x \rightarrow +\infty}{\sim} \lambda x$$

Remarque : attention, il n'existe pas de théorème donnant directement un équivalent à partir d'un encadrement. Il est vrai que si $f(x) \leq g(x) \leq h(x)$ et si f et h sont équivalentes à la même quantité $u(x)$, alors il en est de même pour $g(x)$. Dans une question où ceci n'est qu'un outil pour conclure, on pourrait sans doute l'admettre. Ici, c'est le cœur de la question et on doit détailler l'utilisation du théorème d'encadrement pour les limites (qui est, lui, au programme).

3. Φ est continue sur $\mathbb{R}$ et les seuls problèmes d'existence/intégrabilité sont ceux au voisinage des infinis.

Par croissances comparées, $\Phi(t) = O(1/t^2)$ aux voisinages des infinis. Comme $t \mapsto 1/t^2$ est positive et intégrable aux voisinages des infinis, Φ l'est aussi :

$$\Phi \text{ est intégrable sur } \mathbb{R}$$

ce qui entraîne bien sûr l'existence de l'intégrale. *Remarque : on doit bien insister sur les problèmes de signe. Les théorèmes de comparaison ne sont valables que dans le cas positif (et avec les "grand O", ce qui importe est la positivité de la quantité dans celui-ci).*

4. On dispose du développement usuel de $\ln(1+x)$ et ainsi, au voisinage de 0,

$$\zeta(x) = (1+x)\left(x - \frac{x^2}{2} + o(x^2)\right) = x + \frac{x^2}{2} + o(x^2)$$

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

Remarque : aucune difficulté ici. Il faut faire attention à l'ordre obtenu dans le calcul. Quand on développe l'expression, on trouve $o(x^2)$ et $o(x^3)$ et il faut garder le moins précis, ici $o(x^2)$ puisque l'on est au voisinage de 0.

Etude asymptotique d'une suite

5. Posons $\alpha_k = \mathbb{P}(X_n = k) = \binom{n}{k} p^k q^{n-k}$ pour $k \in \llbracket 0, n \rrbracket$. On a alors

$$\forall k \in \llbracket 0, n-1 \rrbracket, \frac{\alpha_{k+1}}{\alpha_k} = \frac{(n-k)p}{k+1} \frac{p}{q}$$

Cette quantité est plus grande que 1 quand $k \geq np - q$ et plus petite que 1 sinon. En posant $k_0 = \lceil np - q \rceil$, on a donc

$$\alpha_0 \leq \dots \leq \alpha_{k_0} \geq \alpha_{k_0+1} \geq \dots \geq \alpha_n$$

La valeur maximale est atteinte quand $k = k_0$.

$$\boxed{p_n = \lceil np - q \rceil = \max\{\mathbb{P}(X_n = k), k \in \llbracket 0, n \rrbracket\}}$$

Remarque : il importe de bien écrire les inégalités pour comprendre où est atteint le maximum.

6. On a $x_n \sim np$ quand $n \rightarrow +\infty$ (question 2) et donc $x_n \rightarrow +\infty$.
On peut aussi écrire $x_n = np + o(n)$ et donc $x_n - n = n(p-1) + o(n) \sim -nq$ ($n \rightarrow +\infty$) puisque $q \neq 0$ et donc $x_n - n \rightarrow -\infty$. Ainsi

$$\boxed{\lim_{n \rightarrow +\infty} x_n = \lim_{n \rightarrow +\infty} (n - x_n) = +\infty}$$

On peut alors utiliser la formule de Stirling pour obtenir un équivalent de $(n - x_n)!$ et de $x_n!$ et de

$$\binom{n}{x_n} \sim \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{\sqrt{2\pi(n-x_n)} \left(\frac{n-x_n}{e}\right)^{n-x_n} \sqrt{2\pi x_n} \left(\frac{x_n}{e}\right)^{x_n}}$$

Les termes en puissance de e se simplifient. De plus $x_n \sim np$ et $n - x_n \sim nq$ (voir ci-dessus) et on peut obtenir un équivalent des termes "en racine carrée" :

$$\binom{n}{x_n} \sim \frac{\sqrt{n}}{\sqrt{np}\sqrt{nq}} \frac{n^n}{\sqrt{2\pi} x_n^{x_n} (n-x_n)^{n-x_n}}$$

Comme $\mathbb{P}(X = x_n) = \binom{n}{x_n} p^{x_n} q^{n-x_n}$, on en conclut que

$$\sqrt{npqp_n} \underset{n \rightarrow +\infty}{\sim} \frac{n^n p^{x_n} q^{n-x_n}}{\sqrt{2\pi} x_n^{x_n} (n-x_n)^{n-x_n}}$$

L'énoncé donne le résultat et il faut donc donner un minimum de détail. L'important était de bien utiliser l'équivalent, ici avec les "petits o".

7. Supposons que $n > \max\{p/q, q/p\}$.

$n > q/p$ donne $np - q > 0$ et donc $x_n > 0$.

$n > p/q$ donne $nq > p$ et donc $np = n - nq < n - p$ et donc $np - q < n - 1$ puis $x_n \leq n - 1 < n$.

On a donc $x_n \in]0, n[$ et ainsi $\frac{x_n - np}{np} > -1$ et $\frac{np - x_n}{nq} > -1$. On peut ainsi prendre l'image par ζ de ces réels.

$$\zeta\left(\frac{x_n - np}{np}\right) = \frac{x_n}{np} \ln\left(\frac{x_n}{np}\right) \quad \text{et} \quad \zeta\left(\frac{np - x_n}{nq}\right) = \frac{n - x_n}{nq} \ln\left(\frac{n - x_n}{nq}\right)$$

On en déduit que

$$\begin{aligned} -np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right) &= x_n \ln\left(\frac{np}{x_n}\right) + (n - x_n) \ln\left(\frac{nq}{n - x_n}\right) \\ &= \ln\left(\frac{(np)^{x_n} (nq)^{n-x_n}}{x_n^{x_n} (n - x_n)^{n-x_n}}\right) \\ &= \ln\left(\frac{n^n p^{x_n} q^{n-x_n}}{x_n^{x_n} (n - x_n)^{n-x_n}}\right) \end{aligned}$$

Il reste à passer à l'exponentielle pour conclure que

$$\frac{n^n p^{x_n} q^{n-x_n}}{x_n^{x_n} (n - x_n)^{n-x_n}} = e^{-np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right)}$$

Remarque : pour ce type de question (idem en question 8), penser à faire un brouillon pour éviter une copie trop raturée. Par ailleurs, il est de bon ton de vérifier que les quantités proposées existent, c'est à dire que les arguments auxquels on applique ζ sont dans son domaine. Même si une question n'est pas explicitement posée (comme l'existence d'une intégrale par exemple), il faut prendre l'initiative de détailler.

8. Les deux questions précédentes donnent ainsi, quand $n \rightarrow +\infty$,

$$\sqrt{npqp_n} \sim \frac{1}{\sqrt{2\pi}} e^{-np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right)}$$

Comme $x_n \sim np$, $x_n - np = o(n)$ et donc $\frac{x_n - np}{np} \rightarrow 0$ et avec la question 4,

$$\zeta\left(\frac{x_n - np}{np}\right) = \frac{x_n - np}{np} + \frac{1}{2} \left(\frac{x_n - np}{np}\right)^2 + o\left(\left(\frac{x_n - np}{np}\right)^2\right)$$

De même $np - x_n = o(np)$ donne $\frac{np - x_n}{nq} = o(1)$ et on peut aussi dire

$$\zeta\left(\frac{np - x_n}{nq}\right) = \frac{np - x_n}{nq} + \frac{1}{2} \left(\frac{np - x_n}{nq}\right)^2 + o\left(\left(\frac{np - x_n}{nq}\right)^2\right)$$

Les deux "petits o" valent $o((x_n - np)^2/n^2)$ (les termes constants peuvent être supprimés du petit o). En combinant, on obtient

$$np\zeta\left(\frac{x_n - np}{np}\right) + nq\zeta\left(\frac{np - x_n}{nq}\right) = \frac{(x_n - np)^2}{2n} \left(\frac{1}{p} + \frac{1}{q}\right) + o\left(\frac{(np - x_n)^2}{n}\right)$$

Mais on a en fait $x_n - np = O(1)$ (car $np - q < x_n \leq np - q + 1$) et ces termes sont de limite nulle et on conclut que

$$\lim_{n \rightarrow +\infty} \sqrt{npq}p_n = \frac{1}{\sqrt{2\pi}}$$

Remarque : $x_n - np = O(1)$ donne aussi $\frac{x_n - np}{np} \rightarrow 0$ et il aurait été plus cohérent de donner cet argument dès le début. Le calcul a ici été mené de manière naïve pour faire comprendre ce qui sert ($x_n - np = o(n)$ ou $x_n = np = O(1)$). Comme dans tout calcul asymptotique, on doit être très attentif à l'ordre obtenu dans les calculs (le petit o).

Convergence en loi

9. X_n est d'espérance np et de variance npq . On en déduit que Y_n est d'espérance nulle et de variance égale à 1 (linéarité de l'espérance et propriété $\mathbb{V}(aX + b) = a^2\mathbb{V}(X)$).

$$Y_n \text{ est centrée réduite}$$

X_n étant à valeurs dans $\llbracket 0, n \rrbracket$ et suivant la loi binomiale de paramètres n et p , et puisque les $\tau_{n,k} = \frac{k - np}{\sqrt{npq}}$ sont distincts, on a

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

qui décrit la loi de Y_n .

Remarque : il est important de noter que $k \mapsto \tau_{n,k}$ est injective, l'argument que les $\tau_{n,k}$ pour $k \in \mathbb{Z}$ à n fixé sont distincts, est important pour l'égalité $(Y_n = \tau_{n,k}) = (X_n = k)$.

10. On a $\tau_{n,0} = -\frac{np}{\sqrt{npq}} \rightarrow -\infty$ et est plus petit que a pour n assez grand, disons $n \geq n_1$.
On a aussi $\tau_{n,n} = \frac{nq}{\sqrt{npq}} \rightarrow +\infty$ qui est plus grand que b pour n assez grand, disons $n \geq n_2$.
Enfin, $\frac{1}{\sqrt{npq}} \rightarrow 0$ est plus petit que $b - a > 0$ pour n assez grand, disons $n \geq n_3$.
En posant $N = \max(n_1, n_2, n_3)$, on a

$$N \in \mathbb{N}^* \text{ et } \forall n \geq N, [a, b] \subset [\tau_{n,0}, \tau_{n,n}] \text{ et } \frac{1}{\sqrt{npq}} \leq b - a$$

11. A n fixé, $\tau_{n,j}$ croît avec j . Or, k_n est une fonction croissante.
On en déduit que e_n est aussi une fonction croissante.
Comme la fonction partie entière, k_n est en escalier, les marches étant de largeur $\frac{1}{\sqrt{npq}}$.
Plus précisément, si $t \in [\tau_{j,n}, \tau_{j+1,n}[$, on a $\frac{j - np}{\sqrt{npq}} \leq t < \frac{j+1 - np}{\sqrt{npq}}$ et donc $k_n(t) = j$ (et la réunion des $[\tau_{j,n}, \tau_{j+1,n}[$ est bien égale à $\mathbb{R}$).
On en déduit que pour tout j , e_n est constante sur $[\tau_{j,n}, \tau_{j+1,n}]$ (égale à $\tau_{n,j}$) et donc

e_n est en escalier

Pour un t donné, il existe un unique $j \in \mathbb{Z}$ tel que $\tau_{n,j} = \frac{j-np}{\sqrt{npq}} \leq t < \frac{j+1-np}{\sqrt{npq}} = \tau_{n,j+1}$ et on a alors

$$\forall t, e_n(t) \leq t < e_n(t) + \frac{1}{\sqrt{npq}}$$

Si on écrit plutôt que

$$t - \frac{1}{\sqrt{npq}} < e_n(t) \leq t$$

On voit par encadrement que $e_n(t) \rightarrow t$.

 (e_n) converge simplement sur $\mathbb{R}$ vers la fonction identité

Remarque : le rapport indique qu'il était attendu que le candidat donne les intervalles sur lesquels e_n est constante. Notez que ceci n'était pas vraiment demandé par l'énoncé mais permettait de bien s'assurer de la compréhension du candidat. On pourrait d'ailleurs (c'est vivement conseillé) faire un dessin rapide du graphe de e_n montrant que l'on se situe sous la première bissectrice. Par ailleurs, l'énoncé parle de convergence simple sans préciser sur quel ensemble, ce n'est pas bien !

12. On a

$$\tau_{n,k_n(a)} = \frac{k_n(a) - np}{\sqrt{npq}}$$

Par définition de la partie entière, $\sqrt{npq}a - 1 < k_n(a) - np \leq \sqrt{npq}a$. En divisant par $\sqrt{npq} > 0$, on en déduit par théorème d'encadrement que

$$\lim_{n \rightarrow +\infty} \tau_{n,k_n(a)} = a$$

De la même façon, on montre que $\tau_{n,k_n(b)+1} \rightarrow b$. Par continuité de l'intégrale d'une fonction continue par morceaux vis à vis de ses bornes, on a donc

$$\lim_{n \rightarrow +\infty} \int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} \Phi(t) dt = \int_a^b \Phi(t) dt$$

Remarque : Φ étant continue, le théorème fondamental permet même de dire que $\int_x^y \Phi(t) dt$ est une fonction de classe C^1 de la variable x et de la variable y et donc a fortiori continue.

On écrit que

$$[\tau_{n,k_n(a)}, \tau_{n,k_n(b)+1}[= \bigcup_{j=k_n(a)}^{k_n(b)} [\tau_{n,j}, \tau_{n,j+1}[$$

et par relation de Chasles (les intervalles sont disjoints)

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \sum_{j=k_n(a)}^{k_n(b)} \int_{\tau_{n,j}}^{\tau_{n,j+1}} f_n(t) dt$$

Pour un entier j donné,

$$e_n(t) = \tau_{n,j} \iff k_n(t) = j \iff t \in [\tau_{n,j}, \tau_{n,j+1}[$$

Ainsi pour $t \in [\tau_{n,j}, \tau_{n,j+1}[$, on a $f_n(t) = \sqrt{npq}\mathbb{P}(Y_n = \tau_{n,j})$ et donc

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \sum_{j=k_n(a)}^{k_n(b)} (\tau_{n,j+1} - \tau_{n,j}) \sqrt{npq} \mathbb{P}(Y_n = \tau_{n,j}) = \sum_{j=k_n(a)}^{k_n(b)} \mathbb{P}(Y_n = \tau_{n,j})$$

Puisque Y_n ne prend que les valeurs $\tau_{n,j}$ qui forment une suite croissante, on a donc montré que

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \mathbb{P}(\tau_{n,k_n(a)} \leq Y_n \leq \tau_{n,k_n(b)})$$

Comme $e_n(a) = \tau_{n,k_n(a)}$ et $e_n(b) = \tau_{n,k_n(b)}$

$$\boxed{\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b))}$$

13. On a vu que $e_n(t) = \tau_{n,j} \iff k_n(t) = j \iff t \in [\tau_{n,j}, \tau_{n,j+1}[$ et en particulier,

$$f_n(\tau_{n,k}) = \sqrt{npq} \mathbb{P}(Y_n = \tau_{n,k}) = \sqrt{npq} \binom{n}{k} p^k q^{n-k} = \sqrt{npq} \frac{n!}{k!(n-k)!} p^k q^{n-k}$$

D'après la question 1,

$$\begin{aligned} \frac{n!}{k!(n-k)!} &= \frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} \frac{n^n}{k^k (n-k)^{n-k}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})} \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} \frac{\sqrt{n}}{\sqrt{k(n-k)}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})} \end{aligned}$$

et finalement

$$\boxed{f_n(\tau_{n,k}) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{pqn^2}}{\sqrt{k(n-k)}} \frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})}}$$

Remarque : il ne faut pas se laisser effrayer par la formule. En revanche, il faut écrire proprement les symboles.

14. On a $k_n(t) \sim np + \sqrt{npq}t \sim np$ quand $n \rightarrow +\infty$. Si on écrit cela $k_n(t) = np + o(n)$, on voit que $n - k_n(t) = nq + o(n) \sim nq$. On peut élever les équivalents à une puissance constante et on obtient $\sqrt{k_n(t)(n - k_n(t))} \sim pqn^2$ et donc

$$\boxed{\lim_{n \rightarrow +\infty} \sqrt{\frac{pqn^2}{k_n(t)(n - k_n(t))}} = 1}$$

Comme $n - k_n(t)$ et $k_n(t)$ tendent vers $+\infty$, par composition des limites, $\varepsilon_{k_n(t)}$ et $\varepsilon_{n-k_n(t)}$ tendent aussi vers 1. Par théorèmes d'opération,

$$\lim_{n \rightarrow +\infty} \frac{1 + \varepsilon_n}{(1 + \varepsilon_{k_n(t)})(1 + \varepsilon_{n-k_n(t)})} = 1$$

Remarque : si on veut sommer des équivalents, ce qui est interdit, il suffit de contourner le problème en traduisant ces équivalents avec des “petits o”. L’idée est d’obtenir un terme où les parties principales ne se sont pas simplifiées entre elles.

15. Pour que $\sqrt{\frac{q}{np}}\tau_{n,k} > -1$ et $-\sqrt{\frac{p}{nq}}\tau_{n,k} > -1$, il faut et suffit que

$$-\sqrt{\frac{np}{q}} < \tau_{n,k} < \sqrt{\frac{nq}{p}}$$

c’est à dire que

$$|\tau_{n,k}| \leq \min\left(\sqrt{\frac{np}{q}}, \sqrt{\frac{nq}{p}}\right)$$

Comme le passage à l’inverse est une opération décroissante sur $\mathbb{R}^{+*}$, on a

$$\min\left(\sqrt{\frac{np}{q}}, \sqrt{\frac{nq}{p}}\right) = \frac{1}{\max\left(\sqrt{\frac{q}{np}}, \sqrt{\frac{p}{nq}}\right)}$$

Quand $\max\left(\sqrt{\frac{q}{np}}, \sqrt{\frac{p}{nq}}\right)|\tau_{n,k}| < 1$, on peut donc écrire (les arguments sont > -1)

$$\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) = \zeta\left(\frac{k-np}{np}\right)$$

$$\zeta\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right) = \zeta\left(\frac{np-k}{nq}\right)$$

Le calcul de la question 7 donne alors

$$e^{-np\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) - nq\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right)} = \frac{n^n p^k q^{n-k}}{k^k (n-k)^{n-k}}$$

ce qui s’écrit aussi

$$e^{-np\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) - nq\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right)} = \frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}}$$

16. On a

$$\sqrt{\frac{q}{np}}\tau_{n,k_n(t)} = \sqrt{\frac{q}{np}}e_n(t) \quad \text{et} \quad -\sqrt{\frac{p}{nq}}\tau_{n,k_n(t)} = -\sqrt{\frac{p}{nq}}e_n(t)$$

qui tendent vers 0 quand $n \rightarrow +\infty$ (puisque $e_n(t) \rightarrow t$).

On peut donc utiliser le développement de la question 4

$$\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) = \sqrt{\frac{q}{np}}e_n(t) + \frac{1}{2}\frac{q}{np}e_n(t)^2 + o\left(\frac{1}{n^2}\right)$$

$$\zeta\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right) = -\sqrt{\frac{p}{nq}}e_n(t) + \frac{1}{2}\frac{p}{nq}e_n(t)^2 + o\left(\frac{1}{n^2}\right)$$

On en déduit que

$$np\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) + nq\zeta\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right) = \frac{p+q}{2}e_n(t)^2 + o\left(\frac{1}{n^2}\right)$$

De plus, $e_n(t)^2 = \tau_{n,k_n(t)}^2 = \frac{(k-np)^2}{npq}$ et donc

$$np\zeta\left(\sqrt{\frac{q}{np}}\tau_{n,k}\right) + nq\zeta\left(-\sqrt{\frac{p}{nq}}\tau_{n,k}\right) \underset{n \rightarrow +\infty}{\sim} \frac{(k_n(t) - np)^2}{2n} \left(\frac{1}{p} + \frac{1}{q}\right)$$

Or, $\sqrt{npqt} - 1 < k_n(t) - np \leq \sqrt{npqt}$ montre que $k_n(t) - np \sim \sqrt{npqt}$ et donc le terme précédent tend vers $t^2/2$. Avec la question 15 et la continuité de l'exponentielle, on en conclut que

$$\lim_{n \rightarrow +\infty} \frac{p^{k_n(t)} q^{n-k_n(t)}}{\left(\frac{k_n(t)}{n}\right)^{k_n(t)} \left(\frac{n-k_n(t)}{n}\right)^{n-k_n(t)}} = e^{-\frac{t^2}{2}}$$

17. Si on met ensemble les questions 13, 14 et 16, on obtient

$$\forall t \in [a, b], \quad \lim_{n \rightarrow +\infty} f_n(\tau_{n,k_n(t)}) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} = \Phi(t)$$

Fixons $t \in [a, b]$. On a $e_n(t) = \tau_{n,k_n(t)} = e_n(\tau_{n,k_n(t)})$ et pour tout t , on a donc $f_n(\tau_{n,k_n(t)}) = f_n(t)$. On a ainsi montré que

$$\boxed{\forall t \in [a, b], \quad \lim_{n \rightarrow +\infty} f_n(t) = \Phi(t)}$$

On veut maintenant utiliser un théorème d'interversion entre une limite et une intégrale. On choisit le théorème de convergence dominée.

- La suite (f_n) est constituée de fonctions continues par morceaux (elles sont en escalier comme e_n)
- On vient de voir que (f_n) converge simplement sur $[a, b]$ vers la fonction Φ .
- Φ est continue sur $[a, b]$ (la continuité par morceaux suffirait).
- Enfin,

$$\mathbb{P}(Y_n = e_n(t)) = \mathbb{P}(X_n = \sqrt{npq}e_n(t) + np) \leq p_n$$

où p_n est défini en question 5. On a ainsi

$$\forall t \in [a, b], \quad \forall n \in \mathbb{N}^*, \quad 0 \leq f_n(t) \leq \sqrt{npq}p_n \leq M$$

où M est un majorant de la suite $(\sqrt{npq}p_n)$ qui est bornée puisque convergente (question 8). Le majorant est intégrable sur $[a, b]$ (une constante sur un segment).

On peut donc utiliser le théorème de convergence dominée :

$$\boxed{\lim_{n \rightarrow +\infty} \int_a^b f_n(t) dt = \int_a^b \Phi(t) dt}$$

Remarque : une question de synthèse ! Il faut commencer par repérer qu'il y a un problème d'interversion. Même si vous n'arrivez pas à obtenir la fonction dominatrice, vous pouvez expliquer le reste et ce qu'il reste à faire. Cela sera bonifié.

18. Avec la question 12,

$$\begin{aligned}\mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) &= \int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt \\ &= \int_a^b f_n(t) dt - \int_a^{\tau_{n,k_n(a)}} f_n(t) dt - \int_{\tau_{n,k_n(b)+1}}^b f_n(t) dt\end{aligned}$$

Comme $\tau_{n,k_n(a)} \rightarrow a$ quand $n \rightarrow +\infty$ et $\tau_{n,k_n(b)+1} \rightarrow b$ quand $n \rightarrow +\infty$, en majorant $|f_n|$ par une constante (voir question précédente), on montre que les deuxième et troisième intégrales ci-dessus sont de limite nulle. On vient d'étudier la première et on conclut que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) = \int_a^b \Phi(t) dt$$

On sait que pour tout t , $e_n(t) \leq t < e_n(t) + \frac{1}{\sqrt{npq}}$.

Pour n assez grand ($n \geq N$, question 10), $\frac{1}{\sqrt{npq}} \leq b - a$ et donc

$$e_n(a) \leq a \leq b - \frac{1}{\sqrt{npq}} < e_n(b) \leq b$$

On peut ainsi écrire que

$$\mathbb{P}(a \leq Y_n \leq b) = \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) - \mathbb{P}(e_n(a) \leq Y_n < a) + \mathbb{P}(e_n(b) < Y_n \leq b)$$

Il s'agit de montrer que les deux derniers termes sont de limite nulle. Or (comme $e_n(a) \leq a < e_n(a) + \frac{1}{\sqrt{npq}}$)

$$\begin{aligned}\mathbb{P}(e_n(a) \leq Y_n < a) &\leq \mathbb{P}(e_n(a) \leq Y_n < e_n(a) + \frac{1}{\sqrt{npq}}) \\ &\leq \mathbb{P}(k_n(a) \leq X_n < k_n(a) + 1) \\ &\leq \mathbb{P}(X_n = k_n(a)) \quad (X_n \text{ étant à valeurs entières}) \\ &\leq p_n\end{aligned}$$

La question 8 montre que $p_n \sim \frac{1}{\sqrt{2\pi npq}} \rightarrow 0$ et le terme ci-dessus est de limite nulle. De même, $\mathbb{P}(e_n(b) < Y_n \leq b)$ est de limite nulle.

On peut finalement en conclure que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(a \leq Y_n \leq b) = \int_a^b \Phi(t) dt$$

Applications

19. La variable Y_n étant centrée réduite, l'inégalité de Tchebychev (utilisable avec une variable possédant un moment d'ordre 2) permet d'affirmer que $\mathbb{P}(|Y_n| \geq T) \leq \frac{1}{T^2}$. Ainsi

$$\mathbb{P}(|Y_n| \leq T) \geq \mathbb{P}(|Y_n| < T) = 1 - \mathbb{P}(|Y_n| \geq T) \geq 1 - \frac{1}{T^2}$$

En faisant tendre n vers $+\infty$, on en déduit que

$$\int_{-T}^T \Phi(t) dt \geq 1 - \frac{1}{T^2}$$

De même, $\mathbb{P}(|Y_n| \leq T) \leq 1$ donne $\int_{-T}^T \Phi(t) dt \leq 1$. Par théorème d'encadrement (on fait tendre T vers $+\infty$)

$$\int_{-\infty}^{+\infty} \Phi(t) dt = 1$$

20. Au vu de la question 18, il est raisonnable d'espérer que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(Y_n \leq b) = \int_{-\infty}^b \Phi(t) dt$$

On le prouve en revenant à la définition des limites. On se donne $b \in \mathbb{R}$ et $\varepsilon > 0$. Comme Φ est intégrable sur $] -\infty, b]$,

$$\exists a_0 < b, \forall a \leq a_0, \int_{-\infty}^a |\Phi(t)| dt \leq \varepsilon$$

On écrit alors que, pour tout $n \geq n_0$,

$$\begin{aligned} \left| \mathbb{P}(Y_n \leq b) - \int_{-\infty}^b \Phi(t) dt \right| &= \left| \mathbb{P}(a \leq Y_n \leq b) + \mathbb{P}(Y_n < a) - \int_a^b \Phi - \int_{-\infty}^a \Phi \right| \\ &\leq \left| \mathbb{P}(a \leq Y_n \leq b) - \int_a^b \Phi \right| + \mathbb{P}(Y_n < a) + \int_{-\infty}^a |\Phi| \end{aligned}$$

Pour des $a < 0$, on a $\mathbb{P}(Y_n < a) \leq \mathbb{P}(Y_n^2 \geq -a)$. On peut utiliser l'inégalité de Markov avec la variable positive Y_n^2 dont l'espérance vaut 1 :

$$\mathbb{P}(Y_n < a) \leq \frac{1}{a^2}$$

Le majorant étant de limite nulle quand $a \rightarrow -\infty$, il est plus petit que ε pour a assez petit, disons $a \leq a_1$.

On choisit alors $a = \min(a_0, a_1)$. Ce a étant fixé, il existe d'après la question 18 un rang n_0 tel que

$$\forall n \geq n_0, \left| \mathbb{P}(a \leq Y_n \leq b) - \int_a^b \Phi \right| \leq \varepsilon$$

On a alors

$$\forall n \geq n_0, \left| \mathbb{P}(Y_n \leq b) - \int_{-\infty}^b \Phi(t) dt \right| \leq 3\varepsilon$$

Ceci justifie le résultat annoncé. On montre de manière tout à fait semblable que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(Y_n \geq a) = \int_a^{+\infty} \Phi(t) dt$$

Généralisation

21. φ' est continue et ne s'annule pas sur l'intervalle $\mathbb{R}$. Elle est donc (par le théorème des valeurs intermédiaires) de signe constant sur $\mathbb{R}$. φ est donc continue et strictement monotone sur $\mathbb{R}$ et réalise une bijection de $\mathbb{R}$ dans son image $\varphi(\mathbb{R}) = \mathbb{R}$.

Supposons φ croissante. On a alors $(\alpha \leq Z_n \leq \beta) = (\varphi^{-1}(\alpha) \leq Y_n \leq \varphi^{-1}(\beta))$ et donc

$$\lim_{n \rightarrow +\infty} \mathbb{P}(\alpha \leq Z_n \leq \beta) = \int_{\varphi^{-1}(\alpha)}^{\varphi^{-1}(\beta)} \Phi(t) dt$$

où on convient que $\varphi^{-1}(+\infty) = +\infty$ et de même pour $-\infty$.

Le changement de variable $u = \varphi(t)$ donne alors

$$\lim_{n \rightarrow +\infty} \mathbb{P}(\alpha \leq Z_n \leq \beta) = \int_{\alpha}^{\beta} \Psi(u) du$$

où on a posé $\Psi = \frac{\Phi \circ \varphi^{-1}}{\varphi' \circ \varphi^{-1}}$. Cette limite est justifiée quand $\alpha < \beta$ par

- la question 18 quand α et β sont finis
- la question 20 quand l'un des deux α ou β est infini.

Elle reste vraie si $\alpha = \beta$ car

$$\mathbb{P}(Z_n = \alpha) = \mathbb{P}(Y_n = \varphi^{-1}(\alpha)) = \mathbb{P}(Y_n \leq \varphi^{-1}(\alpha) \cap Y_n \geq \varphi^{-1}(\alpha))$$

et comme $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$,

$$\mathbb{P}(Z_n = \alpha) = \mathbb{P}(Y_n \leq \varphi^{-1}(\alpha)) + \mathbb{P}(Y_n \geq \varphi^{-1}(\alpha)) - 1 \rightarrow 0$$

Dans le cas de décroissance, on a $(\alpha \leq Z_n \leq \beta) = (\varphi^{-1}(\beta) \leq Y_n \leq \varphi^{-1}(\alpha))$ et le calcul se poursuit de même et donne le même résultat.

Si on ne suppose plus $\varphi(\mathbb{R}) = \mathbb{R}$, le résultat ne reste valable que pour α et β adhérents à $\varphi(\mathbb{R})$.

BeginExpansion

SOLUTION

Résultats préliminaires

1. La formule de Stirling dit que $n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n$. Posons alors

$$\varepsilon_n = \frac{n!}{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n} - 1$$

C'est une suite bien définie et (par définition d'un équivalent, si $u_n \sim v_n$ et v_n ne s'annule pas alors $\frac{u_n}{v_n} \rightarrow 1$) on a

$$n! = \sqrt{2\pi n} \left(\frac{n}{e}\right)^n (1 + \varepsilon_n) \text{ et } \varepsilon_n \rightarrow 0$$

Remarque : pour certains, la formule demandée pourrait être la définition de l'équivalent. Si c'est le cas, il faut s'efforcer de trouver malgré tout des arguments. La caractérisation comme quotient de limite nulle (qui n'est valable que si le quotient existe asymptotiquement) est sans doute la plus facile à manipuler.

2. Par définition, on a

$$\forall y \in \mathbb{R}, y - 1 < \lfloor y \rfloor \leq y \text{ et } y \leq \lceil y \rceil < y + 1$$

En particulier, pour $y = \lambda x + \mu$ et en divisant par $\lambda x > 0$,

$$\forall x > 0, 1 + \frac{\mu - 1}{\lambda x} \leq \frac{\lfloor \lambda x + \mu \rfloor}{\lambda x} \leq 1 \text{ et } 1 \leq \frac{\lceil \lambda x + \mu \rceil}{\lambda x} < 1 + \frac{\mu + 1}{\lambda x}$$

Dans chaque cas, le théorème d'encadrement montre que le quotient tend vers 1 quand $x \rightarrow +\infty$. Ainsi

$$\boxed{\lfloor \lambda x + \mu \rfloor \underset{x \rightarrow +\infty}{\sim} \lambda x \text{ et } \lceil \lambda x + \mu \rceil \underset{x \rightarrow +\infty}{\sim} \lambda x}$$

Remarque : attention, il n'existe pas de théorème donnant directement un équivalent à partir d'un encadrement. Il est vrai que si $f(x) \leq g(x) \leq h(x)$ et si f et h sont équivalentes à la même quantité $u(x)$, alors il en est de même pour $g(x)$. Dans une question où ceci n'est qu'un outil pour conclure, on pourrait sans doute l'admettre. Ici, c'est le cœur de la question et on doit détailler l'utilisation du théorème d'encadrement pour les limites (qui est, lui, au programme).

3. Φ est continue sur $\mathbb{R}$ et les seuls problèmes d'existence/intégrabilité sont ceux au voisinage des infinis.

Par croissances comparées, $\Phi(t) = O(1/t^2)$ aux voisinages des infinis. Comme $t \mapsto 1/t^2$ est positive et intégrable aux voisinages des infinis, Φ l'est aussi :

$$\boxed{\Phi \text{ est intégrable sur } \mathbb{R}}$$

ce qui entraîne bien sûr l'existence de l'intégrale. *Remarque : on doit bien insister sur les problématiques de signe. Les théorèmes de comparaison ne sont valables que dans le cas positif (et avec les "grand O", ce qui importe est la positivité de la quantité dans celui-ci).*

4. On dispose du développement usuel de $\ln(1+x)$ et ainsi, au voisinage de 0,

$$\zeta(x) = (1+x)\left(x - \frac{x^2}{2} + o(x^2)\right) = x + \frac{x^2}{2} + o(x^2)$$

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

Remarque : aucune difficulté ici. Il faut faire attention à l'ordre obtenu dans le calcul. Quand on développe l'expression, on trouve $o(x^2)$ et $o(x^3)$ et il faut garder le moins précis, ici $o(x^2)$ puisque l'on est au voisinage de 0.

Etude asymptotique d'une suite

5. Posons $\alpha_k = \mathbb{P}(X_n = k) = \binom{n}{k} p^k q^{n-k}$ pour $k \in \llbracket 0, n \rrbracket$. On a alors

$$\forall k \in \llbracket 0, n-1 \rrbracket, \frac{\alpha_{k+1}}{\alpha_k} = \frac{(n-k)p}{k+1} \frac{1}{q}$$

Cette quantité est plus grande que 1 quand $k \geq np - q$ et plus petite que 1 sinon. En posant $k_0 = \lceil np - q \rceil$, on a donc

$$\alpha_0 \leq \dots \leq \alpha_{k_0} \geq \alpha_{k_0+1} \geq \dots \geq \alpha_n$$

La valeur maximale est atteinte quand $k = k_0$.

$$p_n = \lceil np - q \rceil = \max\{\mathbb{P}(X_n = k), k \in \llbracket 0, n \rrbracket\}$$

Remarque : il importe de bien écrire les inégalités pour comprendre où est atteint le maximum.

6. On a $x_n \sim np$ quand $n \rightarrow +\infty$ (question 2) et donc $x_n \rightarrow +\infty$.

On peut aussi écrire $x_n = np + o(n)$ et donc $x_n - n = n(p-1) + o(n) \sim -nq$ ($n \rightarrow +\infty$) puisque $q \neq 0$ et donc $x_n - n \rightarrow -\infty$. Ainsi

$$\lim_{n \rightarrow +\infty} x_n = \lim_{n \rightarrow +\infty} (n - x_n) = +\infty$$

On peut alors utiliser la formule de Stirling pour obtenir un équivalent de $(n - x_n)!$ et de $x_n!$ et de

$$\binom{n}{x_n} \sim \frac{\sqrt{2\pi n} \left(\frac{n}{e}\right)^n}{\sqrt{2\pi(n-x_n)} \left(\frac{n-x_n}{e}\right)^{n-x_n} \sqrt{2\pi x_n} \left(\frac{x_n}{e}\right)^{x_n}}$$

Les termes en puissance de e se simplifient. De plus $x_n \sim np$ et $n - x_n \sim nq$ (voir ci-dessus) et on peut obtenir un équivalent des termes "en racine carré" :

$$\binom{n}{x_n} \sim \frac{\sqrt{n}}{\sqrt{np}\sqrt{nq}} \frac{n^n}{\sqrt{2\pi x_n^{x_n}} (n - x_n)^{n-x_n}}$$

Comme $\mathbb{P}(X = x_n) = \binom{n}{x_n} p^{x_n} q^{n-x_n}$, on en conclut que

$$\sqrt{npq} p_n \underset{n \rightarrow +\infty}{\sim} \frac{n^n p^{x_n} q^{n-x_n}}{\sqrt{2\pi x_n^{x_n}} (n - x_n)^{n-x_n}}$$

L'énoncé donne le résultat et il faut donc donner un minimum de détail. L'important était de bien utiliser l'équivalent, ici avec les "petits o ".

7. Supposons que $n > \max\{p/q, q/p\}$.

$n > q/p$ donne $np - q > 0$ et donc $x_n > 0$.

$n > p/q$ donne $nq > p$ et donc $np = n - nq < n - p$ et donc $np - q < n - 1$ puis $x_n \leq n - 1 < n$.

On a donc $x_n \in]0, n[$ et ainsi $\frac{x_n - np}{np} > -1$ et $\frac{np - x_n}{nq} > -1$. On peut ainsi prendre l'image par ζ de ces réels.

$$\zeta\left(\frac{x_n - np}{np}\right) = \frac{x_n}{np} \ln\left(\frac{x_n}{np}\right) \quad \text{et} \quad \zeta\left(\frac{np - x_n}{nq}\right) = \frac{n - x_n}{nq} \ln\left(\frac{n - x_n}{nq}\right)$$

On en déduit que

$$\begin{aligned} -np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right) &= x_n \ln\left(\frac{np}{x_n}\right) + (n - x_n) \ln\left(\frac{nq}{n - x_n}\right) \\ &= \ln\left(\frac{(np)^{x_n} (nq)^{n - x_n}}{x_n^{x_n} (n - x_n)^{n - x_n}}\right) \\ &= \ln\left(\frac{n^n p^{x_n} q^{n - x_n}}{x_n^{x_n} (n - x_n)^{n - x_n}}\right) \end{aligned}$$

Il reste à passer à l'exponentielle pour conclure que

$$\frac{n^n p^{x_n} q^{n - x_n}}{x_n^{x_n} (n - x_n)^{n - x_n}} = e^{-np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right)}$$

Remarque : pour ce type de question (idem en question 8), penser à faire un brouillon pour éviter une copie trop raturée. Par ailleurs, il est de bon ton de vérifier que les quantités proposées existent, c'est à dire que les arguments auxquels on applique ζ sont dans son domaine. Même si une question n'est pas explicitement posée (comme l'existence d'une intégrale par exemple), il faut prendre l'initiative de détailler.

8. Les deux questions précédentes donnent ainsi, quand $n \rightarrow +\infty$,

$$\sqrt{npqp_n} \sim \frac{1}{\sqrt{2\pi}} e^{-np\zeta\left(\frac{x_n - np}{np}\right) - nq\zeta\left(\frac{np - x_n}{nq}\right)}$$

Comme $x_n \sim np$, $x_n - np = o(n)$ et donc $\frac{x_n - np}{np} \rightarrow 0$ et avec la question 4,

$$\zeta\left(\frac{x_n - np}{np}\right) = \frac{x_n - np}{np} + \frac{1}{2} \left(\frac{x_n - np}{np}\right)^2 + o\left(\left(\frac{x_n - np}{np}\right)^2\right)$$

De même $np - x_n = o(np)$ donne $\frac{np - x_n}{nq} = o(1)$ et on peut aussi dire

$$\zeta\left(\frac{np - x_n}{nq}\right) = \frac{np - x_n}{nq} + \frac{1}{2} \left(\frac{np - x_n}{nq}\right)^2 + o\left(\left(\frac{np - x_n}{nq}\right)^2\right)$$

Les deux "petits o" valent $o((x_n - np)^2/n^2)$ (les termes constants peuvent être supprimés du petit o). En combinant, on obtient

$$np\zeta\left(\frac{x_n - np}{np}\right) + nq\zeta\left(\frac{np - x_n}{nq}\right) = \frac{(x_n - np)^2}{2n} \left(\frac{1}{p} + \frac{1}{q}\right) + o\left(\frac{(np - x_n)^2}{n}\right)$$

Mais on a en fait $x_n - np = O(1)$ (car $np - q < x_n \leq np - q + 1$) et ces termes sont de limite nulle et on conclut que

$$\lim_{n \rightarrow +\infty} \sqrt{npqp_n} = \frac{1}{\sqrt{2\pi}}$$

Remarque : $x_n - np = O(1)$ donne aussi $\frac{x_n - np}{np} \rightarrow 0$ et il aurait été plus cohérent de donner cet argument dès le début. Le calcul a ici été mené de manière naïve pour faire comprendre ce qui sert ($x_n - np = o(n)$ ou $x_n = np = O(1)$). Comme dans tout calcul asymptotique, on doit être très attentif à l'ordre obtenu dans les calculs (le petit o).

Convergence en loi

9. X_n est d'espérance np et de variance npq . On en déduit que Y_n est d'espérance nulle et de variance égale à 1 (linéarité de l'espérance et propriété $\mathbb{V}(aX + b) = a^2\mathbb{V}(X)$).

Y_n est centrée réduite

X_n étant à valeurs dans $\llbracket 0, n \rrbracket$ et suivant la loi binomiale de paramètres n et p , et puisque les $\tau_{n,k} = \frac{k-np}{\sqrt{npq}}$ sont distincts, on a

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

qui décrit la loi de Y_n .

Remarque : il est important de noter que $k \mapsto \tau_{n,k}$ est injective, l'argument que les $\tau_{n,k}$ pour $k \in \mathbb{Z}$ à n fixé sont distincts, est important pour l'égalité $(Y_n = \tau_{n,k}) = (X_n = k)$.

10. On a $\tau_{n,0} = -\frac{np}{\sqrt{npq}} \rightarrow -\infty$ et est plus petit que a pour n assez grand, disons $n \geq n_1$.
On a aussi $\tau_{n,n} = \frac{nq}{\sqrt{npq}} \rightarrow +\infty$ qui est plus grand que b pour n assez grand, disons $n \geq n_2$.
Enfin, $\frac{1}{\sqrt{npq}} \rightarrow 0$ est plus petit que $b - a > 0$ pour n assez grand, disons $n \geq n_3$.
En posant $N = \max(n_1, n_2, n_3)$, on a

$$N \in \mathbb{N}^* \text{ et } \forall n \geq N, [a, b] \subset [\tau_{n,0}, \tau_{n,n}] \text{ et } \frac{1}{\sqrt{npq}} \leq b - a$$

11. A n fixé, $\tau_{n,j}$ croît avec j . Or, k_n est une fonction croissante.

On en déduit que e_n est aussi une fonction croissante.

Comme la fonction partie entière, k_n est en escalier, les marches étant de largeur $\frac{1}{\sqrt{npq}}$.

Plus précisément, si $t \in [\tau_{j,n}, \tau_{j+1,n}[$, on a $\frac{j-np}{\sqrt{npq}} \leq t < \frac{j+1-np}{\sqrt{npq}}$ et donc $k_n(t) = j$ (et la réunion des $[\tau_{j,n}, \tau_{j+1,n}[$ est bien égale à $\mathbb{R}$).

On en déduit que pour tout j , e_n est constante sur $[\tau_{j,n}, \tau_{j+1,n}]$ (égale à $\tau_{n,j}$) et donc

e_n est en escalier

Pour un t donné, il existe un unique $j \in \mathbb{Z}$ tel que $\tau_{n,j} = \frac{j-np}{\sqrt{npq}} \leq t < \frac{j+1-np}{\sqrt{npq}} = \tau_{n,j+1}$ et on a alors

$$\forall t, e_n(t) \leq t < e_n(t) + \frac{1}{\sqrt{npq}}$$

Si on écrit plutôt que

$$t - \frac{1}{\sqrt{npq}} < e_n(t) \leq t$$

On voit par encadrement que $e_n(t) \rightarrow t$.

(e_n) converge simplement sur $\mathbb{R}$ vers la fonction identité

Remarque : le rapport indique qu'il était attendu que le candidat donne les intervalles sur lesquels e_n est constante. Notez que ceci n'était pas vraiment demandé par l'énoncé mais permettait de bien s'assurer de la compréhension du candidat. On pourrait d'ailleurs (c'est vivement conseillé) faire un dessin rapide du graphe de e_n montrant que l'on se situe sous la première bissectrice. Par ailleurs, l'énoncé parle de convergence simple sans préciser sur quel ensemble, ce n'est pas bien !

12. On a

$$\tau_{n,k_n(a)} = \frac{k_n(a) - np}{\sqrt{npq}}$$

Par définition de la partie entière, $\sqrt{npq}a - 1 < k_n(a) - np \leq \sqrt{npq}a$. En divisant par $\sqrt{npq} > 0$, on en déduit par théorème d'encadrement que

$$\lim_{n \rightarrow +\infty} \tau_{n,k_n(a)} = a$$

De la même façon, on montre que $\tau_{n,k_n(b)+1} \rightarrow b$. Par continuité de l'intégrale d'une fonction continue par morceaux vis à vis de ses bornes, on a donc

$$\lim_{n \rightarrow +\infty} \int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} \Phi(t) dt = \int_a^b \Phi(t) dt$$

Remarque : Φ étant continue, le théorème fondamental permet même de dire que $\int_x^y \Phi(t) dt$ est une fonction de classe C^1 de la variable x et de la variable y et donc a fortiori continue.

On écrit que

$$[\tau_{n,k_n(a)}, \tau_{n,k_n(b)+1}[= \bigcup_{j=k_n(a)}^{k_n(b)} [\tau_{n,j}, \tau_{n,j+1}[$$

et par relation de Chasles (les intervalles sont disjoints)

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \sum_{j=k_n(a)}^{k_n(b)} \int_{\tau_{n,j}}^{\tau_{n,j+1}} f_n(t) dt$$

Pour un entier j donné,

$$e_n(t) = \tau_{n,j} \iff k_n(t) = j \iff t \in [\tau_{n,j}, \tau_{n,j+1}[$$

Ainsi pour $t \in [\tau_{n,j}, \tau_{n,j+1}[$, on a $f_n(t) = \sqrt{npq} \mathbb{P}(Y_n = \tau_{n,j})$ et donc

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \sum_{j=k_n(a)}^{k_n(b)} (\tau_{n,j+1} - \tau_{n,j}) \sqrt{npq} \mathbb{P}(Y_n = \tau_{n,j}) = \sum_{j=k_n(a)}^{k_n(b)} \mathbb{P}(Y_n = \tau_{n,j})$$

Puisque Y_n ne prend que les valeurs $\tau_{n,j}$ qui forment une suite croissante, on a donc montré que

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \mathbb{P}(\tau_{n,k_n(a)} \leq Y_n \leq \tau_{n,k_n(b)})$$

Comme $e_n(a) = \tau_{n,k_n(a)}$ et $e_n(b) = \tau_{n,k_n(b)}$

$$\int_{\tau_{n,k_n(a)}}^{\tau_{n,k_n(b)+1}} f_n(t) dt = \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b))$$

13. On a vu que $e_n(t) = \tau_{n,j} \iff k_n(t) = j \iff t \in [\tau_{n,j}, \tau_{n,j+1}[$ et en particulier,

$$f_n(\tau_{n,k}) = \sqrt{npq} \mathbb{P}(Y_n = \tau_{n,k}) = \sqrt{npq} \binom{n}{k} p^k q^{n-k} = \sqrt{npq} \frac{n!}{k!(n-k)!} p^k q^{n-k}$$

D'après la question 1,

$$\begin{aligned} \frac{n!}{k!(n-k)!} &= \frac{\sqrt{2\pi n}}{\sqrt{2\pi k} \sqrt{2\pi(n-k)}} \frac{n^n}{k^k (n-k)^{n-k}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})} \\ &= \frac{1}{\sqrt{2\pi}} \frac{1}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} \frac{\sqrt{n}}{\sqrt{k(n-k)}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})} \end{aligned}$$

et finalement

$$f_n(\tau_{n,k}) = \frac{1}{\sqrt{2\pi}} \frac{\sqrt{pqn^2}}{\sqrt{k(n-k)}} \frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}} \frac{1 + \varepsilon_n}{(1 + \varepsilon_k)(1 + \varepsilon_{n-k})}$$

Remarque : il ne faut pas se laisser effrayer par la formule. En revanche, il faut écrire proprement les symboles.

14. On a $k_n(t) \sim np + \sqrt{npq}t \sim np$ quand $n \rightarrow +\infty$. Si on écrit cela $k_n(t) = np + o(n)$, on voit que $n - k_n(t) = nq + o(n) \sim nq$. On peut élever les équivalents à une puissance constante et on obtient $\sqrt{k_n(t)(n - k_n(t))} \sim pqn^2$ et donc

$$\lim_{n \rightarrow +\infty} \sqrt{\frac{pqn^2}{k_n(t)(n - k_n(t))}} = 1$$

Comme $n - k_n(t)$ et $k_n(t)$ tendent vers $+\infty$, par composition des limites, $\varepsilon_{k_n(t)}$ et $\varepsilon_{n-k_n(t)}$ tendent aussi vers 1. Par théorèmes d'opération,

$$\lim_{n \rightarrow +\infty} \frac{1 + \varepsilon_n}{(1 + \varepsilon_{k_n(t)})(1 + \varepsilon_{n-k_n(t)})} = 1$$

Remarque : si on veut sommer des équivalents, ce qui est interdit, il suffit de contourner le problème en traduisant ces équivalents avec des "petits o". L'idée est d'obtenir un terme où les parties principales ne se sont pas simplifiées entre elles.

15. Pour que $\sqrt{\frac{q}{np}}\tau_{n,k} > -1$ et $-\sqrt{\frac{p}{nq}}\tau_{n,k} > -1$, il faut et suffit que

$$-\sqrt{\frac{np}{q}} < \tau_{n,k} < \sqrt{\frac{nq}{p}}$$

c'est à dire que

$$|\tau_{n,k}| \leq \min \left(\sqrt{\frac{np}{q}}, \sqrt{\frac{nq}{p}} \right)$$

Comme le passage à l'inverse est une opération décroissante sur $\mathbb{R}^{+*}$, on a

$$\min \left(\sqrt{\frac{np}{q}}, \sqrt{\frac{nq}{p}} \right) = \frac{1}{\max \left(\sqrt{\frac{q}{np}}, \sqrt{\frac{p}{nq}} \right)}$$

Quand $\max \left(\sqrt{\frac{q}{np}}, \sqrt{\frac{p}{nq}} \right) |\tau_{n,k}| < 1$, on peut donc écrire (les arguments sont > -1)

$$\zeta \left(\sqrt{\frac{q}{np}} \tau_{n,k} \right) = \zeta \left(\frac{k - np}{np} \right)$$

$$\zeta \left(-\sqrt{\frac{p}{nq}} \tau_{n,k} \right) = \zeta \left(\frac{np - k}{nq} \right)$$

Le calcul de la question 7 donne alors

$$e^{-np\zeta(\sqrt{\frac{q}{np}}\tau_{n,k}) - nq(-\sqrt{\frac{p}{nq}}\tau_{n,k})} = \frac{n^n p^k q^{n-k}}{k^k (n-k)^{n-k}}$$

ce qui s'écrit aussi

$$e^{-np\zeta(\sqrt{\frac{q}{np}}\tau_{n,k}) - nq(-\sqrt{\frac{p}{nq}}\tau_{n,k})} = \frac{p^k q^{n-k}}{\left(\frac{k}{n}\right)^k \left(\frac{n-k}{n}\right)^{n-k}}$$

16. On a

$$\sqrt{\frac{q}{np}} \tau_{n,k_n(t)} = \sqrt{\frac{q}{np}} e_n(t) \quad \text{et} \quad -\sqrt{\frac{p}{nq}} \tau_{n,k_n(t)} = -\sqrt{\frac{p}{nq}} e_n(t)$$

qui tendent vers 0 quand $n \rightarrow +\infty$ (puisque $e_n(t) \rightarrow t$).

On peut donc utiliser le développement de la question 4

$$\zeta \left(\sqrt{\frac{q}{np}} \tau_{n,k} \right) = \sqrt{\frac{q}{np}} e_n(t) + \frac{1}{2} \frac{q}{np} e_n(t)^2 + o \left(\frac{1}{n^2} \right)$$

$$\zeta \left(-\sqrt{\frac{p}{nq}} \tau_{n,k} \right) = -\sqrt{\frac{p}{nq}} e_n(t) + \frac{1}{2} \frac{p}{nq} e_n(t)^2 + o \left(\frac{1}{n^2} \right)$$

On en déduit que

$$np\zeta \left(\sqrt{\frac{q}{np}} \tau_{n,k} \right) + nq\zeta \left(-\sqrt{\frac{p}{nq}} \tau_{n,k} \right) = \frac{p+q}{2} e_n(t)^2 + o \left(\frac{1}{n^2} \right)$$

De plus, $e_n(t)^2 = \tau_{n,k_n(t)}^2 = \frac{(k-np)^2}{npq}$ et donc

$$np\zeta \left(\sqrt{\frac{q}{np}} \tau_{n,k} \right) + nq\zeta \left(-\sqrt{\frac{p}{nq}} \tau_{n,k} \right) \underset{n \rightarrow +\infty}{\sim} \frac{(k_n(t) - np)^2}{2n} \left(\frac{1}{p} + \frac{1}{q} \right)$$

Or, $\sqrt{npqt} - 1 < k_n(t) - np \leq \sqrt{npqt}$ montre que $k_n(t) - np \sim \sqrt{npqt}$ et donc le terme précédent tend vers $t^2/2$. Avec la question 15 et la continuité de l'exponentielle, on en conclut que

$$\lim_{n \rightarrow +\infty} \frac{p^{k_n(t)} q^{n-k_n(t)}}{\left(\frac{k_n(t)}{n}\right)^{k_n(t)} \left(\frac{n-k_n(t)}{n}\right)^{n-k_n(t)}} = e^{-\frac{t^2}{2}}$$

17. Si on met ensemble les questions 13, 14 et 16, on obtient

$$\forall t \in [a, b], \lim_{n \rightarrow +\infty} f_n(\tau_{n,k_n}(t)) = \frac{1}{\sqrt{2\pi}} e^{-\frac{t^2}{2}} = \Phi(t)$$

Fixons $t \in [a, b]$. On a $e_n(t) = \tau_{n,k_n}(t) = e_n(\tau_{n,k_n}(t))$ et pour tout t , on a donc $f_n(\tau_{n,k_n}(t)) = f_n(t)$. On a ainsi montré que

$$\boxed{\forall t \in [a, b], \lim_{n \rightarrow +\infty} f_n(t) = \Phi(t)}$$

On veut maintenant utiliser un théorème d'interversion entre une limite et une intégrale. On choisit le théorème de convergence dominée.

- La suite (f_n) est constituée de fonctions continues par morceaux (elles sont en escalier comme e_n)
- On vient de voir que (f_n) converge simplement sur $[a, b]$ vers la fonction Φ .
- Φ est continue sur $[a, b]$ (la continuité par morceaux suffirait).
- Enfin,

$$\mathbb{P}(Y_n = e_n(t)) = \mathbb{P}(X_n = \sqrt{npq}e_n(t) + np) \leq p_n$$

où p_n est défini en question 5. On a ainsi

$$\forall t \in [a, b], \forall n \in \mathbb{N}^*, 0 \leq f_n(t) \leq \sqrt{npqp_n} \leq M$$

où M est un majorant de la suite $(\sqrt{npqp_n})$ qui est bornée puisque convergente (question 8). Le majorant est intégrable sur $[a, b]$ (une constante sur un segment).

On peut donc utiliser le théorème de convergence dominée :

$$\boxed{\lim_{n \rightarrow +\infty} \int_a^b f_n(t) dt = \int_a^b \Phi(t) dt}$$

Remarque : une question de synthèse ! Il faut commencer par repérer qu'il y a un problème d'interversion. Même si vous n'arrivez pas à obtenir la fonction dominatrice, vous pouvez expliquer le reste et ce qu'il reste à faire. Cela sera bonifié.

18. Avec la question 12,

$$\begin{aligned} \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) &= \int_{\tau_{n,k_n}(a)}^{\tau_{n,k_n}(b)+1} f_n(t) dt \\ &= \int_a^b f_n(t) dt - \int_a^{\tau_{n,k_n}(a)} f_n(t) dt - \int_{\tau_{n,k_n}(b)+1}^b f_n(t) dt \end{aligned}$$

Comme $\tau_{n,k_n}(a) \rightarrow a$ quand $n \rightarrow +\infty$ et $\tau_{n,k_n}(b)+1 \rightarrow b$ quand $n \rightarrow +\infty$, en majorant $|f_n|$ par une constante (voir question précédente), on montre que les deuxième et troisième intégrales ci-dessus sont de limite nulle. On vient d'étudier la première et on conclut que

$$\boxed{\lim_{n \rightarrow +\infty} \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) = \int_a^b \Phi(t) dt}$$

On sait que pour tout t , $e_n(t) \leq t < e_n(t) + \frac{1}{\sqrt{npq}}$.

Pour n assez grand ($n \geq N$, question 10), $\frac{1}{\sqrt{npq}} \leq b - a$ et donc

$$e_n(a) \leq a \leq b - \frac{1}{\sqrt{npq}} < e_n(b) \leq b$$

On peut ainsi écrire que

$$\mathbb{P}(a \leq Y_n \leq b) = \mathbb{P}(e_n(a) \leq Y_n \leq e_n(b)) - \mathbb{P}(e_n(a) \leq Y_n < a) + \mathbb{P}(e_n(b) < Y_n \leq b)$$

Il s'agit de montrer que les deux derniers termes sont de limite nulle. Or (comme $e_n(a) \leq a < e_n(a) + \frac{1}{\sqrt{npq}}$)

$$\begin{aligned} \mathbb{P}(e_n(a) \leq Y_n < a) &\leq \mathbb{P}(e_n(a) \leq Y_n < e_n(a) + \frac{1}{\sqrt{npq}}) \\ &\leq \mathbb{P}(k_n(a) \leq X_n < k_n(a) + 1) \\ &\leq \mathbb{P}(X_n = k_n(a)) \quad (X_n \text{ étant à valeurs entières}) \\ &\leq p_n \end{aligned}$$

La question 8 montre que $p_n \sim \frac{1}{\sqrt{2\pi npq}} \rightarrow 0$ et le terme ci-dessus est de limite nulle. De même, $\mathbb{P}(e_n(b) < Y_n \leq b)$ est de limite nulle.

On peut finalement en conclure que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(a \leq Y_n \leq b) = \int_a^b \Phi(t) dt$$

Applications

19. La variable Y_n étant centrée réduite, l'inégalité de Tchebychev (utilisable avec une variable possédant un moment d'ordre 2) permet d'affirmer que $\mathbb{P}(|Y_n| \geq T) \leq \frac{1}{T^2}$. Ainsi

$$\mathbb{P}(|Y_n| \leq T) \geq \mathbb{P}(|Y_n| < T) = 1 - \mathbb{P}(|Y_n| \geq T) \geq 1 - \frac{1}{T^2}$$

En faisant tendre n vers $+\infty$, on en déduit que

$$\int_{-T}^T \Phi(t) dt \geq 1 - \frac{1}{T^2}$$

De même, $\mathbb{P}(|Y_n| \leq T) \leq 1$ donne $\int_{-T}^T \Phi(t) dt \leq 1$. Par théorème d'encadrement (on fait tendre T vers $+\infty$)

$$\int_{-\infty}^{+\infty} \Phi(t) dt = 1$$

20. Au vu de la question 18, il est raisonnable d'espérer que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(Y_n \leq b) = \int_{-\infty}^b \Phi(t) dt$$

On le prouve en revenant à la définition des limites. On se donne $b \in \mathbb{R}$ et $\varepsilon > 0$. Comme Φ est intégrable sur $] -\infty, b]$,

$$\exists a_0 < b, \forall a \leq a_0, \int_{-\infty}^a |\Phi(t)| dt \leq \varepsilon$$

On écrit alors que, pour tout $n \geq n_0$,

$$\begin{aligned} \left| \mathbb{P}(Y_n \leq b) - \int_{-\infty}^b \Phi(t) dt \right| &= \left| \mathbb{P}(a \leq Y_n \leq b) + \mathbb{P}(Y_n < a) - \int_a^b \Phi - \int_{-\infty}^a \Phi \right| \\ &\leq \left| \mathbb{P}(a \leq Y_n \leq b) - \int_a^b \Phi \right| + \mathbb{P}(Y_n < a) + \int_{-\infty}^a |\Phi| \end{aligned}$$

Pour des $a < 0$, on a $\mathbb{P}(Y_n < a) \leq \mathbb{P}(Y_n^2 \geq -a)$. On peut utiliser l'inégalité de Markov avec la variable positive Y_n^2 dont l'espérance vaut 1 :

$$\mathbb{P}(Y_n < a) \leq \frac{1}{a^2}$$

Le majorant étant de limite nulle quand $a \rightarrow -\infty$, il est plus petit que ε pour a assez petit, disons $a \leq a_1$.

On choisit alors $a = \min(a_0, a_1)$. Ce a étant fixé, il existe d'après la question 18 un rang n_0 tel que

$$\forall n \geq n_0, \left| \mathbb{P}(a \leq Y_n \leq b) - \int_a^b \Phi \right| \leq \varepsilon$$

On a alors

$$\forall n \geq n_0, \left| \mathbb{P}(Y_n \leq b) - \int_{-\infty}^b \Phi(t) dt \right| \leq 3\varepsilon$$

Ceci justifie le résultat annoncé. On montre de manière tout à fait semblable que

$$\lim_{n \rightarrow +\infty} \mathbb{P}(Y_n \geq a) = \int_a^{+\infty} \Phi(t) dt$$

Généralisation

21. φ' est continue et ne s'annule pas sur l'intervalle $\mathbb{R}$. Elle est donc (par le théorème des valeurs intermédiaires) de signe constant sur $\mathbb{R}$. φ est donc continue et strictement monotone sur $\mathbb{R}$ et réalise une bijection de $\mathbb{R}$ dans son image $\varphi(\mathbb{R}) = \mathbb{R}$.

Supposons φ croissante. On a alors $(\alpha \leq Z_n \leq \beta) = (\varphi^{-1}(\alpha) \leq Y_n \leq \varphi^{-1}(\beta))$ et donc

$$\lim_{n \rightarrow +\infty} \mathbb{P}(\alpha \leq Z_n \leq \beta) = \int_{\varphi^{-1}(\alpha)}^{\varphi^{-1}(\beta)} \Phi(t) dt$$

où on convient que $\varphi^{-1}(+\infty) = +\infty$ et de même pour $-\infty$.

Le changement de variable $u = \varphi(t)$ donne alors

$$\lim_{n \rightarrow +\infty} \mathbb{P}(\alpha \leq Z_n \leq \beta) = \int_{\alpha}^{\beta} \Psi(u) du$$

où on a posé $\Psi = \frac{\Phi \circ \varphi^{-1}}{\varphi' \circ \varphi^{-1}}$. Cette limite est justifiée quand $\alpha < \beta$ par

- la question 18 quand α et β sont finis
- la question 20 quand l'un des deux α ou β est infini.

Elle reste vraie si $\alpha = \beta$ car

$$\mathbb{P}(Z_n = \alpha) = \mathbb{P}(Y_n = \varphi^{-1}(\alpha)) = \mathbb{P}(Y_n \leq \varphi^{-1}(\alpha) \cap Y_n \geq \varphi^{-1}(\alpha))$$

et comme $\mathbb{P}(A \cup B) = \mathbb{P}(A) + \mathbb{P}(B) - \mathbb{P}(A \cap B)$,

$$\mathbb{P}(Z_n = \alpha) = \mathbb{P}(Y_n \leq \varphi^{-1}(\alpha)) + \mathbb{P}(Y_n \geq \varphi^{-1}(\alpha)) - 1 \rightarrow 0$$

Dans le cas de décroissance, on a $(\alpha \leq Z_n \leq \beta) = (\varphi^{-1}(\beta) \leq Y_n \leq \varphi^{-1}(\alpha))$ et le calcul se poursuit de même et donne le même résultat.

Si on ne suppose plus $\varphi(\mathbb{R}) = \mathbb{R}$, le résultat ne reste valable que pour α et β adhérents à $\varphi(\mathbb{R})$.



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2021

DEUXIÈME ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 4 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES II - MP

L'énoncé de cette épreuve comporte 5 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Fonctions de matrices symétriques, continuité et convexité

Dans ce problème, on propose de définir la notion d'image d'une matrice réelle symétrique par une fonction d'une variable réelle, puis d'étudier quelques propriétés de cette notion (en particulier, relativement à la continuité et à la convexité). Ces notions présentent un intérêt en sciences physiques (statistique ou quantique).

Notations

Dans tout le problème :

- n désigne un entier naturel non nul ;
- si p et q sont des entiers naturels, l'ensemble des entiers k tels que $p \leq k \leq q$ est noté $\llbracket p, q \rrbracket$;
- si i et j sont des entiers naturels, alors $\delta_{i,j} = 1$ si $i = j$ et $\delta_{i,j} = 0$ sinon ;
- B_n désigne l'ensemble des bijections de $\llbracket 1, n \rrbracket$ dans lui-même ;
- I est un intervalle de $\mathbf{R}$ qui n'est ni vide ni réduit à un singleton ;
- $\mathcal{C}^0(I, \mathbf{R})$ désigne l'ensemble des fonctions continues de I dans $\mathbf{R}$;
- une fonction φ de I dans $\mathbf{R}$ est dite polynomiale s'il existe P un polynôme réel tel que, pour tout $x \in I$, $\varphi(x) = P(x)$;
- $\mathcal{M}_n(\mathbf{R})$ (respectivement $D_n(\mathbf{R})$, resp. $S_n(\mathbf{R})$, resp. $O_n(\mathbf{R})$), désigne l'ensemble des matrices carrées (resp. diagonales, resp. symétriques, resp. orthogonales) d'ordre n à coefficients réels, et on confond un élément de $\mathcal{M}_1(\mathbf{R})$ avec son unique coefficient ;
- on note Tr l'application trace définie sur $\mathcal{M}_n(\mathbf{R})$;
- si $M \in \mathcal{M}_n(\mathbf{R})$, on note ${}^tM$ sa transposée, on note $\text{Sp}(M)$ son spectre réel, et si $(i, j) \in \llbracket 1, n \rrbracket^2$, $[M]_{i,j}$ est le coefficient de M situé à la i -ème ligne et j -ème colonne ;
- on munit $\mathcal{M}_n(\mathbf{R})$ de sa norme infinie, notée $\|\cdot\|$ et définie par :

$$\forall M \in \mathcal{M}_n(\mathbf{R}), \quad \|M\| = \max \{|[M]_{i,j}|, 1 \leq i, j \leq n\} ;$$

- $S_n(I)$ désigne l'ensemble des matrices de $S_n(\mathbf{R})$ dont le spectre réel est inclus dans I ;

- si $u = (u_i)_{1 \leq i \leq n} \in \mathbf{R}^n$, on dit que ce n -uplet est croissant si pour tout $(i, j) \in \llbracket 1, n \rrbracket^2$,

$$(i \leq j) \implies (u_i \leq u_j) ;$$

- si $i_0 \in \llbracket 1, n \rrbracket$, on appelle nombre d'occurrences de u_{i_0} dans u le cardinal de l'ensemble $\{i \in \llbracket 1, n \rrbracket ; u_i = u_{i_0}\}$;

- enfin $\text{Diag}((u_i)_{1 \leq i \leq n})$ désigne l'élément D de $D_n(\mathbf{R})$ tel que :

$$\forall i \in \llbracket 1, n \rrbracket, [D]_{i,i} = u_i$$

on pourra noter cet élément en extension $D = \text{Diag}(u_1, \dots, u_n)$.

Matrices de permutations

Le but de cette partie est d'étudier l'action sur les matrices diagonales de la conjugaison par des matrices de permutations. On considère l'application ω de B_n dans $\mathcal{M}_n(\mathbf{R})$ définie par :

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

- 1 ▷ Démontrer que pour tout $(\sigma, \sigma') \in B_n^2$, $\omega(\sigma \circ \sigma') = \omega(\sigma) \omega(\sigma')$.

- 2 ▷ Démontrer que $\omega(B_n) \subset O_n(\mathbf{R})$.

- 3 ▷ Soit $\sigma \in B_n$ et $(d_i)_{1 \leq i \leq n} \in \mathbf{R}^n$. Vérifier que :

$$\text{Diag}((d_i)_{1 \leq i \leq n}) \omega(\sigma) = \omega(\sigma) \text{Diag}((d_{\sigma(i)})_{1 \leq i \leq n}).$$

- 4 ▷ En déduire l'équivalence suivante concernant deux éléments D et D' de $D_n(\mathbf{R})$,

- i) D et D' ont le même ensemble de coefficients diagonaux, chacun ayant le même nombre d'occurrences dans D et D' .
- ii) il existe $M \in \omega(B_n)$ telle que $D' = {}^t M D M$.

Fonctions de matrices symétriques

Cette partie a pour objectif de définir une correspondance entre l'espace des fonctions de I dans $\mathbf{R}$ et l'espace des fonctions de $S_n(I)$ dans $S_n(\mathbf{R})$, puis d'en démontrer quelques propriétés. Dans cette partie, f est une fonction de I dans $\mathbf{R}$.

5 ▷ Soit $S \in S_n(I)$. Justifier l'existence de $\Omega \in O_n(\mathbf{R})$ et de $(s_i)_{1 \leq i \leq n} \in I^n$ tels que :

$$S = {}^t\Omega \text{Diag}((s_i)_{1 \leq i \leq n}) \Omega.$$

6 ▷ Pour tout $(s_i)_{1 \leq i \leq n} \in I^n$, justifier l'existence d'un élément P de $\mathbf{R}[X]$ tel que :

$$\forall i \in \llbracket 1, n \rrbracket, P(s_i) = f(s_i).$$

Soit $S \in S_n(I)$. On suppose que l'on dispose des deux écritures :

$$S = {}^t\Omega \text{Diag}((s_i)_{1 \leq i \leq n}) \Omega \quad \text{et} \quad S = {}^t\Omega' \text{Diag}((s'_i)_{1 \leq i \leq n}) \Omega',$$

avec $\Omega, \Omega' \in O_n(\mathbf{R})$ et $(s_i)_{1 \leq i \leq n}, (s'_i)_{1 \leq i \leq n} \in I^n$.

7 ▷ Montrer que l'on a alors :

$${}^t\Omega' \text{Diag}((f(s'_i))_{1 \leq i \leq n}) \Omega' = {}^t\Omega \text{Diag}((f(s_i))_{1 \leq i \leq n}) \Omega,$$

$$\text{puis que } {}^t\Omega \text{Diag}((f(s_i))_{1 \leq i \leq n}) \Omega \in S_n(\mathbf{R}).$$

Dans la suite du problème, on note u l'application qui, à toute fonction φ de I dans $\mathbf{R}$, associe $u(\varphi)$ la fonction de $S_n(I)$ dans $S_n(\mathbf{R})$ définie par :

$$\forall S \in S_n(I), u(\varphi)(S) = {}^t\Omega \text{Diag}((\varphi(s_i))_{1 \leq i \leq n}) \Omega,$$

où $S = {}^t\Omega \text{Diag}((s_i)_{1 \leq i \leq n}) \Omega$, avec $\Omega \in O_n(\mathbf{R})$ et $(s_i)_{1 \leq i \leq n} \in I^n$.

Cette fonction est bien définie puisque, d'après la question précédente, $u(\varphi)(S)$ ne dépend pas du choix des matrices $\Omega \in O_n(\mathbf{R})$ et $D = \text{Diag}((s_i)_{1 \leq i \leq n})$ avec $(s_i)_{1 \leq i \leq n} \in I^n$, tel que $S = {}^t\Omega D \Omega$.

Enfin, on désigne par v l'application $\text{Tr} \circ u$.

8 ▷ Vérifier que u et v sont linéaires, puis calculer, pour toute fonction φ de I dans $\mathbf{R}$ et pour tout $x \in I$, $u(\varphi)(xI_n)$.

9 ▷ Étudier l'injectivité et la surjectivité de u .

10 ▷ On suppose que f est polynomiale ; montrer qu'il existe $P \in \mathbf{R}[X]$ tel que pour tout $S \in S_n(I)$, $u(f)(S) = P(S)$.

Réciproquement, est-il vrai que, s'il existe $P \in \mathbf{R}[X]$ tel que pour tout $S \in S_n(I)$, $u(f)(S) = P(S)$, alors f est polynomiale ?

- 11** ▷ Démontrer que, si $(\varphi_k)_{k \in \mathbf{N}}$ est une suite de fonctions de I dans $\mathbf{R}$ qui converge simplement sur I vers une fonction φ , alors les suites $(u(\varphi_k))_{k \in \mathbf{N}}$ et $(v(\varphi_k))_{k \in \mathbf{N}}$ convergent simplement sur $S_n(I)$.

Y a-t-il convergence uniforme sur $S_n(I)$ si l'on suppose que $(\varphi_k)_{k \in \mathbf{N}}$ converge uniformément sur I ?

Norme et convexité

L'objectif de cette partie est de munir $S_n(\mathbf{R})$ d'une nouvelle norme qui permettra de compléter l'étude des fonctions de matrices symétriques.

- 12** ▷ On note $\Sigma = \{X \in \mathcal{M}_{n,1}(\mathbf{R}) ; {}^t X X = 1\}$. Démontrer que si $S \in S_n(\mathbf{R})$ on a :

$$\min(\mathrm{Sp}(S)) = \min \{ {}^t X S X ; X \in \Sigma \} \text{ et } \max(\mathrm{Sp}(S)) = \max \{ {}^t X S X ; X \in \Sigma \}.$$

- 13** ▷ Montrer finalement que $S_n(I)$ est une partie convexe de $S_n(\mathbf{R})$ et que l'application ρ , de $S_n(\mathbf{R})$ dans $\mathbf{R}$, qui à toute matrice $M \in S_n(\mathbf{R})$ associe

$$\max \{ |\lambda| ; \lambda \in \mathrm{Sp}(M) \},$$

est une norme sur $S_n(\mathbf{R})$.

Continuité des fonctions de matrices symétriques

Dans cette partie, à l'aide de la norme précédemment introduite, on démontre quelques résultats relatifs à la continuité des fonctions de matrices symétriques. On suppose désormais $S_n(\mathbf{R})$ muni de la norme ρ et on appelle χ l'application de $S_n(\mathbf{R})$ dans $\mathbf{R}[X]$ qui, à tout élément de $S_n(\mathbf{R})$, associe son polynôme caractéristique.

On définit aussi l'application, notée $\mathrm{Sp}_\uparrow$, qui à toute matrice $S \in S_n(\mathbf{R})$, associe son spectre croissant (c'est-à-dire le n -uplet croissant des valeurs propres de S dans lequel le nombre d'occurrences de chaque valeur propre coïncide avec son ordre de multiplicité).

- 14** ▷ Démontrer que χ est continue.

On souhaite maintenant prouver que $\mathrm{Sp}_\uparrow$ est continue. À cet effet, on introduit un élément M de $S_n(\mathbf{R})$ et une suite $(M_k)_{k \in \mathbf{N}}$ à valeurs dans $S_n(\mathbf{R})$ qui converge vers M . Si $k \in \mathbf{N}$, on note $\Lambda_k = \mathrm{Sp}_\uparrow(M_k)$.

- 15** ▷ Démontrer que la suite $(\Lambda_k)_{k \in \mathbf{N}}$ admet une valeur d'adhérence croissante.

16 ▷ Montrer que, si α est une application strictement croissante de $\mathbf{N}$ dans $\mathbf{N}$ telle que la suite $(\Lambda_{\alpha(k)})_{k \in \mathbf{N}}$ converge, alors : $\Lambda_{\alpha(k)} \xrightarrow[k \rightarrow +\infty]{} \text{Sp}_1(M)$.

17 ▷ En déduire que Sp_1 est continue.

18 ▷ Démontrer que $O_n(\mathbf{R})$ est une partie compacte de $\mathcal{M}_n(\mathbf{R})$.

19 ▷ Démontrer que, si $\varphi \in \mathcal{C}^0(I, \mathbf{R})$, alors $u(\varphi)$ et $v(\varphi)$ sont continues.

Convexité des fonctions de matrices symétriques

On démontre maintenant quelques résultats relatifs à la convexité des fonctions de matrices symétriques. Dans cette partie, f est une fonction de I dans $\mathbf{R}$.

20 ▷ On suppose ici que f est convexe sur I et que $S \in S_n(I)$. On note

$$\mathcal{U}_S = \{ {}^t \Omega S \Omega ; \Omega \in O_n(\mathbf{R}) \}.$$

Justifier que pour tout $U \in \mathcal{U}_S$, pour tout $k \in \llbracket 1, n \rrbracket$, $[U]_{k,k} \in I$.

Démontrer alors que :

$$\max \left\{ \sum_{k=1}^n f([U]_{k,k}) ; U \in \mathcal{U}_S \right\} = v(f)(S).$$

21 ▷ En déduire que, si f est convexe sur I , pour tout $(A, B) \in S_n(I)^2$, pour tout $t \in [0, 1]$, on a :

$$v(f)((1-t)A + tB) \leq (1-t)v(f)(A) + tv(f)(B).$$

On dit qu'une fonction ψ de $S_n(I)$ dans $\mathbf{R}$ est convexe sur $S_n(I)$ si elle vérifie la relation :

$$\forall (A, B) \in S_n(I)^2, \forall t \in [0, 1], \quad \psi((1-t)A + tB) \leq (1-t)\psi(A) + t\psi(B).$$

22 ▷ Démontrer finalement que la fonction $v(f)$ est convexe sur $S_n(I)$ si, et seulement si, f est convexe sur I .

FIN DU PROBLÈME

SOLUTION

Matrices de permutation

1. Notons u_σ l'endomorphisme canoniquement associé à $\omega(\sigma)$. On a alors $u(e_j) = e_{\sigma(j)}$ (où les e_i sont les éléments de la base canonique). Ainsi, pour $\sigma, \sigma' \in B_n$,

$$\forall j, u_{\sigma \circ \sigma'}(e_j) = e_{\sigma \circ \sigma'(e_j)} = u_\sigma(u_{\sigma'}(e_j)) = u_\sigma \circ u_{\sigma'}(e_j)$$

En revenant aux matrices,

$$\boxed{\forall \sigma, \sigma' \in B_n, \omega(\sigma \circ \sigma') = \omega(\sigma)\omega(\sigma')}$$

Remarque : dans cette première question, il est tout à fait possible de prouver le résultat à l'aide de la formule du produit de matrices :

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

Dans la somme le terme est nul si $k \neq \sigma'(j)$ et donc

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

ce qui permet de retrouver le résultat. Le premier calcul est bien sûr beaucoup plus élégant et dans la suite, nous essayerons de garder cette élégance permettant d'éviter certains calculs. Si vous ne pensez pas à transiter par des endomorphismes, vous devez être clairs dans les calculs et, en particulier, la manipulation du symbole de Kronecker δ .

2. Avec les notation précédentes, u_σ permute les éléments de la base canonique et envoie donc une base orthonormée sur une base orthonormée. $\omega(\sigma)$ représente donc une isométrie en b.o.n et est donc orthogonale.

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

Remarque : si on choisit de travailler matriciellement, on doit montrer que $\omega(\sigma)^T \omega(\sigma) = I_n$. On forme donc

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

Ce terme vaut $\delta_{\sigma(j),\sigma(i)}$ (terme de la somme nul si $k \neq \sigma(j)$). Le terme est donc nul si $i \neq j$ et vaut 1 sinon, ce qui permet de conclure par le calcul.

3. Multiplier à droite (resp à gauche) $\text{diag}(d_1, \dots, d_n)$ revient à multiplier chaque colonne (resp ligne) par d_j . Ainsi

$$[\text{diag}(d_1, \dots, d_n)\omega(\sigma)]_{i,j} = d_i \delta_{i,\sigma(j)}$$

$$[\omega(\sigma)\text{diag}(d_{\sigma(1)}, \dots, d_{\sigma(n)})]_{i,j} = d_{\sigma(j)} \delta_{i,\sigma(j)} = d_i \delta_{i,\sigma(j)}$$

(pour la dernière égalité : les termes sont nuls si $i \neq \sigma(j)$ et donc égaux ; ils sont aussi égaux si $i = \sigma(j)$)

On a donc montré que

$$\text{diag}(d_1, \dots, d_n) \omega(\sigma) = \omega(\sigma) \text{diag}(d_{\sigma(1)}, \dots, d_{\sigma(n)})$$

Remarque : il faut se souvenir qu'une multiplication à gauche fait agir sur les lignes et qu'une multiplication à droite fait agir sur les colonnes et revoir la traduction des opérations élémentaires sur les lignes/colonnes d'une matrice en termes de produit matriciel.

4. La propriété (i) signifie qu'il existe $\sigma \in B_n$ telle que $\forall i, d'_i = d_{\sigma(i)}$.
La propriété (ii) s'écrit, puisque les éléments de $\omega(B_n)$ sont des matrices orthogonales, qu'il existe $\alpha \in B_n$ telle que $\omega(\alpha)D' = D\omega(\alpha)$.

Si (i) a lieu, alors (ii) aussi avec $\alpha = \sigma$ (question précédente).

Réciproquement, on suppose (ii) vérifiée ce qui nous donne $\alpha \in B_n$ telle que $\omega(\alpha)D' = D\omega(\alpha)$. La question 3 donne ainsi que

$$\omega(\alpha)D' = \omega(\alpha) \text{diag}(d_{\alpha(i)})_{1 \leq i \leq n}$$

et comme $\omega(\alpha)$ est inversible, $D' = \text{diag}(d_{\alpha(i)})_{1 \leq i \leq n}$ ce qui correspond à (i) avec $\sigma = \alpha$.

(i) et (ii) sont équivalentes

Remarque : il est tentant ici de raisonner par équivalence. C'est à déconseiller (surtout en début de problème) car les arguments dans les sens direct et réciproque sont parfois légèrement différents.

Fonctions de matrices symétriques

5. Le théorème spectral (s'appliquant aux matrices symétriques réelles) indique que S est diagonalisable en base orthonormée (i.e. via une matrice de passage orthogonale) :

$$\exists \Omega \in O_n(\mathbb{R}), \Omega S \Omega^{-1} = D \in D_n$$

L'inverse d'une matrice orthogonale est égale à sa transposée. De plus, les coefficients diagonaux de D sont les valeurs propres de S qui sont supposées dans I . Ainsi

$$\forall S \in S_n(I), \exists \Omega \in O_n(\mathbb{R}), \exists (s_i) \in I^n, S = \Omega^T \text{diag}(s_1, \dots, s_n) \Omega$$

Remarque : cette question est élémentaire. Nous vous conseillons la lecture du rapport de l'épreuve pour voir que même ici, on attend des justifications. Notez que plus ce qui précède est bien rédigé, moins vous serez sanctionnés pour des oublis mineurs (ce qui n'est pas anodin car le rapport semble attendre un niveau de détail qui nous semble parfois excessif).

6. Notons $J \subset I$ un ensemble tel que $\{s_i, i \in I\} = \{s_j, j \in J\}$ et $\forall i, j \in J, i \neq j \implies s_i \neq s_j$. J est ainsi un ensemble d'indice donnant les éléments distincts parmi les s_i .
En notant $d = \text{card}(J)$, l'application $P \in \mathbb{R}_{d-1}[X] \mapsto (P(s_j))_{j \in J} \in \mathbb{R}^d$ est linéaire et injective (si P de degré $\leq d-1$ admet d racines différentes, il est nul). Par dimension, c'est un isomorphisme. $(f(s_j))_{j \in J}$ admet donc un (unique) antécédent.
On a alors $P(s_i) = f(s_i)$ pour tout $i \in J$ et cela reste vrai pour les $i \in I$ par choix de J .

$$\boxed{\forall (s_i) \in I^n, \exists P \in \mathbb{R}[X], \forall i \in \llbracket 1, n \rrbracket, P(s_i) = f(s_i)}$$

On pourrait même, ce sont les formules d'interpolation de Lagrange, donner une expression explicite d'un P convenable :

$$P = \sum_{j \in J} f(s_j) L_j \quad \text{avec} \quad L_j = \prod_{i \in J \setminus \{j\}} \frac{X - s_i}{s_j - s_i}$$

Remarque : la seule difficulté est de penser à introduire un ensemble de valeurs distinctes pour utiliser l'interpolation de Lagrange. Une fois cet ensemble posé, il nous semble trop rapide d'invoquer l'interpolation de Lagrange sans l'un ou l'autre des arguments donnés ci-dessus.

7. Montrons par récurrence sur k que pour toute matrice inversible Q et toute matrice M , on a

$$(Q^{-1}MQ)^k = Q^{-1}M^kQ$$

- C'est vrai pour $k = 0$.
- Si c'est vrai au rang k alors

$$(Q^{-1}MQ)^{k+1} = (Q^{-1}MQ)^k Q^{-1}MQ = Q^{-1}M^kQQ^{-1}MQ = Q^{-1}M^{k+1}Q$$

et le résultat est vrai au rang $k + 1$.

Par combinaisons linéaires, on en déduit que

$$\forall P \in \mathbb{R}[X], \forall Q \in GL_n(\mathbb{R}), \forall M \in \mathcal{M}_n(\mathbb{R}), P(Q^{-1}MQ) = Q^{-1}P(M)Q$$

Par ailleurs, on a aussi

$$\forall k \in \mathbb{R}, \text{diag}(d_1, \dots, d_n)^k = \text{diag}(d_1^k, \dots, d_n^k)$$

et en combinant linéairement

$$\forall P \in \mathbb{R}[X], P(\text{diag}(d_1, \dots, d_n)) = \text{diag}(P(d_1), \dots, P(d_n))$$

Ici, on a donc (Ω est orthogonale et donc égale à son inverse), P étant le polynôme de la question 6,

$$\Omega^T \text{diag}(P(s_1), \dots, P(s_n)) \Omega = P(S) = (\Omega')^T \text{diag}(P(s'_1), \dots, P(s'_n)) \Omega'$$

De plus, $\text{Sp}(S) = \{s_1, \dots, s_n\} = \{s'_1, \dots, s'_n\}$ (les s'_i sont les s_i dans un autre ordre puisque ce sont les valeurs propres de S) et donc $P(s_i) = f(s_i)$ et $P(s'_i) = f(s'_i)$ pour tout i . Ainsi

$$\boxed{(\Omega')^T \text{diag}(f(s'_1), \dots, f(s'_n)) \Omega' = \Omega^T \text{diag}(f(s_1), \dots, f(s_n)) \Omega}$$

Comme $(AB)^T = B^T A^T$, il est immédiat que

$$\boxed{\Omega^T \text{diag}(f(s_1), \dots, f(s_n)) \Omega \in S_n(\mathbb{R})}$$

8. On a $\Omega^T(D_1 + \lambda D_2)\Omega = \Omega^T D_1 \Omega + \lambda \Omega^T D_2 \Omega$ qui me semble donner aisément

$$u(\varphi_1 + \lambda \varphi_2)(S) = u(\varphi_1)(S) + \lambda u(\varphi_2)(S)$$

Ceci étant vrai pour tout S , $u(\varphi_1 + \lambda \varphi_2) = u(\varphi_1) + \lambda u(\varphi_2)$ et u est linéaire.

La trace étant linéaire, par composition, v est linéaire ($v(\varphi) = \text{Tr}(u(\varphi))$).

Soient $\varphi : I \rightarrow \mathbb{R}$ et $x \in I$. On a $\text{diag}(x, \dots, x) = I_n^T(xI_n)I_n$ et donc (xI_n) est symétrique et on peut lui appliquer $u(\varphi)$

$$u(\varphi)(xI_n) = I_n^T \text{diag}(\varphi(x), \dots, \varphi(x))I_n = \varphi(x)I_n$$

$$\boxed{\forall x \in \mathbb{R}, u(\varphi)(xI_n) = \varphi(x)I_n}$$

Remarque : cette question permet de bien comprendre la nature des objets manipulés. Il ne s'agit, par exemple, pas de montrer la linéarité de $u(\varphi)$, qui agit sur des matrices, mais de u qui agit sur des fonctions. La linéarité de $u(\varphi)$ n'a d'ailleurs aucun sens puisque cette application est définie sur $S_n(I)$ qui n'est a priori pas un espace vectoriel. Il est crucial que les égalités écrites soient homogènes (on ne doit pas écrire une matrice égale à un scalaire par exemple).

9. Supposons que $u(\varphi) = 0$. Alors $\forall S \in S_n(\mathbb{R})$, $u(\varphi)(S) = 0$. Avec la question précédente (on utilise $S = xI_n$, $\forall x \in I$, $\varphi(x) = 0$ et donc $\varphi = 0$). Ainsi φ est injective (noyau restreint au neutre).

Si $n = 1$ alors une application V de $S_n(I)$ dans $S_n(\mathbb{R})$ s'assimile à une application φ de I dans $\mathbb{R}$ (une matrice de taille 1 est assimilée à un scalaire et $V((x)) = (\varphi(x))$) et on a $V = u(\varphi)$. u est ainsi surjective.

Si $n \geq 2$, on considère l'application V définie sur $S_n(I)$ constante égale à $E_{1,2} + E_{2,1}$. V est à valeurs dans $S_n(\mathbb{R})$. I étant non vide, il contient un élément x . On a $V(xI_n) = E_{1,2} + E_{2,1}$ qui n'est pas scalaire et n'est donc égale à $u(\varphi)(xI_n)$ pour aucune application φ . Ainsi, V n'a pas d'antécédent par u (pour tout φ , $u(\varphi) \neq V$ puisque xI_n n'a pas la même image par ces deux applications).

$$\boxed{u \text{ est non surjective sauf si } n = 1}$$

Remarque : une erreur courante est de conclure à la bijectivité à partir de l'injectivité par un argument de dimension. Ceci est bien sûr impossible ici car l'espace de travail (espace de fonction) n'est pas de dimension finie. La question précédente, qui nous a fait découvrir la nature des objets, doit permettre d'éviter cette très grossière erreur. Il est à noter que le rapport ne parle pas du cas $n = 1$ qui est pourtant particulier.

10. On suppose f polynomiale et il lui est donc associé un polynôme $P \in \mathbb{R}[X]$ tel que $\forall x \in I$, $P(x) = f(x)$.
Soit $S \in S_n(I)$. Il existe des éléments $s_i \in I$ et $\Omega \in O_n(\mathbb{R})$ telles que $S = \Omega^T \text{diag}(s_1, \dots, s_n)\Omega$ et on a

$$u(f)(S) = \Omega^T \text{diag}(f(s_1), \dots, f(s_n))\Omega = \Omega^T \text{diag}(P(s_1), \dots, P(s_n))\Omega$$

Avec les arguments de la question 7, ceci donne

$$u(f)(S) = P(\Omega^T \text{diag}(s_1, \dots, s_n)\Omega) = P(S)$$

Si f est polynomiale, il existe $P \in \mathbb{R}[X]$ tel que $\forall S \in S_n(I)$, $u(f)(S) = P(S)$

On suppose, réciproquement, que f est telle qu'un tel polynôme P existe. On a en particulier

$$\forall x \in I, f(x)I_n = u(f)(xI_n) = P(xI_n) = P(x)I_n$$

et donc $\forall x \in I$, $f(x) = P(x)$. f est donc polynomiale et la réciproque est vraie.

11. On suppose que $\forall x \in I$, $\varphi_k(x) \rightarrow \varphi(x)$. On se donne alors $S \in S_n(I)$. Il existe des éléments $s_i \in I$ et $\Omega \in O_n(\mathbb{R})$ telles que $S = \Omega^T \text{diag}(s_1, \dots, s_n) \Omega$ et on a

$$u(\varphi_k)(S) = \Omega^T \text{diag}(\varphi_k(s_1), \dots, \varphi_k(s_n)) \Omega = \Omega^T D_k \Omega$$

(D_k) converge vers $D = \text{diag}(\varphi(s_1), \dots, \varphi(s_n))$ et $M \mapsto \Omega^T M \Omega$ est continue (par exemple car elle est linéaire en dimension finie). Ainsi, $u(\varphi_k)(S)$ tend vers $u(\varphi)(S)$.

La convergence simple de (φ_k) vers φ sur I entraîne celle de $(u(\varphi_k))$ vers $u(\varphi)$ sur $S_n(I)$

La trace étant une application continue (linéaire en dimension finie)

La convergence simple de (φ_k) vers φ sur I entraîne celle de $(v(\varphi_k))$ vers $u(\varphi)$ sur $S_n(I)$

Remarque : les arguments de continuité utilisés sont centraux. On peut sans doute les remplacer en invoquant des "théorèmes généraux" ou "théorèmes d'opération".

Avec les mêmes notations, on a

$$u(\varphi_k)(S) - u(\varphi)(S) = \Omega^T \text{diag}(\varphi_k(s_1) - \varphi(s_1), \dots, \varphi_k(s_n) - \varphi(s_n)) \Omega$$

Munissons $\mathcal{M}_n(\mathbb{R})$ de la norme euclidienne canonique $\|\cdot\|_2$ définie par

$$\|M\|_2 = \sqrt{\text{Tr}(M^T M)}$$

La trace étant invariante par similitude, deux matrices orthogonalement semblables ont même norme (calcul aisé). Ainsi

$$\begin{aligned} \|u(\varphi_k)(S) - u(\varphi)(S)\|_2^2 &= \|\text{diag}(\varphi_k(s_1) - \varphi(s_1), \dots, \varphi_k(s_n) - \varphi(s_n))\|^2 \\ &= \sum_{i=1}^n |\varphi_k(s_i) - \varphi(s_i)|^2 \\ &\leq n \|\varphi_k - \varphi\|_{\infty, I}^2 \end{aligned}$$

Si (φ_k) converge uniformément vers φ sur I , le majorant, qui est indépendant de S , est de limite nulle. On a ainsi convergence uniforme de $(u(\varphi_k))$ vers $u(\varphi)$. Il est à noter que changer de norme sur $\mathcal{M}_n(\mathbb{R})$ est indifférent puisque toutes les normes sont équivalentes en dimension finie.

La convergence uniforme de (φ_k) vers φ sur I entraîne celle de $(u(\varphi_k))$ vers $u(\varphi)$ sur $S_n(I)$

Avec des notations, similaires, on a

$$|v(\varphi_k)(S) - v(\varphi)(S)| = \left| \sum_{i=1}^n (\varphi_k(s_i) - \varphi(s_i)) \right| \leq n \|\varphi_k - \varphi\|_{\infty, I}$$

et là encore

La convergence uniforme de (φ_k) vers φ sur I entraîne celle de $(v(\varphi_k))$ vers $v(\varphi)$ sur $S_n(I)$

Remarque : il était possible d'utiliser la norme définie par l'énoncé en lieu et place de la norme euclidienne canonique. L'intérêt de cette dernière est l'invariance par orthosimilitude comme évoqué dans l'argumentation.

Norme et convexité

12. Soit $S \in S_n(\mathbb{R})$. Par théorème spectral, il existe une base orthonormée $(X_1, \dots, X_n)$ de $\mathbb{R}^n$ (que l'on assimile à $\mathcal{M}_{n,1}(\mathbb{R})$) formée de vecteurs propres pour S . Notons λ_i la valeur propre associée à X_i . Le spectre de S est ainsi constitué des λ_i . Soit $X \in \mathbb{R}^n$ que l'on décompose en $X = x_1 X_1 + \dots + x_n X_n$. On a alors (la base étant orthonormée)

$$\min(\text{Sp}(S)) \sum_{i=1}^n x_i^2 \leq X^T S X = \sum_{i=1}^n \lambda_i x_i^2 \leq \max(\text{Sp}(S)) \sum_{i=1}^n x_i^2$$

Comme $X^T X = x_1^2 + \dots + x_n^2$ (toujours car la base est orthonormée), on a donc

$$\forall X \in \Sigma, \min(\text{Sp}(S)) \leq X^T S X \leq \max(\text{Sp}(S))$$

Le majorant (resp minorant) est atteint pour $X = X_j$ associé à une valeur propre maximale (resp minimale) et c'est donc un maximum (resp minimum).

$$\min(\text{Sp}(S)) = \min\{X^T S X ; X \in \Sigma\} \text{ et } \max(\text{Sp}(S)) = \max\{X^T S X ; X \in \Sigma\}$$

Remarque : la question est élémentaire à partir du moment où l'on utilise une base ortho-normée de diagonalisation pour S . Mais ceci, équivalent du théorème spectral, doit être quasiment un réflexe.

13. Soient $A, B \in S_n(\mathbb{R})$ et $\lambda \in [0, 1]$. On a

$$\forall X \in \Sigma, X^T((1-\lambda)A + \lambda B)X = (1-\lambda)X^T A X + \lambda X^T B X$$

Comme $\lambda \geq 0$ et $1-\lambda \geq 0$, la question précédente donne

$$\min((1-\lambda)\text{Sp}(A) + \lambda \min(\text{Sp}(B))) \leq X^T((1-\lambda)A + \lambda B)X \leq (1-\lambda) \max(\text{Sp}(A)) + \lambda \max(\text{Sp}(B))$$

puis

$$\text{Sp}((1-\lambda)A + \lambda B) \subset [(1-\lambda) \min(\text{Sp}(A)) + \lambda \min(\text{Sp}(B)), (1-\lambda) \max(\text{Sp}(A)) + \lambda \max(\text{Sp}(B))]$$

Comme I est un intervalle, il est convexe. Comme A, B ont un spectre inclus dans I , les bornes de l'intervalle ci-dessus sont dans I et l'intervalle est donc inclus dans I . Ainsi $(1-\lambda)A + \lambda B \in S_n(I)$ ($S_n(\mathbb{R})$ est un sous-espace, on reste dans celui-ci par combinaisons linéaires). On a montré que

$S_n(I)$ est une partie convexe de $S_n(\mathbb{R})$

Pour montrer que ρ est une norme sur $S_n(\mathbb{R})$, on a quatre propriétés à prouver.

- ρ est immédiatement positive.
- Si $\rho(S) = 0$ alors 0 est la seule valeur propre de S et comme S est diagonalisable, elle est nulle. Ceci montre l'axiome de séparation.
- Soient $S \in S_n(\mathbb{R})$ et $\mu \in \mathbb{R}$. Les valeurs propres de μS sont les $\mu\lambda$ pour $\lambda \in \text{Sp}(S)$ (si S est semblable à $\text{diag}(\lambda_1, \dots, \lambda_n)$ alors μS est semblable à $\text{diag}(\mu\lambda_1, \dots, \mu\lambda_n)$). Comme $x \mapsto |\mu|x$ est croissante, le maximum des $|\mu||\lambda|$ est égal à $|\mu|$ fois le maximum des $|\lambda|$. Ainsi $\rho(\mu S) = |\mu|\rho(S)$ et on a l'homogénéité.
- Soient $A, B \in S_n(\mathbb{R})$. On a alors $\text{Sp}(A) \subset [-\rho(A), \rho(A)]$ et idem pour B . On montre comme plus haut que le spectre de $A + B$ est dans $[-\rho(A) - \rho(B), \rho(A) + \rho(B)]$ et donc $\rho(A + B) \leq \rho(A) + \rho(B)$. Ceci donne l'inégalité triangulaire.

 ρ est une norme sur $S_n(\mathbb{R})$

Continuité des fonctions de matrices symétriques

14. χ va de $S_n(\mathbb{R})$ dans $\mathbb{R}_n[X]$. Les espaces étant de dimension finie, le choix de norme n'importe pas dans l'étude de continuité.

Les coefficients de $\chi(S)$ sont des fonctions polynomiales de ceux de S et donc continues (ce qui est immédiat quand on munit $S_n(\mathbb{R})$ de la norme infinie). Ainsi, les fonctions coordonnées de χ (dans la base canonique de $\mathbb{R}_n[X]$) sont continues. Ceci revient à la continuité de χ (pour une application à valeurs dans un evn de dimension finie, la continuité équivaut à la continuité des fonctions coordonnées ce que l'on montre en utilisant la norme infinie relative à la base choisie, toutes les normes étant équivalentes).

 χ est continue

Remarque : la lecture du rapport (que nous trouvons par ailleurs souvent inutilement pédant) nous indique les points clés qui étaient attendus et en particulier la compréhension des objets et l'équivalence des normes. La simple mention de la "continuité du déterminant" était largement insuffisante.

15. On suppose que $\rho(M_k - M) \rightarrow 0$ et on a donc $\rho(M_k) \rightarrow \rho(M)$ (par seconde forme de l'inégalité triangulaire).

Ainsi, la suite $(\rho(M_k))$ est une suite bornée, disons majorée par un réel M . En notant $\Lambda_k = (\lambda_{1,k}, \dots, \lambda_{n,k})$, on a

$$\forall k, \|\Lambda_k\| = \max_{1 \leq i \leq n} |\lambda_{i,k}| \leq M$$

On a ainsi $(\Lambda_k)_{k \in \mathbb{N}}$ qui est bornée dans $\mathbb{R}^n$ et admet une valeur d'adhérence $\Lambda = (\lambda_1, \dots, \lambda_n)$. En notant φ l'extractrice associé, $\lambda_{i,\varphi(k)} \rightarrow \lambda_i$ et le caractère croissant des Λ_k entraîne celui de Λ .

 $(\Lambda_k)_{k \in \mathbb{N}}$ possède une valeur d'adhérence croissante

16. On sait que χ est continue et donc $\chi(M_k) \rightarrow \chi(M)$. A fortiori, $\chi(M_{\alpha(k)}) \rightarrow \chi(M)$. Or, avec les notations utilisées en question précédente,

$$\chi_{M_{\alpha(k)}} = \prod_{i=1}^n (X - \lambda_{i,\alpha(k)})$$

Notons μ_i la limite de $\lambda_{i,\alpha(k)}$ (qui existe puisque l'on suppose que $(\Lambda_{\alpha(k)})$ converge), les théorèmes d'opération donnent

$$\chi_{M_{\alpha(k)}} \rightarrow \prod_{i=1}^n (X - \mu_i)$$

Par unicité de la limite, on a donc

$$\chi_M = \prod_{i=1}^n (X - \mu_i)$$

La suite (μ_i) étant croissante (par croissance des Λ_k et passage à la limite dans les inégalités), c'est la suite croissante des valeurs propres de M :

$$\boxed{\Lambda_{\alpha(k)} \xrightarrow[k \rightarrow +\infty]{} \text{Sp}_{\uparrow}(M)}$$

17. La suite (Λ_k) possède une valeur d'adhérence (question 15) et celle-ci est forcément $\text{Sp}_{\uparrow}(M)$ (question 16). On a une suite à valeurs dans un compact (ses éléments appartiennent à une boule fermée, on l'a noté en question 15) qui possède une unique valeur d'adhérence et cette suite est donc convergente. On a montré que $\text{Sp}_{\uparrow}(M_k) \rightarrow \text{Sp}_{\uparrow}(M)$. Par caractérisation séquentielle de la limite,

$$\boxed{\text{Sp}_{\uparrow} \text{ est continue}}$$

Remarque : on a utilisé le résultat stipulant que si une suite à valeurs dans un compact possède au plus une valeur d'adhérence, alors elle converge.

18. $O_n(\mathbb{R})$ est fermé comme image réciproque du fermé $\{I_n\}$ par l'application continue $M \mapsto M^T M$ (chaque coordonnée de $M^T M$ est polynomiale en les coefficients de M par exemple ou encore continuité du passage à la transposée, linéaire, et de l'application $(A, B) \mapsto AB$ par bilinéarité, le tout en dimension finie). C'est aussi une partie bornée car $\forall M \in O_n(\mathbb{R}), \|M\| \leq 1$ (chaque colonne est de norme 1 dans $\mathbb{R}^n$ euclidien). Comme $\mathcal{M}_n(\mathbb{R})$ est de dimension finie,

$$\boxed{O_n(\mathbb{R}) \text{ est une partie compacte de } \mathcal{M}_n(\mathbb{R})}$$

Remarque : si on lit attentivement le rapport, il faudrait détailler plus des points comme la continuité de $M \mapsto M^T M$ ou le fait que les coefficients d'une matrice orthogonale sont de module plus petit que 1. A ce niveau du problème, il nous semble pourtant que ces points sont "évidents". Nous sommes convaincus que ces précisions ne sont vraiment attendues que dans les copies où le candidat à essayé de récolter des points sur des questions faciles.

19. Soit $\varphi \in C^0(I, \mathbb{R})$. Soit (M_k) une suite d'éléments de $S_n(I)$ qui converge dans $S_n(I)$ vers une matrice M . On note encore $\Lambda_k = (\lambda_{1,k}, \dots, \lambda_{n,k})$ son spectre ordonné. Pour chaque entier k , il existe $\Omega_k \in O_n(\mathbb{R})$ telle que

$$M_k = \Omega_k^T \text{diag}(\lambda_{i,k}) \Omega_k$$

On a alors

$$u_\varphi(M_k) = \Omega_k^T \text{diag}(\varphi(\lambda_{i,k})) \Omega_k$$

On vient de voir que Λ_k converge vers le spectre ordonné $\Lambda = (\lambda_1, \dots, \lambda_n)$ de M .

Par ailleurs, comme $O_n(\mathbb{R})$ est un compact, il existe une extraite $(\Omega_{\alpha(k)})$ qui converge vers $\Omega \in O_n(\mathbb{R})$. On a alors

$$M_{\alpha(k)} \rightarrow \Omega^T \text{diag}(\varphi(\lambda_i)) \Omega \quad \text{et} \quad u_\varphi(M_{\alpha(k)}) \rightarrow \Omega^T \text{diag}(\varphi(\lambda_i)) \Omega$$

Par unicité de la limite, on a $M = \Omega^T \text{diag}(\varphi(\lambda_i)) \Omega$ et la seconde limite vaut ainsi $u(\varphi)(M)$. On a donc

$$u_\varphi(M_{\alpha(k)}) \rightarrow u(\varphi)(M)$$

Si on considère la suite $(u(\varphi)(M_k))$, on peut en fait, en travaillant sur une extraite quelconque comme ci-dessus, montrer que $u(\varphi)(M)$ est la seule valeur d'adhérence possible. Or, toutes les suites $(\varphi(\lambda_{i,k}))_{k \in \mathbb{N}}$ étant bornée, la suite $(u(\varphi)(M_k))$ est bornée (au sens de ρ , c'est quasi immédiat et le choix de norme n'importe pas). On est dans la même situation qu'en question 17 et on peut affirmer que $(u(\varphi)(M_k))$ converge vers $u(\varphi)(M)$. $u(\varphi)$ est donc continue. Comme la trace est continue, $v(\varphi)$ est aussi continue.

si $\varphi \in \mathcal{C}^0(I, \mathbb{R})$, alors $u(\varphi)$ et $v(\varphi)$ sont continues

Remarque : il y a deux points clés dans notre raisonnement. Le premier est l'utilisation de la caractérisation séquentielle pour la continuité (naturelle avec ce qui précède puisque la compacité s'exprime bien en terme séquentiel). Le second est encore le résultat de convergence pour une suite à valeurs dans un compact. Ces arguments avancés, on peut légitimement ne pas s'attarder sur la rédaction complète (ici, le renvoi à la technique de la question 17).

Convexité des fonctions de matrices symétriques

20. Pour une matrice M quelconque, on a $[M]_{i,j} = E_j^T M E_i$ où $E_1, \dots, E_n$ sont les vecteurs colonnes associés à la base canonique. Soit $\Omega \in O_n(\mathbb{R})$ et soit $U = \Omega^T S \Omega$. On a alors

$$[U]_{j,j} = Y_j^T S Y_j \quad \text{avec} \quad Y_j = \Omega E_j$$

Les vecteurs Y_j étant dans Σ (car $E_j \in \Sigma$ et O orthogonale), la question 12 montre que

$$\forall j, \min(\text{Sp}(S)) \leq [U]_{j,j} \leq \max(\text{Sp}(S))$$

Majorant et minorant sont dans I et I est un intervalle. Ainsi

$$\forall j \in \llbracket 1, n \rrbracket, [U]_{j,j} \in I$$

Notons $\lambda_1, \dots, \lambda_n$ les valeurs propres de S (comptées avec multiplicité).

Soit $\Omega \in \mathcal{U}_S$. U est orthogonalement semblable à S et il existe $\Omega \in O_n(\mathbb{R})$ telle que $U = \Omega^T \text{diag}((\lambda_i)_{1 \leq i \leq n}) \Omega$. Avec les notations précédentes, on a

$$\forall j, f([U]_{j,j}) = f(Y_j^T \text{diag}((\lambda_i)_{1 \leq i \leq n}) Y_j) = f\left(\sum_{i=1}^n \lambda_i [Y_j]_i^2\right)$$

Comme $Y_j \in \Sigma$, les $[Y_j]_i^2$ sont positifs de somme 1. Par convexité de f , on a donc

$$\forall j, f([U]_{j,j}) \leq \sum_{i=1}^n [Y_j]_i^2 f(\lambda_i)$$

En sommant ces relations, on obtient

$$\sum_{j=1}^n f([U]_{j,j}) \leq \sum_{j=1}^n \sum_{i=1}^n [Y_j]_i^2 f(\lambda_i) = \sum_{i=1}^n \left(f(\lambda_i) \sum_{j=1}^n [Y_j]_i^2 \right)$$

Les Y_j sont en fait les colonnes de Ω et $[Y_j]_i = [\Omega]_{i,j}$. Les lignes de Ω forment aussi une famille orthonormée et les sommes intérieures ci-dessus valent 1. On a donc

$$\sum_{j=1}^n f([U]_{j,j}) \leq \sum_{i=1}^n f(\lambda_i) = v(f)(S)$$

L'inégalité est une égalité quand $\Omega = I_n$ et on a donc

$$\max \left\{ \sum_{k=1}^n f([U]_{k,k}) ; U \in \mathcal{U}_S \right\} = v(f)(S)$$

21. Avec la question précédente, il existe une matrice $\Omega \in O_n(\mathbb{R})$ telle qu'en notant $U = \Omega^T((1-t)A + tB)\Omega$ on ait (notons que $S_n(I)$ est convexe et donc $(1-t)A + tB \in S_n(I)$ puisque $A, B \in S_n(I)$)

$$v(f)((1-t)A + tB) = \sum_{k=1}^n f([U]_{k,k})$$

Or, $[U]_{k,k} = (1-t)[\Omega^T A \Omega]_{k,k} + t[\Omega^T B \Omega]_{k,k}$ et ainsi (convexité de f)

$$v(f)((1-t)A + tB) \leq (1-t) \sum_{k=1}^n f([\Omega^T A \Omega]_{k,k}) + t \sum_{k=1}^n f([\Omega^T B \Omega]_{k,k})$$

La question précédente permet de majorer les sommes par $v(f)(A)$ et $v(f)(B)$. En multipliant par t et $1-t$ on ne change pas le sens des inégalités. On trouve

$$v(f)((1-t)A + tB) \leq (1-t)v(f)(A) + tv(f)(B)$$

22. On vient de voir que la convexité de f entraîne celle de $v(f)$.
On suppose, réciproquement, que $v(f)$ est convexe. On applique cette propriété avec $A = xI_n$ et $B = yI_n$ pour $x, y \in I$. On obtient alors
- $$nf((1-t)x + ty) = v(f)((1-t)xI_n + tyI_n) \leq (1-t)v(f)(xI_n) + tv(f)(yI_n) = n((1-t)f(x) + tf(y))$$
- ce qui donne la convexité de f .

$$f \text{ est convexe sur } I \text{ ssi } v(f) \text{ l'est sur } S_n(I)$$



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2021

ÉPREUVE D'INFORMATIQUE MP

Durée de l'épreuve : 3 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

Cette épreuve concerne uniquement les candidats de la filière MP.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE - MP

L'énoncé de cette épreuve comporte 11 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Préliminaires

L'épreuve est composée d'un problème unique, comportant 30 questions. Après cette section de préliminaires et une section de présentation du *jeu du solitaire*, le problème est divisé en deux sections indépendantes, pages 4 et 8. Dans la première section, nous étudions le jeu du solitaire sur un tablier de forme classique par des méthodes de théorie des graphes. Dans la seconde section, nous étudions le jeu du solitaire sur un tablier en bande unidimensionnelle par des méthodes de théorie des automates. Pour répondre à une question, un candidat pourra réutiliser le résultat d'une question antérieure, même s'il n'est pas parvenu à établir ce résultat.

Concernant la programmation

Il faudra coder des fonctions à l'aide du langage de programmation OCaml, en reprenant l'entête de fonction fournie par le sujet, sans nécessairement reprendre la déclaration des types. Pour écrire une fonction, on pourra faire appel à d'autres fonctions définies dans les questions précédentes ; on pourra aussi définir des fonctions auxiliaires. Quand l'énoncé demande de coder une fonction, il n'est pas nécessaire de justifier que celle-ci est correcte, sauf si l'énoncé le demande explicitement. Si les paramètres d'une fonction à coder sont supposés vérifier certaines hypothèses, il ne sera pas utile de tester si les hypothèses sont bien vérifiées dans le code de la fonction.

Dans tout l'énoncé, un même identificateur écrit dans deux polices de caractère différentes désignera la même entité, mais du point de vue mathématique pour la police en italique (par exemple n) et du point de vue informatique pour celle en romain avec espacement fixe (par exemple `n`).

Aide à la programmation en OCaml

Opérations sur les listes : Sans qu'il ne soit imposé de coder dans un style de programmation les utilisant, on pourra s'appuyer sur les fonctions suivantes :

- `append`, de type `'a list -> 'a list -> 'a list`, qui concatène deux listes en une seule liste ;
- `filter`, de type `('a -> bool) -> 'a list -> 'a list`, telle que `filter p l` renvoie la liste des éléments x de la liste ℓ tels que le prédicat $p(x)$ vaut `true`, en respectant l'ordre de la liste ;
- `flatten`, de type `'a list list -> 'a list`, qui concatène une liste de listes en une seule liste ;
- `fold`, de type `('a -> 'b -> 'a) -> 'a -> 'b list -> 'a`, telle que `fold f a [b1; ..; bn]` renvoie `f(..(f(f a b1) b2)..) bn` ;
- `map`, de type `('a -> 'b) -> 'a list -> 'b list`, telle que `map f [a1; ..; an]` renvoie la liste `[f a1; ..; f an]`.

Opérations sur les couples : On rappelle que

- la fonction `fst`, de type `'a * 'b -> 'a`, renvoie le premier terme d'un couple ;
- la fonction `snd`, de type `'a * 'b -> 'b`, renvoie le second terme d'un couple.

Opérateurs logiques sur les entiers : Nous supposons que les entiers naturels, de type `int` en OCaml, sont systématiquement codés à l'aide de 62 bits. Dans un texte mathématique, on pourra signaler la représentation binaire par une barre horizontale. Pour tous entiers n et k avec $0 \leq n < 2^{62}$ et $0 \leq k < 62$, nous notons $[n]_k$ le k^{e} bit de poids le plus faible de la représentation binaire de n , ce qui permet d'écrire $n = \overline{[n]_{61}[n]_{60} \cdots [n]_0}$.

Le tableau ci-dessous rappelle le résultat de quelques-uns des opérateurs logiques de OCaml qui agissent bit à bit sur deux entiers naturels a et b . Tous ces opérateurs renvoient un élément de type `int`. Dans ce tableau, les symboles $\wedge$, $\vee$, $\oplus$ et $\neg$ désignent respectivement le « et », le « ou », le « ou exclusif » entre deux booléens et la négation d'un booléen.

Opérateur	Nom	Résultat exprimé sur 62 bits
<code>a land b</code>	Et logique bit à bit	$\overline{([a]_{61} \wedge [b]_{61}) \cdots ([a]_1 \wedge [b]_1)([a]_0 \wedge [b]_0)}$
<code>a lor b</code>	Ou logique bit à bit	$\overline{([a]_{61} \vee [b]_{61}) \cdots ([a]_1 \vee [b]_1)([a]_0 \vee [b]_0)}$
<code>a lxor b</code>	Xor logique bit à bit	$\overline{([a]_{61} \oplus [b]_{61}) \cdots ([a]_1 \oplus [b]_1)([a]_0 \oplus [b]_0)}$
<code>lnot a</code>	Non logique bit à bit	$\overline{(\neg[a]_{61}) \cdots (\neg[a]_1)(\neg[a]_0)}$
<code>a lsl b</code>	Décalage vers la gauche	$\overline{[a]_{61-b} \cdots [a]_1 [a]_0 \underbrace{0 \cdots 0}_{b \text{ zéros à droite}}}$
<code>a lsr b</code>	Décalage vers la droite	$\overline{\underbrace{0 \cdots 0}_{b \text{ zéros à gauche}} [a]_{61} \cdots [a]_{b+1} [a]_b}$

Par exemple, lorsque $a = 6$ et $b = 3$, les écritures sur 62 bits de a et b sont $a = \overline{0 \cdots 0110}$ et $b = \overline{0 \cdots 0011}$ et

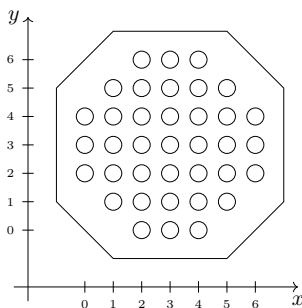
- le calcul de `a land b` vaut l'entier 2, dont l'écriture sur 62 bits est $\overline{0 \cdots 0010}$;
- le calcul de `a lor b` vaut l'entier 7, dont l'écriture sur 62 bits est $\overline{0 \cdots 0111}$;
- le calcul de `a lxor b` vaut l'entier 5, dont l'écriture sur 62 bits est $\overline{0 \cdots 0101}$;
- le calcul de `a lsl b` renvoie 48, dont l'écriture sur 62 bits est $\overline{0 \cdots 0110000}$;
- le calcul de `a lsr b` renvoie 0 puisque les deux bits non nuls de a sont sortis de la représentation.

L'entier 2^k , avec $0 \leq k < 62$, peut commodément se définir en OCaml par `1 lsl k`.

Le jeu du solitaire

Le *jeu du solitaire* se joue sur un tablier percé d'encoches disposées en quadrillage et remplissant une certaine forme géométrique.

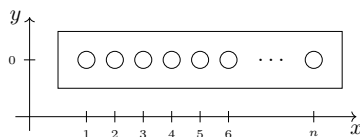
Parmi les tabliers les plus fréquents, le *tablier européen* est formé de 37 encoches organisées en octogone



dont on peut décrire les coordonnées par

$$\{(x, y) \in \mathbb{N}^2; 0 \leq x \leq 6, 0 \leq y \leq 6 \text{ et } |x - 3| + |y - 3| \leq 4\}.$$

D'autres tabliers existent tels que le tablier rectangulaire de dimension $n \times 1$, où $n \geq 1$ est un entier,



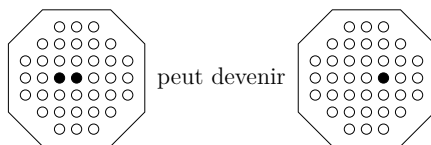
et dont on peut décrire les coordonnées par

$$\{(x, y) \in \mathbb{N}^2; 1 \leq x \leq n \text{ et } y = 0\}.$$

Sur un tablier, toute encoche de coordonnées (x, y) possède au plus quatre encoches *voisines*, à savoir les encoches de coordonnées $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$ et $(x, y - 1)$ si elles existent.

Nous appelons *motif* tout sous-ensemble d'encoches dans le tablier. Nous disons qu'un motif est *ponctuel* s'il est de cardinal 1.

En début de jeu, des fichets (ou des billes) viennent se loger dans les encoches d'un motif initial. En cours de jeu, un *coup simple* consiste à déplacer l'un des fichets en sautant par-dessus l'une des encoches voisines pour atteindre l'encoche immédiatement après. Il peut être joué à condition que l'encoche voisine en question soit occupée et que le fichet déplacé retombe dans une encoche inoccupée. À l'issue d'un coup simple, le fichet planté dans l'encoche au-dessus de laquelle a eu lieu le saut est retiré. Par exemple,



En cours de jeu, un *coup composé* consiste à jouer zéro, un ou plusieurs coups simples en déplaçant le même fichet. Par exemple,



Une suite de motifs obtenus en jouant une suite de coups s'appelle une *partie*. L'objectif de la joueuse ou du joueur est de transformer un motif en un autre motif par une suite de coups.

1 Partie jouée sur le tablier européen en un minimum de coups

Dans toute la section 1 du sujet, une joueuse joue sur le tablier européen. Elle souhaite transformer un motif initial en un motif final par un minimum de coups composés.

- 1 – À titre préliminaire, nommer le type de parcours de graphe le plus approprié pour déterminer un chemin entre deux sommets qui minimise le nombre d'arcs traversés. Quelle politique de mise en attente de sommets caractérise ce parcours ?

1.1 Numérotation du tablier

Nous numérotions l'encoche de coordonnées $(x, y) \in \mathbb{N}^2$ dans le tablier européen par l'entier $z = 8y + x \in \mathbb{N}$ (voir figure 1, page 5). Nous notons l'ensemble des numéros d'encoches du tablier européen par

$$\mathcal{E} = \{8y + x; (x, y) \in \mathbb{N}^2 \text{ avec } 0 \leq x, y \leq 6 \text{ et } |x - 3| + |y - 3| \leq 4\} \subseteq \mathbb{N}.$$

Indication OCaml : En OCaml, on peut retrouver les coordonnées (x, y) d'une encoche à partir de son numéro z en écrivant $z \bmod 8$ pour obtenir l'abscisse x et $z/8$ pour obtenir l'ordonnée y . On peut calculer la valeur absolue d'un entier avec `abs`.

- 2 – Écrire une fonction `numero_interieur (z:int) : bool`, qui teste si un entier naturel z est le numéro d'une encoche du tablier européen $\mathcal{E}$.

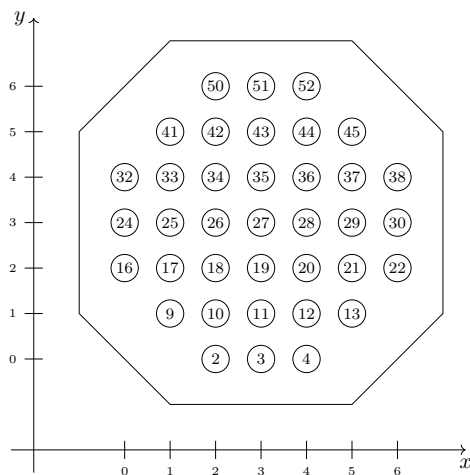


FIGURE 1 – Numérotation des encoches du tablier européen

- 3 – En énumérant les entiers naturels inférieurs à 52 et à l'aide de la fonction `numero_interieur` introduite à la question 2, définir une constante globale `numeros_europeens`, de type `int list`, égale à la liste des numéros d'encoches du tablier européen.
- 4 – En supposant qu'elles existent dans le tablier, donner par une expression mathématique le numéro des quatre encoches voisines de l'encoche de numéro z .

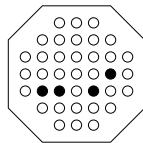
1.2 Motifs

Pour représenter un motif $M \subseteq \mathcal{E}$, nous utilisons un entier naturel m dont nous exploitons une partie des bits de l'écriture binaire comme suit : pour tout numéro d'encoche $z \in \mathcal{E}$, nous fixons

$$[m]_z = \begin{cases} 1 & \text{si } z \in M \\ 0 & \text{si } z \notin M \text{ ou si } z \notin \mathcal{E} \end{cases}.$$

Lorsqu'un motif est ponctuel, c'est-à-dire qu'il ne contient qu'une seule encoche, sa représentation est simplement une puissance de 2.

Par exemple, le motif à quatre encoches suivant



se représente par le nombre entier

$$m_0 = 2^{29} + 2^{20} + 2^{18} + 2^{17} = 538312704 \in \mathbb{N}.$$

Indication OCaml : Pour représenter les motifs et les motifs ponctuels, nous définissons les types `motif` et `ponctuel` par la déclaration suivante :

```
type motif = int;;
type ponctuel = motif;;
```

L'exemple ci-dessus se déclare à l'aide des opérateurs logiques en OCaml par

```
let m0 = (1 lsl 29) lor (1 lsl 20) lor (1 lsl 18) lor (1 lsl 17);;
```

□ 5 – Écrire une fonction `numero_vers_ponctuel (z:int) : ponctuel` qui renvoie le motif ponctuel $\{z\} \subseteq \mathcal{E}$ formé d'une seule encoche de numéro z .

□ 6 – Écrire une fonction `numeros_vers_motif (l:int list) : motif` qui renvoie le motif formé des encoches de numéro appartenant à la liste ℓ .

□ 7 – En s'appuyant sur la constante `numeros_europeens` introduite à la question 3, définir une constante globale `motif_europeen`, de type `motif`, égale au motif formé de toutes les encoches du tablier européen.

□ 8 – Démontrer que la fonction suivante

```
let est_ponctuel (m:motif) : bool =
  (m>0) && ((m land (m-1)) = 0) ;;
```

renvoie `true` si le motif m est un motif ponctuel et `false` sinon.

□ 9 – Écrire, à l'aide des opérateurs logiques sur les entiers, une fonction `inclus (m:motif) (p:ponctuel) : bool` qui renvoie `true` si le motif ponctuel p est inclus dans le motif m ou `false` dans le cas opposé. Donner une preuve mathématique du résultat.

□ 10 – Écrire, à l'aide des opérateurs logiques sur les entiers, une fonction `voisin_g (p:ponctuel) : ponctuel` qui calcule la translation par une encoche vers la gauche du motif ponctuel p . Si le motif ponctuel p sort du tablier, la valeur de retour est le motif vide 0.

Dans la suite du problème, nous supposons avoir codé des fonctions similaires `voisin_d`, `voisin_h` et `voisin_b` qui calculent les décalages d'un motif ponctuel respectivement vers la droite, vers le haut et vers le bas. Nous déclarons une constante globale par `let voisins = [voisin_g; voisin_d; voisin_h; voisin_b];;`

1.3 Coups simples et composés

□ 11 – Écrire une fonction `coup_simple ((m, p) : motif * ponctuel) : (motif * ponctuel) list` qui prend en entrée un motif quelconque m et un motif ponctuel p contenu dans m et qui construit la liste des couples (m', p') où m' est un motif obtenu à partir de m à la suite du déplacement par un coup simple du fichet initialement placé dans p et où p' est le motif ponctuel repérant le même fichet après son déplacement.

□ 12 – Écrire une fonction `coup_compose ((m, p) : motif * ponctuel) : (motif * ponctuel) list` qui prend en entrée un motif quelconque m et un motif ponctuel p contenu dans m et qui construit la liste des couples (m', p') où m' est un motif obtenu à partir de m à la suite du déplacement par un coup composé du fichet initialement placé dans p et où p' est le motif ponctuel repérant le même fichet après son déplacement.

□ 13 – Écrire une fonction `mouvements (m:motif) : motif list` dont la valeur de retour est la liste de tous les motifs que l'on peut obtenir en jouant un coup composé depuis le motif m , n'importe quel fichet pouvant être déplacé.

1.4 Partie de longueur minimale

Indication OCaml : Nous utilisons dans cette sous-section une structure de données de dictionnaire, avec modification en place, permettant d'associer des clés de type `motif` et des valeurs de type `int`. Nous notons le type de cette structure `dico`. Pour utiliser ces dictionnaires, nous disposons des fonctions suivantes :

- la fonction `create`, de type `unit -> dico`, qui permet de créer un dictionnaire vide ;
- la fonction `add`, de type `dico -> motif -> int -> unit`, telle que `add d m n` permet d'ajouter au dictionnaire d une association entre la clé m et l'entier n (si une association de clé m existe déjà, elle est remplacée) ;
- la fonction `mem`, de type `dico -> motif -> bool`, telle que `mem d m` renvoie le booléen `true` si la clé m est membre du dictionnaire d ou `false` sinon.
- la fonction `find`, de type `dico -> motif -> int`, telle que `find d m` renvoie la valeur associée à la clé m si la clé est membre du dictionnaire d ou une erreur sinon.

□ 14 – Écrire une fonction `add_and_mem (d:dico) (s:int) (m:motif) : bool` qui ajoute l'association entre le motif m et l'entier s dans le dictionnaire d si le motif m est initialement absent du dictionnaire d . La valeur de retour de la fonction est `true` si le dictionnaire a été modifié et `false` sinon.

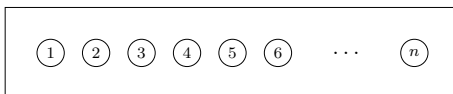
□ 15 – Écrire une fonction `strate (d:dico) (s:int) (l:motif list) : motif list` qui, pour tout motif m que l'on peut obtenir après un coup composé à partir d'un motif de la liste ℓ , ajoute l'association entre le motif m et l'entier s au dictionnaire d si le motif m n'est pas présent dans le dictionnaire d . La valeur de retour de la fonction est la liste des motifs que la fonction ajoute.

□ 16 – Dans cette question, nous supposons qu'il est possible de passer d'un motif m_i à un motif m_f par une suite de coups. Écrire une fonction `partie_minimale (mi:motif) (mf:motif) : int` qui calcule le nombre minimal de coups composés à jouer pour passer du motif initial m_i au motif final m_f . On pourra utiliser un dictionnaire qui associe des motifs au nombre de coups minimal pour les atteindre depuis le motif m_i .

□ 17 – Peut-on considérer que la réponse donnée à la question 16 observe le parcours nommé à la question 1 ? Introduire un graphe puis argumenter.

2 Reconnaissance des motifs résolubles sur le tablier unidimensionnel

Dans toute la section 2 du sujet, une joueuse joue sur des tabliers rectangulaires de dimension $n \times 1$, où n est un entier naturel pouvant varier d'une partie à une autre.



Dans cette section, le but de la joueuse est de faire disparaître tous les fichets sauf un par une suite de coups, autrement dit de jouer une partie qui amène le motif initial à un motif ponctuel. Lorsque ceci est possible, nous disons que le motif initial est *résoluble* et que la partie est *gagnante*.

Nous introduisons l'alphabet binaire $\Sigma = \{0, 1\}$ et notons Σ^* l'ensemble des mots sur Σ . Pour tout mot $w = \sigma_1 \dots \sigma_n \in \Sigma^*$, nous notons $[w]_k$ le k^{e} symbole σ_k de w .

Le mot d'un motif M du tablier rectangulaire de dimension $n \times 1$ est le mot w formé de n symboles de l'alphabet Σ tel que, pour k variant entre 1 et n , le symbole $[w]_k$ vaut 1 si la k^{e} encoche à partir de la gauche appartient au motif M et vaut 0 sinon. Par exemple, sur ce tablier, le motif $\bullet \circ \bullet \bullet \circ$ est décrit par le mot $10110 \in \Sigma^*$.

Dans toute cette section, nous ne considérons que des coups simples. Une partie est par conséquent une suite finie W de mots $w_0, w_1, \dots, w_m$ de même longueur telle que pour tout indice i compris entre 0 et $m - 1$, le mot w_{i+1} est obtenu à partir du mot w_i en remplaçant dans le mot w_i un facteur 110 par les symboles 001 ou bien en remplaçant un facteur 011 par les symboles 100. Dans le premier cas, nous parlons de coup vers la droite; dans le second cas, nous parlons de coup vers la gauche. Une partie est gagnante si, de plus, le dernier mot w_m de la partie ne comprend le symbole 1 qu'une seule fois.

L'objectif de cette section est d'étudier le langage $\mathcal{R} \subseteq \Sigma^*$ des mots de longueur quelconque de motifs résolubles.

2.1 Mots de motifs ponctuels

Nous notons $\mathcal{P} \subseteq \Sigma^*$ le langage des mots de longueur quelconque de motifs ponctuels.

- ☐ 18 – Donner, sans justification, une expression rationnelle qui décrit le langage $\mathcal{P}$.
- ☐ 19 – Dessiner, sans justification, un automate fini qui reconnaît le langage $\mathcal{P}$.
- ☐ 20 – Le langage $\mathcal{P}$ est-il un langage local ?

2.2 Bascules de l'état d'une encoche

Soient $W = (w_0, w_1, \dots, w_m) \in (\Sigma^n)^{m+1}$ une partie en m coups simples joués sur le tablier de dimension $n \times 1$ et k un entier compris entre 1 et n . Dans cette sous-section, nous souhaitons démontrer qu'il existe une constante α indépendante des entiers n , m ou k telle la suite des $(m+1)$ symboles $[w_0]_k, [w_1]_k, \dots, [w_m]_k$ ne change jamais de valeur à plus de α reprises.

Nous notons $\phi = \frac{\sqrt{5}-1}{2}$ l'une des racines du polynôme $X^2 + X - 1$. On pourra utiliser sans preuve l'inégalité $\sum_{j=1}^{+\infty} \phi^j \leq 2$. Pour tout mot $w \in \Sigma^n$ de longueur n , nous introduisons le variant $V_k(w)$ par la somme

$$V_k(w) = \sum_{j=1}^n \phi^{|j-k|} [w]_j \in \mathbb{R}.$$

- ☐ 21 – Montrer qu'il existe un réel S , indépendant des entiers n et k , tel que, pour tout mot $w \in \Sigma^n$, on ait l'inégalité $V_k(w) \leq S$.
- ☐ 22 – Montrer que la suite des $(m+1)$ variants $V_k(w_0), V_k(w_1), \dots, V_k(w_m)$ est une suite de réels décroissante.
- ☐ 23 – Montrer que si, pour un indice i compris entre 0 et $m-1$, le symbole $[w_i]_k$ vaut 1 et le symbole $[w_{i+1}]_k$ vaut 0, alors on a $V_k(w_{i+1}) \leq V_k(w_i) - 1$.
- ☐ 24 – En déduire qu'il existe une constante α , indépendante des entiers n , m et k , telle que la suite $[w_0]_k, [w_1]_k, \dots, [w_m]_k$ ne change jamais de valeur à plus de α reprises.

2.3 Trace d'une partie

À partir d'une partie $W = (w_0, w_1, \dots, w_m) \in (\Sigma^n)^{m+1}$ en m coups simples joués sur le tablier de dimension $n \times 1$, nous construisons la *trace* T de la partie W en suivant la procédure suivante. Pour plus de clarté, un exemple est présenté en colonne de droite avec la partie en deux coups (w_0, w_1, w_2) où $w_0 = 110101$, $w_1 = 001101$ et $w_2 = 010001$.

ÉTAPE 1 : Nous organisons les mots de W sous forme d'un tableau de n colonnes et $m + 1$ lignes en positionnant le mot w_i dans la ligne i . Les lignes sont numérotées de bas en haut.	<table><tr><td>w_2</td><td>0</td><td>1</td><td>0</td><td>0</td><td>0</td><td>1</td></tr><tr><td>w_1</td><td>0</td><td>0</td><td>1</td><td>1</td><td>0</td><td>1</td></tr><tr><td>w_0</td><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>	w_2	0	1	0	0	0	1	w_1	0	0	1	1	0	1	w_0	1	1	0	1	0	1
w_2	0	1	0	0	0	1																
w_1	0	0	1	1	0	1																
w_0	1	1	0	1	0	1																
ÉTAPE 2 : Pour tout indice de ligne i compris entre 0 et $m - 1$, pour tout indice de colonne j compris entre 1 et n , quand les symboles $[w_i]_j$ et $[w_{i+1}]_j$ sont égaux, nous effaçons le symbole $[w_{i+1}]_j$ inscrit en position $(i + 1, j)$, de sorte qu'il ne reste que le mot w_0 dans la ligne 0 et les 3 symboles qui ont été modifiés dans les autres lignes.	<table><tr><td></td><td>1</td><td>0</td><td>0</td><td></td><td></td></tr><tr><td>0</td><td>0</td><td>1</td><td></td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>		1	0	0			0	0	1				1	1	0	1	0	1			
	1	0	0																			
0	0	1																				
1	1	0	1	0	1																	
ÉTAPE 3 : Pour tout indice i compris entre 0 et $m - 1$, lorsqu'un coup vers la gauche a été joué entre le mot w_i et le mot w_{i+1} , nous ajoutons la flèche $\leftarrow$ en indice des symboles de la ligne $i + 1$; lorsqu'un coup vers la droite a été joué, nous ajoutons la flèche $\rightarrow$ en indice.	<table><tr><td></td><td>$1_{\leftarrow}$</td><td>$0_{\leftarrow}$</td><td>$0_{\leftarrow}$</td><td></td><td></td></tr><tr><td>$0_{\rightarrow}$</td><td>$0_{\rightarrow}$</td><td>$1_{\rightarrow}$</td><td></td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>		$1_{\leftarrow}$	$0_{\leftarrow}$	$0_{\leftarrow}$			$0_{\rightarrow}$	$0_{\rightarrow}$	$1_{\rightarrow}$				1	1	0	1	0	1			
	$1_{\leftarrow}$	$0_{\leftarrow}$	$0_{\leftarrow}$																			
$0_{\rightarrow}$	$0_{\rightarrow}$	$1_{\rightarrow}$																				
1	1	0	1	0	1																	
ÉTAPE 4 : Nous numérotons dans chaque ligne différente de la ligne 0 les trois symboles restants par les marques I, II et III en exposant, en allant de gauche à droite.	<table><tr><td></td><td>$1^I_{\leftarrow}$</td><td>$0^{II}_{\leftarrow}$</td><td>$0^{III}_{\leftarrow}$</td><td></td><td></td></tr><tr><td>$0^I_{\rightarrow}$</td><td>$0^{II}_{\rightarrow}$</td><td>$1^{III}_{\rightarrow}$</td><td></td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>		$1^I_{\leftarrow}$	$0^{II}_{\leftarrow}$	$0^{III}_{\leftarrow}$			$0^I_{\rightarrow}$	$0^{II}_{\rightarrow}$	$1^{III}_{\rightarrow}$				1	1	0	1	0	1			
	$1^I_{\leftarrow}$	$0^{II}_{\leftarrow}$	$0^{III}_{\leftarrow}$																			
$0^I_{\rightarrow}$	$0^{II}_{\rightarrow}$	$1^{III}_{\rightarrow}$																				
1	1	0	1	0	1																	
ÉTAPE 5 : Nous abaissons l'ensemble des symboles, colonne par colonne, pour les regrouper dans les alvéoles les plus bas.	<table><tr><td></td><td>$1^I_{\leftarrow}$</td><td>$0^{II}_{\leftarrow}$</td><td></td><td></td><td></td></tr><tr><td>$0^I_{\rightarrow}$</td><td>$0^{II}_{\rightarrow}$</td><td>$1^{III}_{\rightarrow}$</td><td>$0^{III}_{\leftarrow}$</td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>		$1^I_{\leftarrow}$	$0^{II}_{\leftarrow}$				$0^I_{\rightarrow}$	$0^{II}_{\rightarrow}$	$1^{III}_{\rightarrow}$	$0^{III}_{\leftarrow}$			1	1	0	1	0	1			
	$1^I_{\leftarrow}$	$0^{II}_{\leftarrow}$																				
$0^I_{\rightarrow}$	$0^{II}_{\rightarrow}$	$1^{III}_{\rightarrow}$	$0^{III}_{\leftarrow}$																			
1	1	0	1	0	1																	
ÉTAPE 6 : Si x est un symbole marqué par l'exposant I ou II et est initialement issu de la ligne i et si y est son successeur dans le mot w_i , nous adjoignons à x le numéro de ligne dans lequel y se trouve après abaissement.	<table><tr><td></td><td>$(1^I_{\leftarrow}, 2)$</td><td>$(0^{II}_{\leftarrow}, 1)$</td><td></td><td></td><td></td></tr><tr><td>$(0^I_{\rightarrow}, 1)$</td><td>$(0^{II}_{\rightarrow}, 1)$</td><td>$1^{III}_{\rightarrow}$</td><td>$0^{III}_{\leftarrow}$</td><td></td><td></td></tr><tr><td>1</td><td>1</td><td>0</td><td>1</td><td>0</td><td>1</td></tr></table>		$(1^I_{\leftarrow}, 2)$	$(0^{II}_{\leftarrow}, 1)$				$(0^I_{\rightarrow}, 1)$	$(0^{II}_{\rightarrow}, 1)$	$1^{III}_{\rightarrow}$	$0^{III}_{\leftarrow}$			1	1	0	1	0	1			
	$(1^I_{\leftarrow}, 2)$	$(0^{II}_{\leftarrow}, 1)$																				
$(0^I_{\rightarrow}, 1)$	$(0^{II}_{\rightarrow}, 1)$	$1^{III}_{\rightarrow}$	$0^{III}_{\leftarrow}$																			
1	1	0	1	0	1																	

Épreuve d'option informatique MP 2021

La trace de la partie W est la suite des colonnes du tableau obtenue par la procédure ainsi décrite.

☐ 25 – Soit T la trace d'une partie. Décrire une construction qui produit une partie de trace T à partir de la trace T . (Il est suggéré de raisonner par récurrence.)

☐ 26 – Montrer que l'on peut majorer le nombre de lignes non vides dans la trace d'une partie par une constante indépendante de la dimension n du tablier ou de la longueur de la partie m . En déduire qu'une colonne de la trace d'une partie ne peut prendre qu'un nombre fini de valeurs distinctes.

Nous appelons Δ l'ensemble, fini, des valeurs colonnes que peuvent prendre les colonnes d'une trace de partie. Nous appelons $\mathcal{T}$ le langage des mots sur Δ qui représentent la trace d'une partie.

☐ 27 – Donner une caractérisation de l'ensemble des facteurs de longueur 2 de mots du langage $\mathcal{T} \subseteq \Delta^*$.

☐ 28 – Montrer qu'il existe un automate local qui reconnaît exactement le langage $\mathcal{T} \subseteq \Delta^*$.

☐ 29 – Montrer qu'il existe un automate fini qui reconnaît le langage $\mathcal{R} \subseteq \Sigma^*$ des mots de longueur quelconque de motifs résolubles.

Dans la question qui suit, le terme *complexité* désigne un ordre de grandeur asymptotique de la complexité en temps et dans le pire des cas.

☐ 30 – On s'intéresse au problème de déterminer si un motif sur le tablier unidimensionnel est résoluble ou non. Montrer que l'on peut déduire de l'automate construit à la question 29 un algorithme de complexité optimale pour résoudre ce problème.

Les résultats démontrés dans la section 2 ont été établis par B. Ravikumar pour un tablier rectangulaire de dimension $n \times n_0$ où n_0 est un entier fixé et n est un entier pouvant varier.

FIN DE L'ÉPREUVE

SOLUTION

Q1. Il s'agit du parcours en largeur. Pour cela, on utilise la structure de pile (dernier entré, premier sorti car lorsque l'on analyse un sommet, on traite juste après ses voisins qui ne sont pas encore rencontrés).

Commentaires : *Il s'agit d'une question de cours mais très mal traitée par un nombre important de candidats qui ont du mal à distinguer parcours en largeur et profondeur ainsi que la structure associée (pile ou file).*

Q2.

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

Commentaires : *Question élémentaire.*

Q3.

```
let numeros_europeens =
  let res = ref [] in
  for i=52 downto 0 do
    if numero_interieur i then res := i::(!res) done;
  !res;;
```

Commentaires : *Question élémentaire mais un nombre significatif de candidats ont du mal à gérer les références (écrivent par exemple `res = [] ... res = i : res` ou pire `i : res`, présumant que le résultat est affecté à `res`, le caractère en Ocaml n'est pas compris par ces candidats).*

Q4. Comme les voisins immédiats de (x, y) sont respectivement $(x + 1, y)$, $(x - 1, y)$, $(x, y + 1)$, $(x, y - 1)$ et l'entier z associé à (x, y) est $z = 8y + x$, on en déduit que les voisins associés à l'encoche z sont respectivement $z + 1$, $z - 1$, $z + 8$ et $z - 8$.

Commentaires : *Question élémentaire.*

Q5.

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

Commentaires : *Le but du sujet est de manipuler directement les bits sans passer par les entiers, ce que n'ont pas compris un nombre significatif de candidats. Ils ont alors utilisé les puissances. Il faut faire attention que l'exponentiation $a^{**}b$ porte sur des flottants a et b et renvoie un flottant. La question est élémentaire (si le candidat a compris les notations du sujet, notamment les fonctions agissant directement sur les bits des entiers).*

Q6.

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

Commentaires : Même remarque que précédemment.

Q7.

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

Commentaires : Question élémentaire.

Q8. Supposons que m est un motif ponctuel, il s'écrit $m = 2^k$ c'est-à-dire que son écriture binaire est $0\dots 01 \underbrace{0\dots 0}_{k \text{ chiffres}}$ alors

$$m - 1 = 2^k - 1 = 1 + 2 + 2^2 + \dots + 2^{k-1}$$

a pour écriture binaire $0.0 \underbrace{1\dots 1}_{k \text{ chiffres}}$. Par conséquent, m est non nul, m et $m - 1$ non aucun

chiffre commun valant 1 d'où $m \text{ land } m - 1$ vaut 0 (pour chaque bit, on n'a jamais la situation 1 ET 1) donc $\text{est\_ponctuel } m$ renvoie *true*.

Réciproquement, supposons que $\text{est\_ponctuel } m$ renvoie *true* c'est-à-dire m soit non nul avec $m \text{ land } m - 1$ qui vaut 0. Comme m est non nul, son écriture binaire contient au moins un 1. Considérons le 1 le plus à gauche. Si m n'admet qu'un seul 1, son écriture binaire est de la forme $10\dots 0$ donc c'est un motif ponctuel (puisque de la forme 2^k). Supposons que m possède au moins deux 1. Le premier est forcément son bit de poids fort (le chiffre non nul le plus à gauche) et considérons celui le plus à droite alors m a pour écriture binaire $1m_1\dots m_s1 \underbrace{0\dots 0}_{k-1 \text{ chiffres}}$ avec $m_i \in \{0, 1\}$. Si N est le nombre de chiffres de cette écriture,

l'écriture binaire de $m - 1$ est alors $1m_1\dots m_s0 \underbrace{1\dots 1}_{k-1 \text{ chiffres}}$ qui possède aussi N chiffres et m

$\text{land } m - 1$ n'est pas nul (puisque ils ont le même nombre de chiffres et leur premier bit vaut 1). Ceci est absurde donc m possède un seul 1 i.e. m est un motif ponctuel.

Commentaires : Question difficile qui demande au préalable la compréhension par le candidat des opérations bit à bit. Le cas direct est aisé. La réciproque demande une plus grande aisance du candidat en informatique théorique.

Q9.

```
let inclus (m: motif) (p: ponctuel) = m land p == p;;
```

Commentaires : Un candidat peut calculer toutes les puissances de 2 pour la comparer au motif m , solution lourde, juste mais refusé par le sujet car il exige que des solutions avec les opérations sur les bits. La difficulté de la question pour une solution rudimentaire est toujours la compréhension des opérations bit à bit.

Q10.

```

let voisin_g p =
  let g = p lsr 1 in if inclus motif_europeen g then g else
    0;;

let voisin_d p =
  let g = p lsl 1 in if inclus motif_europeen g then g else
    0;;

let voisin_h p =
  let g = p lsl 8 in if inclus motif_europeen g then g else
    0;;

let voisin_b p =
  let g = p lsr 8 in if inclus motif_europeen g then g else
    0;;

```

Commentaires : Question sans difficulté pour ceux qui ont compris les opérations bit à bit.

Q11.

```

let resultat_coup f m p =
  let p' = f p in
  if p' == 0 || not (inclus m p') then false, m, p else
    let p'' = f p' in
    if p'' == 0 || inclus m p'' then false, m, p else
      true, m lxor p lxor p' lxor p'', p'';;

let coup_simple (m, p) =
  let rec aux l acc = match l with
    | [] -> acc
    | f::q -> let b, m', p'' = resultat_coup f m p in if b then
      aux q ((m', p'')::acc) else aux q acc
  in aux voisins [];;

```

Commentaires : Question difficile car il y a de nombreux chausse-trappe (vérifier toutes les conditions pour que le saut soit possible, est-ce que le fichet peut arriver à la destination à priori, etc). Au final, elle est sélective.

Q12.

```

let coup_compose (m, p) =
  let rec aux l acc = match l with
    | [] -> acc
    | (m', p')::q -> let l' = coup_simple (m', p') in aux q ((m', p')::(l'@acc))

```

```
in aux [(m,p)] [];;
```

Commentaires : *Question de difficulté standard.*

Q13.

```
let mouvements m =
  let rec coup l acc = match l with
    | [] -> acc
    | i::q -> let p = numero_vers_ponctuel i in if inclus m p
      then coup q ((coup_compose (m,p))@acc) else coup q acc
  in coup numeros_europeens [];;
```

Commentaires : *Même remarque que précédemment.*

Q14.

```
let add_and_mem d s m =
  let b = mem d m in add d m s ; b;;
```

Commentaires : *Question élémentaire.*

Q15.

```
let strate d s l =
  let rec aux l = match l with
    | [] -> []
    | (m,p)::q -> if add_and_mem d s m then m::(aux q) else aux q
  in let motifs = List.flatten (List.map mouvements l) in aux motifs;;
```

Commentaires : *Question de difficulté standard. Elle est simplifiée par l'utilisation des fonctions d'ordre supérieur (flatten et map), ce qui n'est pas un attendu du programme, même si les candidats peuvent les recoder par eux-même fréquemment (et c'est normal).*

Q16.

```
let partie_minimale mi mf =
  let dic = create() in add dic mi 0 ;
  let rec aux l d =
    if mem dic mf then find dic mf else
    if l = [] then failwith "inaccessible"
    else let l' = strate dic (d+1) l in aux l' (d+1)
  in aux [mi] 0;;
```

Commentaires : Question difficile réussie par les meilleures copies.

Q17. Considérons le graphe dont les sommets sont les configurations du jeu et les arcs (s, t) sont les configurations s et t pour lesquelles on passe de s à t par un coup composé. Comme on ne mémorise jamais le moment de découverte d'une position donnée (via une pile ou une autre chose), la fonction *partie_minimale* ne respecte pas le parcours en largeur donc ne respecte la stratégie de la question 1.

Commentaires : Même remarque que précédemment (normal, celle-ci a été peu réussie !).

Q18. L'écriture binaire contient un seul 1 et un nombre arbitraire de 0 avant ou après celui-ci, l'expression 0^*10^* décrit $\mathcal{P}$.

Commentaires : Question élémentaire.

Q19.



Commentaires : Question élémentaire.

Q20. L'ensemble $P(\mathcal{P})$ (respectivement $S(\mathcal{P})$) des préfixes (respectivement suffixes) de longueur 1 de $\mathcal{P}$ est $\{0, 1\}$. Les facteurs interdits de longueur 2 forment l'ensemble $\{11\}$. Remarquons que $m = 0101$ appartient à $\mathcal{P}$, que son préfixe (resp. suffixe) de longueur 1 appartient à $P(\mathcal{P})$ (resp. $S(\mathcal{P})$), qu'aucun de ses facteurs de longueur 2 n'est interdit alors que m n'appartient pas à $\mathcal{P}$ donc $\mathcal{P}$ n'est pas un langage local.

Commentaires : Si formellement la question est au programme (langages et automates locaux). En pratique, peu de candidats se rappellent de ces notions donc sont incapables d'y répondre.

Q21. Pour tout entier j , $[w]_j$ vaut 0 ou 1 donc il est majoré par 1 ce qui nous donne la majoration

$$V_k(w) \leq \sum_{j=1}^n \phi^{|j-k|} \leq \sum_{j=1}^{+\infty} \phi^{|j-k|}.$$

(car tous les termes sont positifs). Puisque $|j - k|$ est la distance entre j et k et que j varie de 1 à $+\infty$, cette distance varie entre 0 et $+\infty$. Chaque entier étant pris au plus 2 fois par

la distance $|j - k|$ (si k est fixé et j varie dans les entiers naturels non nuls), chaque nombre ϕ^p , $p \in \mathbb{N}$, apparaît au plus deux fois. On en déduit la majoration suivante :

$$\sum_{j=1}^{+\infty} \phi^{|j-k|} \leq \sum_{p=0}^{+\infty} 2\phi^p = 2 + 2 \sum_{p=1}^{+\infty} \phi^p \leq 2 + 2 \times 2 = 6$$

donc $V_k(w)$ est majoré par 6 d'où $S = 6$ convient.

Commentaires : Question très difficile car c'est une question de mathématiques évoluée (prendre de la hauteur pour ne pas se noyer dans la technicité) dans une épreuve d'informatique ! Au final, peu de bonnes réponses.

Q22. Si w est un mot et w' le mot obtenu par un coup simple. Supposons qu'il s'agit d'un coup simple à droite donc w s'écrit $m110m'$ et w' s'écrit $m001m'$. Ainsi, tous les chiffres de w et w' sont identiques sauf ceux en position j_0 , $j_0 + 1$ et $j_0 + 2$ pour un certain entier j_0 . En position j_0 , w (resp. w') a pour chiffre 0 (resp. 1) et aux positions $j_0 + 1$ et $j_0 + 2$, w (resp. w') a pour chiffre 1 (resp. 0). Par conséquent, tous les termes $\phi^{|j-k|} [w]_j$ et $\phi^{|j-k|} [w']_j$ sont égaux sauf pour les positions $j = j_0$, $j = j_0 + 1$ et $j = j_0 + 2$. Pour w , la somme de ces termes pour $j \in \{j_0, j_0 + 1, j_0 + 2\}$ vaut $S_w = \phi^{|j_0+1-k|} + \phi^{|j_0+2-k|}$ alors que celui de w est $S_{w'} = \phi^{|j_0-k|}$ donc

$$V_k(w) - V_k(w') = S_w - S_{w'} = \phi^{|j_0+2-k|} + \phi^{|j_0+1-k|} - \phi^{|j_0-k|}.$$

Premier cas Si $k \geq j_0 + 2$, comme $|a - b|$ est l'écart entre a et b , un dessin montre que

$$|j_0 + 2 - k| = k - (j_0 + 2), |j_0 + 1 - k| = k - (j_0 + 1), |j_0 - k| = k - j_0$$

donc

$$\begin{aligned} V_k(w) - V_k(w') &= \phi^{k-(j_0+2)} + \phi^{k-(j_0+1)} - \phi^{k-j_0} \\ &= \phi^{k-(j_0+2)} \underbrace{[1 + \phi - \phi^2]}_{=1+\phi-(1-\phi)=2\phi} = 2\phi^{k-(j_0+1)} \geq 0 \end{aligned}$$

Deuxième cas Si $k = j_0 + 1$ alors

$$V_k(w) - V_k(w') = \phi + \phi^0 - \phi = 1 \geq 0$$

Troisième cas Si $k \leq j_0$ alors

$$\begin{aligned} V_k(w) - V_k(w') &= \phi^{j_0+2-k} + \phi^{j_0+1-k} - \phi^{j_0-k} \\ &= \phi^{j_0-k} \underbrace{[\phi + \phi^2 - 1]}_{=0} = 0 \geq 0 \end{aligned}$$

Au final, on voit que $V_k(w) \geq V_k(w')$ donc le variant d'un mot diminue après un coup simple à droite.

Si le coup est à gauche $011- > 100$, par un raisonnement semblable, on obtient

$$V_k(w) - V_k(w') = \phi^{|j_0-k|} + \phi^{|j_0+1-k|} - \phi^{|j_0+2-k|}.$$

On effectue encore les mêmes distinctions de cas.

Premier cas Si $k \geq j_0 + 2$ alors

$$\begin{aligned} V_k(w) - V_k(w') &= \phi^{k-j_0} + \phi^{k-(j_0+1)} - \phi^{k-(j_0+2)} \\ &= \phi^{k-(j_0+2)} \underbrace{[\phi^2 + \phi - 1]}_{=1-\phi+\phi-1=2\phi} = 0 \geq 0 \end{aligned}$$

Deuxième cas Si $k = j_0 + 1$ alors

$$V_k(w) - V_k(w') = \phi + \phi^0 - \phi = 1 \geq 0$$

Troisième cas Si $k \leq j_0$ alors

$$\begin{aligned} V_k(w) - V_k(w') &= \phi^{j_0-k} + \phi^{j_0+1-k} - \phi^{j_0+2-k} \\ &= \phi^{j_0-k} \underbrace{[1 + \phi - \phi^2]}_{=1+\phi-(1-\phi)=2\phi} = 2\phi^{j_0+1-k} \geq 0 \end{aligned}$$

Au final, on voit que $V_k(w) \geq V_k(w')$ donc le variant d'un mot diminue après un coup simple à droite.

Conclusion : le variant d'un mot diminue après un coup simple donc la suite $(V_k(w_i))_{0 \leq i \leq m}$ est décroissante.

Commentaires : Seuls les candidats ayant pensé à traiter les différents cas ont pu produire des réponses satisfaisantes. Question très sélective.

Q23. La k^e lettre de w_i est $[w_i]_k = 1$ et elle devient $[w_i]_{k+1} = 0$ donc elle fait partie de la seule séquence de longueur 3 transformée par un coup simple.

Supposons que le coup simple est à droite donc w_i s'écrit $m110m'$ et $w_{i+1} = m001m'$. Avec les notations de la réponse à la question précédente, comme k est l'indice pour lequel $[w_i]_k = 1$ et $[w_{i+1}]_k = 0$ (analogue à $[w]_j = 1$ et $[w']_j = 0$), on a $k = j_0 + 1$ ou $k = j_0 + 2$. D'après les calculs de la réponse à la question précédente, on a :

$$k = j_0 + 1 : V_k(w_i) - V_k(w_{i+1}) = 1, \quad k = j_0 + 2 : V_k(w_i) - V_k(w_{i+1}) = 2\phi$$

Supposons que le coup simple est à gauche donc w_i s'écrit $m011m'$ et $w_{i+1} = m100m'$. On a $k = j_0$ ou $k = j_0 + 1$ avec les notations de la réponse à la question précédente. D'après les calculs de la réponse à la question précédente, on a :

$$k = j_0 + 1 : V_k(w_i) - V_k(w_{i+1}) = 1, \quad k = j_0 : V_k(w_i) - V_k(w_{i+1}) = 2\phi \geq 1$$

Comme $2\phi = \sqrt{5} - 1 \simeq 2,3 - 1 \simeq 1,3$ donc $2\phi \geq 1$ d'où

$$V_k(w_i) - V_k(w_{i+1}) \geq 1 \Rightarrow V_k(w_{i+1}) - V_k(w_i) \leq -1$$

quelque soit le type de coup et les configurations traitées.

Commentaires : Question difficile. Seules les meilleures copies ont donné des réponses satisfaisantes.

Q24. Chaque changement possibles de $[w_i]_k = 1$ à $[w_{i+1}]_k = 1$ fait diminuer d'au moins 1 le nombre $V_k(w_i)$ (d'après la question précédente) donc, si s est le nombre de changements transformant 1 en 0 parmi les $[w_0]_k, [w_1]_k, \dots, [w_m]_k$ alors $V_k(w_0)$ a diminué d'au moins s . Comme c'est un nombre position, on a $s \leq V_k(w_0)$ et comme $V_k(w_0) \leq 6$ d'après la question 21, on en déduit que $s \leq 6$ d'où il y a au plus 6 changements parmi les $[w_0]_k, [w_1]_k, \dots, [w_m]_k$ changeant 1 en 0. Par le même argumentaire, le nombre de changement de 0 à 1 est majoré par 6 donc le nombre total de changements est au plus de 12.

Commentaires : Même remarque que précédemment.

Q25. On reconstruit successivement les mots w_0 puis w_1 puis w_2 , etc. La ligne 1 de la trace nous donne le mot w_0 . Supposons avoir construit le mot w_i , pour un certain entier i et qu'il se lit directement à la ligne 1. Dans la ligne 2, il existe des cellules de la forme $(0_{\rightarrow}^I, 1)$, $(0_{\leftarrow}^{II}, 1)$, 1^{III} , $(1_{\leftarrow}^I, 1)$, $(1_{\leftarrow}^{II}, 1)$, 0^{III} . On note ℓ la position de l'une de ces cellules. On remplace la ligne 1 par le mot en jouant le coup décrit en ligne 2 depuis la cellule ℓ . On supprime les éléments des cellules ℓ , $\ell + 1$ et $\ell + 2$. On abaisse ensuite toutes les cellules des colonnes ℓ , $\ell + 1$ et $\ell + 2$ (correspond à l'étape 5 depuis le mot w_i). Tout élément de la forme (a, j) des colonnes $\ell - 1$ et $\ell + 1$ sont remplacés par $(a, j - 1)$ (étape 6). On fait cela pour chaque entier ℓ , ce qui nous donne le mot w_{i+1} .

Commentaires : *Bravo aux rares copies où une réponse à peu près convenable est donnée.*

Q26. Si l'on fixe la colonne numéro k , les éléments qui restent dans cette colonne à l'issue de l'étape 2 (c'est-à-dire les cellules non vide de celle-ci) est une alternance de 0 et 1 qui correspondent aux alternances des $[w_0]_k$, $[w_1]_k$, ..., $[w_m]_k$ (après l'étape 5 qui abaissent tous les symboles dans les alvéoles les plus basses). D'après la question 24, ce nombre est majoré par un certain nombre α donc, à l'issue de l'algorithme, la colonne k a au maximum α cellules. D'après l'étape 6, toutes les cellules de la trace hormis la première, donc de la colonne k , sont de la forme (i_s^t, b) avec $i \in \{0, 1\}$ qui prend un nombre fini de valeurs, $s \in \{\leftarrow, \rightarrow\}$ prend un nombre fini de valeurs, $t \in \{I, II, III\}$ prend un nombre fini de valeurs et $b \in \{0, \dots, \alpha - 1\}$ prend un nombre fini de valeurs donc ces couples (i_s^t, b) prennent un nombre fini de valeurs. Comme la première cellule ne peut prendre que deux valeurs au maximum (0 ou 1), la colonne numéro k ne peut prendre qu'un nombre fini de valeurs distinctes.

Commentaires : *Question de combinatoire de niveau élevé donc inaccessible en pratique aux candidats en informatique ! Au final, de bien rares réponses convenables.*

Q27. Notons $a = \begin{pmatrix} a_{n-1} \\ \vdots \\ a_0 \end{pmatrix}$ et $b = \begin{pmatrix} b_{m-1} \\ \vdots \\ b_0 \end{pmatrix}$ deux lettres de Δ . On considérons i et j deux positions respectivement dans a et b en commençant par le bas de a et b c'est-à-dire on fixe $i = 1$ et $j = 1$. Tant que $i < \alpha$ et $j < \alpha$ (α le nombre défini à la question précédente).

- si $a_i = x^{III}$, on passe à la ligne suivante de a c'est-à-dire i devient $i + 1$.
- si $b_j = (x_{\rightarrow}^I, k)$, on passe à la ligne suivante de b c'est-à-dire j devient $j + 1$.
- si (a_i, b_j) existe et est de la forme $((0_{\rightarrow}^I, j), (0_{\rightarrow}^I, kj))$ ou $((1_{\leftarrow}^I, j), (0_{\leftarrow}^{II}, k))$ ou $((0_{\rightarrow}^{II}, j), 1^{III})$ ou $((0_{\leftarrow}^{II}, j), 0^{III})$, on passe aux lignes suivantes dans a et b c'est-à-dire i devient $i + 1$ et j devient $j + 1$.

Alors ab est un facteur de logueur 2 de $\mathcal{T}$ si et seulement si $i = n$ et $j = m$ à la fin de la boucle.

Commentaires : *Bravo aux candidats ayant abordé cette question et toutes mes félicitations aux éventuels candidats ayant répondu à la question (le rapport sous-entendu qu'il n'existe pas !).*

Q28. Il suffit de montrer que $\mathcal{T}$ est un langage local donc il sera reconnu par un automate local.

- L'ensemble des préfixes $\mathcal{P}(\mathcal{T})$ de longueur 1 de $\mathcal{T}$ sont les colonnes $t \in \Delta$ qui sont composés uniquement de caractères de la forme (x^I, k) (sauf la cellule du bas!).
- L'ensemble des suffixes $\mathcal{S}(\mathcal{T})$ de longueur 1 de $\mathcal{T}$ sont les colonnes $t \in \Delta$ qui sont composés uniquement de caractères de la forme x^{III} (sauf la cellule du bas!).
- les facteurs $F(\mathcal{T})$ de longueur 2 de $\mathcal{T}$ sont ceux de la question précédentes.

Si un mot $m = t_1 \cdots t_s$ avec $t_1 \in \mathcal{P}(I)$, pour tout i , $t_i t_{i+1} \in \mathcal{F}(I)$ et $t_s \in \mathcal{S}(\mathcal{T})$ alors la colonne t_1 ne contient que les coups du débuts, les facteurs $t_i t_{i+1} \in \mathcal{F}(I)$ assurent la cohérence des coups consécutifs entre eux et t_s assure que les derniers coups sont toujours des fins de coups donc m correspond à une trace de partie.

Commentaires : *Même remarque que précédemment.*

Q29 et Q30. Question de difficulté inaccessible à ce niveau vu que le sujet ne fait que 3h et 30 questions! Le rapport indique que quasiment personne n'a abordé ces questions et personne n'a donné une ébauche de réponse un tant soit peu pertinente. Il est absurde de corriger ces questions.

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES 1

Durée : 4 heures

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé d'un exercice et d'un problème dont les parties sont largement indépendantes.

EXERCICE

M. Toutlemonde habite dans un immeuble dont la porte d'entrée est sécurisée par un code à 4 chiffres dont chacun est compris entre 0 et 9. Malheureusement, il se trouve devant cette porte et il en a oublié le code.

- Q1.** Déterminer la fonction génératrice d'une variable aléatoire suivant une loi géométrique de paramètre $p \in]0,1[$ puis en déduire son espérance.
- Q2.** En essayant un code au hasard, quelle est la probabilité de tomber sur le bon code ?
- Q3.** M. Toutlemonde décide de trouver le bon code en procédant de la manière suivante : il essaye un code au hasard choisi par les codes non encore testés. On note X la variable aléatoire égale au nombre de codes testés jusqu'à obtenir le bon code. Déterminer la loi de X et donner son espérance.
- Q4.** À la place de la stratégie précédente, M. Toutlemonde essaye des codes au hasard, sans se soucier du fait qu'il les ait déjà essayés ou non. On note encore X la variable aléatoire égale au nombre de codes testés jusqu'à obtenir le bon code. Déterminer la loi de X et donner son espérance.

Q5. *Informatique Pour Tous.*

Compléter, en langage Python, le script suivant pour qu'il simule une personne essayant de deviner le code 4714 :

```
code=4714
n=int(input('Taper un code à 4 chiffres : '))
k= .....
while .....
    .....
    .....
print('Vous avez trouvé le code en '+str(k)+' essais.')
```

(..... représente une instruction ou une partie d'instruction à compléter.)

Pour ne plus oublier le code, M. Toutlemonde décide de l'écrire sur un papier qu'il garde dans sa poche. Pour ne pas se faire dérober le code il le crypte de la manière suivante : il remplace chacun des 4 chiffres par lui-même additionné de 5 et réduit modulo 10. Par exemple, le code 4714 est crypté 9269.

Q6. *Informatique Pour Tous.*

Écrire, en langage Python, une fonction `crypte(m)` qui reçoit en entrée une liste m de 4 chiffres et renvoie en sortie la version cryptée de cette liste. Par exemple, `crypte([4,7,1,4])` renvoie `[9,2,6,9]`.

PROBLÈME - Intégrales de Fresnel

Dans ce problème, on étudie certaines intégrales et séries numériques reliées aux intégrales dites de Fresnel. Augustin Fresnel (1788-1827) démontra le caractère ondulatoire de la lumière et, pour cette raison, il est considéré comme un des fondateurs de l'optique moderne.

Partie I - Intégrales fonctions de leur borne

Dans cette partie, on définit la fonction H par l'expression $H(x) = \int_0^x e^{it^2} dt$, où e^{it^2} signifie $\exp(it^2)$.

Q7. Démontrer que H est définie et de classe C^∞ sur $\mathbb{R}$. Donner une expression de $H'(x)$.

Q8. Étudier la parité de la fonction H .

Q9. Démontrer que la fonction $t \mapsto e^{it^2}$ est développable en série entière au voisinage de 0. En déduire un développement en série entière de la fonction H au voisinage de 0, en précisant l'intervalle sur lequel ce développement est valable.

Q10. Si $x > 0$, démontrer que :

$$H(x) = \frac{1}{2} \int_0^{x^2} \frac{e^{iu}}{\sqrt{u}} du.$$

Q11. Pour $x > 4\pi^2$, en déduire que :

$$H(x) - H(\sqrt{2\pi}) = -i \frac{e^{ix^2}}{2x} + \frac{i}{2\sqrt{2\pi}} - \frac{i}{4} \int_{2\pi}^{x^2} \frac{e^{iu}}{u^{\frac{3}{2}}} du.$$

Q12. En déduire que l'intégrale généralisée $\int_0^{+\infty} e^{it^2} dt$ converge.

Q13. Informatique Pour Tous.

Proposer, en langage Python, une fonction $I(f, a, b, n)$ qui prend en entrée une fonction f à valeurs réelles ou complexes, deux réels a et b et un entier naturel n et qui renvoie une valeur approchée avec la méthode des rectangles de $\int_a^b f(t) dt$ calculée avec n rectangles.

Q14. Informatique Pour Tous.

Proposer, en langage Python, une fonction $H(x, n)$ qui prend en entrée un réel x et un entier naturel n et qui renvoie une valeur approchée de $H(x)$ calculée avec la fonction de la question précédente. On rappelle que le code Python, pour e^{it^2} , est $\exp(1j*t**2)$.

Partie II - Calcul des intégrales de Fresnel

Dans cette partie, on étudie la fonction g d'expression :

$$g(x) = \int_{-\infty}^{+\infty} \frac{e^{-x^2(t^2-i)}}{t^2-i} dt.$$

Pour cela, on pose $f(x, t) = \frac{e^{-x^2(t^2-i)}}{t^2-i}$.

Q15. Si $(x, t) \in \mathbb{R}^2$, déterminer les modules des nombres complexes $e^{-x^2(t^2-i)}$ et t^2-i .

Q16. Démontrer que g est définie et continue sur $\mathbb{R}$ (on pourra utiliser un argument de parité).

Q17. Soit $(x_n)_n$ une suite divergente vers $+\infty$. À l'aide du théorème de convergence dominée, démontrer que $\lim_{n \rightarrow +\infty} g(x_n) = 0$. En déduire la limite de g en $+\infty$ et en $-\infty$.

Q18. Démontrer que g est de classe C^1 sur $\mathbb{R}^*$.

Q19. On admet dans cette question que l'intégrale $\int_{-\infty}^{+\infty} e^{-t^2} dt$ converge et est égale à $\sqrt{\pi}$.

Vérifier que :

$$\forall x > 0, \quad g'(x) = -2\sqrt{\pi} e^{-ix^2}.$$

Q20. Décomposer dans $\mathbb{C}(X)$ la fraction rationnelle $\frac{1}{X^2-i}$.

On admet ensuite que :

$$\frac{1}{X^2-i} = \frac{1-i}{4} \times \left(\frac{\sqrt{2}}{2} \times \frac{2X-\sqrt{2}}{X^2-X\sqrt{2}+1} + \frac{i}{X^2-X\sqrt{2}+1} - \frac{\sqrt{2}}{2} \times \frac{2X+\sqrt{2}}{X^2+X\sqrt{2}+1} + \frac{i}{X^2+X\sqrt{2}+1} \right).$$

Démontrer que $\int_{-\infty}^{+\infty} \frac{1}{t^2-\sqrt{2}t+1} dt = \pi\sqrt{2}$. Donner la valeur de $\int_{-\infty}^{+\infty} \frac{1}{t^2+\sqrt{2}t+1} dt$ puis déterminer la valeur de $g(0)$.

Q21. En déduire que :

$$\forall x > 0, \quad g(x) = \frac{(1+i)\pi}{\sqrt{2}} - 2\sqrt{\pi} \times H(x)$$

où la fonction H a été introduite dans la partie I.

Donner ensuite les valeurs de $\int_0^{+\infty} e^{it^2} dt$, de $\int_0^{+\infty} \cos(t^2) dt$ et de $\int_0^{+\infty} \sin(t^2) dt$.

Partie III - Étude d'une série de fonctions

Dans cette partie, on étudie la fonction S d'expression :

$$S(x) = \sum_{n=1}^{+\infty} \frac{e^{inx}}{\sqrt{n}}.$$

Pour tout entier naturel n non nul, on note f_n la fonction d'expression $f_n(x) = \frac{e^{inx}}{\sqrt{n}}$.

Q22. On suppose que $(a_n)_{n \in \mathbb{N}^*}$ est une suite réelle positive décroissante de limite nulle et que $(b_n)_{n \in \mathbb{N}}$ est une suite bornée. En admettant l'identité suivante :

$$\forall N \in \mathbb{N}^*, \quad \sum_{n=1}^N a_n (b_n - b_{n-1}) = \sum_{n=1}^N (a_n - a_{n+1}) b_n + a_{N+1} b_N - a_1 b_0,$$

démontrer que la série $\sum a_n (b_n - b_{n-1})$ converge.

Q23. Soient $x \in]0, 2\pi[$ et $n \in \mathbb{N}^*$. Démontrer que :

$$\sum_{k=1}^n e^{ikx} = e^{\frac{i(n+1)x}{2}} \times \frac{\sin\left(\frac{nx}{2}\right)}{\sin\left(\frac{x}{2}\right)}.$$

Q24. À l'aide des deux questions précédentes, démontrer que S est définie sur $]0, 2\pi[$.

Q25. On admet dans cette question que si $k \in \mathbb{N}^*$ et $x \in]0, 2\pi[$:

$$\left| \frac{e^{i(k+1)x} - e^{ikx}}{ix\sqrt{k}} - \int_k^{k+1} \frac{e^{itx}}{\sqrt{t}} dt \right| \leq \frac{1}{4k^{\frac{3}{2}}}.$$

Démontrer qu'il existe une constante $C > 0$ telle que pour tout $x \in]0, 2\pi[$:

$$\left| \frac{e^{ix} - 1}{ix} S(x) - \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt \right| \leq C.$$

Q26. Déterminer la limite, quand x tend vers 0^+ , de :

$$I(x) = \sqrt{x} \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt.$$

Q27. Déterminer la limite en 0^+ de la fonction $x \mapsto \frac{e^{ix} - 1}{ix}$. Donner alors un équivalent de $S(x)$ quand x tend vers 0^+ .

FIN

5/5

EXERCICE

M. Toutlemonde habite dans un immeuble dont la porte d'entrée est sécurisée par un code à 4 chiffres dont chacun est compris entre 0 et 9. Malheureusement, il se trouve devant cette porte et il en a oublié le code.

- Q1.** Déterminer la fonction génératrice d'une variable aléatoire suivant une loi géométrique de paramètre $p \in]0,1[$ puis en déduire son espérance.
- Q2.** En essayant un code au hasard, quelle est la probabilité de tomber sur le bon code ?
- Q3.** M. Toutlemonde décide de trouver le bon code en procédant de la manière suivante : il essaye un code au hasard choisi par les codes non encore testés. On note X la variable aléatoire égale au nombre de codes testés jusqu'à obtenir le bon code. Déterminer la loi de X et donner son espérance.
- Q4.** À la place de la stratégie précédente, M. Toutlemonde essaye des codes au hasard, sans se soucier du fait qu'il les ait déjà essayés ou non. On note encore X la variable aléatoire égale au nombre de codes testés jusqu'à obtenir le bon code. Déterminer la loi de X et donner son espérance.

Q5. *Informatique Pour Tous.*

Compléter, en langage Python, le script suivant pour qu'il simule une personne essayant de deviner le code 4714 :

```
code=4714
n=int(input('Taper un code à 4 chiffres : '))
k= .....
while .....
    .....
    .....
print('Vous avez trouvé le code en '+str(k)+' essais.')
```

(..... représente une instruction ou une partie d'instruction à compléter.)

Pour ne plus oublier le code, M. Toutlemonde décide de l'écrire sur un papier qu'il garde dans sa poche. Pour ne pas se faire dérober le code il le crypte de la manière suivante : il remplace chacun des 4 chiffres par lui-même additionné de 5 et réduit modulo 10. Par exemple, le code 4714 est crypté 9269.

Q6. *Informatique Pour Tous.*

Écrire, en langage Python, une fonction `crypte(m)` qui reçoit en entrée une liste m de 4 chiffres et renvoie en sortie la version cryptée de cette liste. Par exemple, `crypte([4,7,1,4])` renvoie `[9,2,6,9]`.

PROBLÈME - Intégrales de Fresnel

Dans ce problème, on étudie certaines intégrales et séries numériques reliées aux intégrales dites de Fresnel. Augustin Fresnel (1788-1827) démontra le caractère ondulatoire de la lumière et, pour cette raison, il est considéré comme un des fondateurs de l'optique moderne.

Partie I - Intégrales fonctions de leur borne

Dans cette partie, on définit la fonction H par l'expression $H(x) = \int_0^x e^{it^2} dt$, où e^{it^2} signifie $\exp(it^2)$.

Q7. Démontrer que H est définie et de classe C^∞ sur $\mathbb{R}$. Donner une expression de $H'(x)$.

Q8. Étudier la parité de la fonction H .

Q9. Démontrer que la fonction $t \mapsto e^{it^2}$ est développable en série entière au voisinage de 0. En déduire un développement en série entière de la fonction H au voisinage de 0, en précisant l'intervalle sur lequel ce développement est valable.

Q10. Si $x > 0$, démontrer que :

$$H(x) = \frac{1}{2} \int_0^{x^2} \frac{e^{iu}}{\sqrt{u}} du.$$

Q11. Pour $x > 4\pi^2$, en déduire que :

$$H(x) - H(\sqrt{2\pi}) = -i \frac{e^{ix^2}}{2x} + \frac{i}{2\sqrt{2\pi}} - \frac{i}{4} \int_{2\pi}^{x^2} \frac{e^{iu}}{u^{\frac{3}{2}}} du.$$

Q12. En déduire que l'intégrale généralisée $\int_0^{+\infty} e^{it^2} dt$ converge.

Q13. Informatique Pour Tous.

Proposer, en langage Python, une fonction $I(f, a, b, n)$ qui prend en entrée une fonction f à valeurs réelles ou complexes, deux réels a et b et un entier naturel n et qui renvoie une valeur approchée avec la méthode des rectangles de $\int_a^b f(t) dt$ calculée avec n rectangles.

Q14. Informatique Pour Tous.

Proposer, en langage Python, une fonction $H(x, n)$ qui prend en entrée un réel x et un entier naturel n et qui renvoie une valeur approchée de $H(x)$ calculée avec la fonction de la question précédente. On rappelle que le code Python, pour e^{it^2} , est `exp(1j*t**2)`.

Partie II - Calcul des intégrales de Fresnel

Dans cette partie, on étudie la fonction g d'expression :

$$g(x) = \int_{-\infty}^{+\infty} \frac{e^{-x^2(t^2-i)}}{t^2-i} dt.$$

Pour cela, on pose $f(x, t) = \frac{e^{-x^2(t^2-i)}}{t^2-i}$.

Q15. Si $(x, t) \in \mathbb{R}^2$, déterminer les modules des nombres complexes $e^{-x^2(t^2-i)}$ et t^2-i .

Q16. Démontrer que g est définie et continue sur $\mathbb{R}$ (on pourra utiliser un argument de parité).

Q17. Soit $(x_n)_n$ une suite divergente vers $+\infty$. À l'aide du théorème de convergence dominée, démontrer que $\lim_{n \rightarrow +\infty} g(x_n) = 0$. En déduire la limite de g en $+\infty$ et en $-\infty$.

Q18. Démontrer que g est de classe C^1 sur $\mathbb{R}^*$.

Q19. On admet dans cette question que l'intégrale $\int_{-\infty}^{+\infty} e^{-t^2} dt$ converge et est égale à $\sqrt{\pi}$.

Vérifier que :

$$\forall x > 0, \quad g'(x) = -2\sqrt{\pi} e^{-x^2}.$$

Q20. Décomposer dans $\mathbb{C}(X)$ la fraction rationnelle $\frac{1}{X^2-i}$.

On admet ensuite que :

$$\frac{1}{X^2-i} = \frac{1-i}{4} \times \left(\frac{\sqrt{2}}{2} \times \frac{2X-\sqrt{2}}{X^2-X\sqrt{2}+1} + \frac{i}{X^2-X\sqrt{2}+1} - \frac{\sqrt{2}}{2} \times \frac{2X+\sqrt{2}}{X^2+X\sqrt{2}+1} + \frac{i}{X^2+X\sqrt{2}+1} \right).$$

Démontrer que $\int_{-\infty}^{+\infty} \frac{1}{t^2-\sqrt{2}t+1} dt = \pi\sqrt{2}$. Donner la valeur de $\int_{-\infty}^{+\infty} \frac{1}{t^2+\sqrt{2}t+1} dt$ puis déterminer la valeur de $g(0)$.

Q21. En déduire que :

$$\forall x > 0, \quad g(x) = \frac{(1+i)\pi}{\sqrt{2}} - 2\sqrt{\pi} \times H(x)$$

où la fonction H a été introduite dans la partie I.

Donner ensuite les valeurs de $\int_0^{+\infty} e^{-t^2} dt$, de $\int_0^{+\infty} \cos(t^2) dt$ et de $\int_0^{+\infty} \sin(t^2) dt$.

Partie III - Étude d'une série de fonctions

Dans cette partie, on étudie la fonction S d'expression :

$$S(x) = \sum_{n=1}^{+\infty} \frac{e^{inx}}{\sqrt{n}}.$$

Pour tout entier naturel n non nul, on note f_n la fonction d'expression $f_n(x) = \frac{e^{inx}}{\sqrt{n}}$.

Q22. On suppose que $(a_n)_{n \in \mathbb{N}^*}$ est une suite réelle positive décroissante de limite nulle et que $(b_n)_{n \in \mathbb{N}}$ est une suite bornée. En admettant l'identité suivante :

$$\forall N \in \mathbb{N}^*, \quad \sum_{n=1}^N a_n (b_n - b_{n-1}) = \sum_{n=1}^N (a_n - a_{n+1}) b_n + a_{N+1} b_N - a_1 b_0,$$

démontrer que la série $\sum a_n (b_n - b_{n-1})$ converge.

Q23. Soient $x \in]0, 2\pi[$ et $n \in \mathbb{N}^*$. Démontrer que :

$$\sum_{k=1}^n e^{ikx} = e^{\frac{i(n+1)x}{2}} \times \frac{\sin\left(\frac{nx}{2}\right)}{\sin\left(\frac{x}{2}\right)}.$$

Q24. À l'aide des deux questions précédentes, démontrer que S est définie sur $]0, 2\pi[$.

Q25. On admet dans cette question que si $k \in \mathbb{N}^*$ et $x \in]0, 2\pi[$:

$$\left| \frac{e^{i(k+1)x} - e^{ikx}}{ix\sqrt{k}} - \int_k^{k+1} \frac{e^{itx}}{\sqrt{t}} dt \right| \leq \frac{1}{4k^{\frac{3}{2}}}.$$

Démontrer qu'il existe une constante $C > 0$ telle que pour tout $x \in]0, 2\pi[$:

$$\left| \frac{e^{ix} - 1}{ix} S(x) - \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt \right| \leq C.$$

Q26. Déterminer la limite, quand x tend vers 0^+ , de :

$$I(x) = \sqrt{x} \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt.$$

Q27. Déterminer la limite en 0^+ de la fonction $x \mapsto \frac{e^{ix} - 1}{ix}$. Donner alors un équivalent de $S(x)$ quand x tend vers 0^+ .

FIN

5/5

SOLUTION

Q1. Soit X une variable aléatoire suivant la loi géométrique de paramètre p alors, pour tout entier $n \geq 1$, on a $P(X = n) = q^{n-1}p$ avec $q = 1 - p$. Par conséquent, sa série génératrice G_X vaut

$$G_X(t) = \sum_{n=1}^{+\infty} P(X = n) t^n = pt \sum_{n=1}^{+\infty} (qt)^{n-1} = pt \sum_{k=0}^{+\infty} (qt)^k = \frac{pt}{1 - qt}.$$

Cette fonction étant définie et dérivable en $t = 1$, X admet une espérance définie par

$$\begin{aligned} E(X) &= G'_X(1) = \frac{d}{dt} \left(\frac{pt}{1 - qt} \right) \Big|_{t=1} = \left(\frac{p(1 - qt) - pt(-q)}{(1 - qt)^2} \right) \Big|_{t=1} \\ &= \frac{p}{(1 - q)^2} = \frac{p}{p^2} = \frac{1}{p}. \end{aligned}$$

Commentaires : *Il s'agit d'une question de cours. Contre toute attente, elle fut bien mal traitée par une majorité de candidats ! Pensez à consolider votre cours (définition, théorème et question de cours en colle). C'est dommage de ne pas récupérer les points des questions simples et classiques !*

Q2. Chaque chiffre ayant 10 choix et comme il y a 4 chiffres, il a 10^4 possibilités de codes. Comme ils sont choisis selon la loi uniforme, la probabilité d'obtenir un code donné est $p = \frac{1}{10^4} = 10^{-4}$.

Commentaires : *Même remarque que précédemment !*

Q3. On pose $N = 10^4$. Comme il ne reteste jamais un code déjà testé, les valeurs possibles de X sont tous les entiers entre 0 et $N - 1$. Pour tout entier $k \in \{0, \dots, N - 1\}$, l'évènement $(X = k)$ est réalisé s'il choisit le mauvais code au premier essai (probabilité $\frac{N-1}{N}$), le mauvais au deuxième (probabilité $\frac{N-2}{N-1}$ car le code précédent ne sera pas choisi), le mauvais au troisième (probabilité $\frac{N-3}{N-2}$ car les deux premiers codes ne sont pas tentés), etc jusqu'au $k - 1$ ième (probabilité $\frac{N-k}{N-k+1}$) et il réussit au k ième (probabilité $\frac{1}{N-k}$). On en déduit que

$$P(X = k) = \frac{N-1}{N} \times \frac{N-2}{N-1} \times \frac{N-3}{N-2} \times \dots \times \frac{N-k}{N-k+1} \times \frac{1}{N-k} = \frac{1}{N}$$

donc X suit la loi uniforme sur $\{0, \dots, N - 1\}$. Son espérance vaut alors

$$E(X) = \sum_{k=0}^{N-1} kP(X = k) = \frac{1}{N} \sum_{k=0}^{N-1} k = \frac{1}{N} \times \frac{N(N-1)}{2} = \frac{N-1}{2} = 4999.5$$

Commentaires : *Question de niveau standard.*

Q4. X suit la loi géométrique de paramètre $p = \frac{1}{10^4}$ (nombres d'expériences identiques, mutuellement indépendantes, ayant 2 issues : obtenir le bon code ou pas). Son espérance est $\frac{1}{p} = 10^4$.

Commentaires : Même remarque qu'aux questions 1 et 2 ! C'est étonnant que le cours fut si mal traité à cette épreuve (qui représente presque 15 % du sujet !).

Q5.

```
code=4714
n=int(input("Taper un code à 4 chiffres : "))
k= 0
while n!=code:
    k=k+1
    n=int(input("Taper un code à 4 chiffres : "))
print("Vous avez trouvé le code en "+str(k)+" essais.")
```

Commentaires : Question élémentaire.

Q6.

```
def crypte(m):
    return [(c+5)%10 for c in m]

def crypte(m):
    temp=[]
    for i in range(len(m)):
        temp.append((m[i]+5)%10)
    return temp
```

Commentaires : Question élémentaire.

Q7. La fonction $h : t \mapsto e^{it^2}$ est continue sur $\mathbb{R}$ donc H est sa primitive (celle s'annulant en 0) donc elle est dérivable et sa dérivée est h . Comme $H' = h$ est de classe C^∞ sur $\mathbb{R}$, on en déduit que H est également de classe C^∞ sur $\mathbb{R}$.

Commentaires : Il s'agit d'un résultat classique du cours d'analyse de première et deuxième année. Vous devez le connaître à tout moment tant à l'écrit qu'à l'oral de tous les concours CCINP, Centrale et Mines car il est d'utilisation très fréquente. A retravailler si vous ne l'avez pas réussi.

Q8. A l'aide du changement de variable $u = -t$ ($dt = -du$, $t = 0$ alors $u = 0$, $t = x$ alors $t = -x$), on peut écrire pour tout réel x :

$$H(-x) = \int_0^{-x} e^{it^2} dt = \int_0^x e^{i(-u)^2} (-du) = -\int_0^x e^{iu^2} du = -H(x)$$

donc H est impaire.

Commentaires : Question élémentaire.

Q9. Pour tout complexe z , on a $e^z = \sum_{n=0}^{+\infty} \frac{z^n}{n!}$ donc, pour tout réel t , on a :

$$e^{it^2} = \sum_{n=0}^{+\infty} \frac{(it^2)^n}{n!} = \sum_{n=0}^{+\infty} \frac{i^n t^{2n}}{n!}$$

donc $t \mapsto e^{it^2}$ est développable en série entière sur $\mathbb{R}$. En particulier, son rayon de convergence vaut $R = +\infty$. Les développements en série entière étant stable par primitivation (et on peut les primitiver terme à terme), on peut écrire pour tout réel x :

$$H(x) = \sum_{n=0}^{+\infty} \frac{i^n x^{2n+1}}{n! (2n+1)}.$$

Commentaires : Il s'agit de deux questions de cours. Si vous avez eu des difficultés sur celles-ci, retravailler les fondamentaux du cours de séries entières.

Q10. On effectue le changement de variable $u = t^2$ donc $t = \sqrt{u}$. Quand $t = 0$ alors $u = 0$, quand $t = x$ alors $u = x^2$ et on a

$$dt = (\sqrt{u})' du = \frac{1}{2\sqrt{u}} du.$$

On peut alors écrire

$$H(x) = \int_0^x e^{it^2} dt = \int_0^{x^2} e^{iu} \frac{du}{2\sqrt{u}}.$$

Commentaires : Question de niveau standard pour ce concours.

Q11. On commence par remarquer que

$$H(\sqrt{2\pi}) = \frac{1}{2} \int_0^{2\pi} \frac{e^{iu}}{\sqrt{u}} du \Rightarrow H(x) - H(\sqrt{2\pi}) = \frac{1}{2} \int_{2\pi}^{x^2} \frac{e^{iu}}{\sqrt{u}} du$$

par la relation de Chasles. On procède par une intégration par parties en intégrant e^{iu} (primitive $\frac{e^{iu}}{i}$) et en dérivant $\frac{1}{\sqrt{u}} = u^{-1/2}$ (dérivée $-\frac{1}{2}u^{-1/2-1} = -\frac{1}{2u^{3/2}}$).

$$\begin{aligned} \int_{2\pi}^{x^2} \frac{e^{iu}}{\sqrt{u}} du &= \left[\frac{e^{iu}}{i} \times \frac{1}{\sqrt{u}} \right]_{2\pi}^{x^2} - \int_{2\pi}^{x^2} \frac{e^{iu}}{i} \left(-\frac{1}{2u^{3/2}} \right) du \\ &= \frac{e^{ix^2}}{ix} - \frac{1}{i\sqrt{2\pi}} + \frac{1}{2i} \int_{2\pi}^{x^2} \frac{e^{iu}}{u^{3/2}} du \end{aligned}$$

(car $e^{2\pi i} = 1$ et $\sqrt{x^2} = x$ puisque x est positif). On conclut en multipliant cette égalité par $\frac{1}{2}$.

Commentaires : *Même remarque que précédemment.*

Q12. La fonction $g : u \mapsto \frac{e^{iu}}{u^{3/2}}$ est continue sur $[2\pi, +\infty[$ et $|g(u)| = \frac{1}{u^{3/2}}$ est intégrable sur cet intervalle (intégrale de Riemann de paramètre $\frac{3}{2} > 1$). En particulier, l'intégrale $\int_{2\pi}^{+\infty} g$ converge c'est-à-dire que $\lim_{x \rightarrow +\infty} \int_{2\pi}^x g$ existe dans $\mathbb{C}$. Comme $\left| \frac{e^{ix^2}}{ix} \right| = \frac{1}{x}$ tend vers 0 quand $x \rightarrow +\infty$, on en déduit que

$$\lim_{x \rightarrow +\infty} \left(\frac{e^{ix^2}}{ix} - \frac{1}{i\sqrt{2\pi}} + \frac{1}{2i} \int_{2\pi}^{x^2} \frac{e^{iu}}{u^{3/2}} du \right) = \lim_{x \rightarrow +\infty} \int_{2\pi}^{x^2} \frac{e^{iu}}{\sqrt{u}} du$$

existe dans $\mathbb{C}$. Par conséquent, $\lim_{x \rightarrow +\infty} H(x) = \lim_{x \rightarrow +\infty} \int_0^x e^{it^2} dt$ existe dans $\mathbb{C}$ c'est-à-dire que

l'intégrale $\int_0^{+\infty} e^{it^2} dt$ converge.

Commentaires : *Question discriminante car elle demande de la part du candidat de connaître à la fois ses définitions et ses théorèmes du chapitre d'intégration ... tout en passant aisément de l'un à l'autre. Si vous avez eu des difficultés à cette question, n'hésitez pas à la retravailler, elle est standard à ce concours (et à d'autres).*

Q13.

```
def I(f,a,b,n):
    pas = (b-a)/n
    S=0
    point=a
    for k in range(n+1):
        S=S+f(point)
        point=point+pas
    return S*pas

def I(f,a,b,n):
    S=0
    for k in range(n+1):
        S=S+f(a+k*(b-a)/n)
    return S*(b-a)/n
```

Commentaires : *Question de difficulté standard.*

Q14.

```
import numpy as np
def H(x,n):
    def g(u):
        return np.exp(1j*u**2)
    return I(g,0,x,n)
```

Commentaires : Question de difficulté standard.

Q15. D'après les règles de calculs sur l'exponentielle et les modules, on a :

$$\left| e^{-x^2(t^2-i)} \right| = \left| e^{-x^2t^2+ix^2} \right| = \left| e^{-x^2t^2} \right| \left| e^{ix^2} \right| = e^{-x^2t^2}$$

car $e^{-x^2t^2}$ est un réel positif et ix^2 est un imaginaire pur. Pour finir, par définition du module d'un complexe écrit sous forme algébrique, on a :

$$|t^2 - i| = \sqrt{(t^2)^2 + 1} = \sqrt{t^4 + 1}$$

Commentaires : Question de cours. Si elle vous avez posé problème, retravailler votre chapitre nombres complexes de MPSI, notamment la partie exponentielle d'un complexe (utilisé assez fréquemment aux écrits de concours).

Q16. Soit $x \in \mathbb{R}$. La fonction $t \mapsto f(x, t)$ est continue sur $\mathbb{R}$ et

$$|f(x, t)| = \frac{e^{-x^2t^2}}{\sqrt{t^4 + 1}} \leq \frac{1}{\sqrt{t^4 + 1}} = \varphi(t).$$

La fonction φ est continue sur $[0, +\infty[$ donc elle est intégrable sur le segment $[0, 1]$. On dispose de l'équivalent suivant $\varphi(t) \underset{t \rightarrow +\infty}{\sim} \frac{1}{\sqrt{t^4}} = \frac{1}{t^2}$. La fonction $t \mapsto \frac{1}{t^2}$ est intégrable sur $[1, +\infty[$ (intégrale de Riemann de paramètre $2 > 1$) donc φ l'est aussi. Ainsi, φ est intégrable sur $\mathbb{R}_+$ donc $t \mapsto f(x, t)$ l'est aussi d'où, par parité de cette fonction, elle est intégrable sur $\mathbb{R}$.

Commentaires : Question de niveau standard. En pratique, les candidats gèrent assez mal les modules de nombres complexes (d'où la question précédente) et justifient assez mal l'intégrabilité de la fonction de domination φ . A retravailler si cela fut votre cas (question traditionnel à ce concours ... et aux autres).

Q17. Pour tout entier n , on pose $f_n(t) = f(x_n, t)$ qui est une fonction continue (par rapport à t). Pour chaque $t > 0$, $\lim_{n \rightarrow +\infty} f(x_n, t) = 0$ (car $|f_n(t)| = \frac{e^{-x_n^2t^2}}{\sqrt{t^4 + 1}} \xrightarrow{n \rightarrow +\infty} 0$ puisque $t^2 > 0$) et $f(x_n, 0) = 1 \xrightarrow{n \rightarrow +\infty} 1$. On dispose de la domination suivante :

$$\forall n \in \mathbb{N}, \forall t \geq 0, |f_n(t)| \leq \frac{1}{\sqrt{t^4 + 1}} = \varphi(t).$$

La fonction φ est intégrable (cf. question précédente). donc, d'après le théorème de convergence dominée, on peut affirmer que

$$\lim_{n \rightarrow +\infty} g(x_n) = \int_0^{+\infty} \lim_{n \rightarrow +\infty} f_n(t) dt = 0.$$

Puisque, pour toute suite $(x_n)_n$ tendant vers $+\infty$, on a $\lim_{n \rightarrow +\infty} g(x_n) = 0$, on peut affirmer que $\lim_{x \rightarrow +\infty} g(x) = 0$. Par parité de g , on a $\lim_{x \rightarrow -\infty} g(x) = 0$.

Commentaires : *Même remarque que précédemment.*

Q18. Pour tout $t \in \mathbb{R}$, la fonction $x \mapsto f(x, t)$ est de classe C^1 sur $\mathbb{R}$. Pour chaque $x \in \mathbb{R}$, les fonctions $t \mapsto f(x, t)$ et

$$t \mapsto \frac{\partial f}{\partial x}(x, t) = \frac{-2x(t^2 - i)e^{-x^2(t^2 - i)}}{t^2 - i} = -2xe^{-x^2(t^2 - i)}$$

sont continues sur $\mathbb{R}$. Pour chaque $x \in \mathbb{R}$, la fonction $t \mapsto f(x, t)$ est intégrable sur $\mathbb{R}$ (question 16). Vérifions la domination de la dérivée $\frac{\partial f}{\partial x}$. Pour tout segment $[a, b]$ de $\mathbb{R}_+$ (i.e. $a > 0$), on a :

$$\forall t \in \mathbb{R}, \forall x \in [-a, a], \left| \frac{\partial f}{\partial x}(x, t) \right| = 2|x|e^{-x^2 t^2} \leq 2be^{-a^2 t^2} = \psi_{a,b}(t).$$

La fonction $\psi_{a,b}$ est continue sur $\mathbb{R}$ donc elle est intégrable sur le segment $[0, 1]$. D'après les croissances comparées, on a la domination $\psi_{a,b}(t) \underset{t \rightarrow +\infty}{=} o\left(\frac{1}{t^2}\right)$ et la fonction $t \mapsto \frac{1}{t^2}$ est intégrable sur $[1, +\infty[$ (intégrale de Riemann de paramètre $2 > 1$). Par conséquent, la fonction $\psi_{a,b}$ est intégrable sur cet intervalle donc sur $\mathbb{R}_+$ et, par parité, sur $\mathbb{R}$. Ainsi, on

peut affirmer que la fonction $x \mapsto \int_{-\infty}^{+\infty} f(x, t) dt = g(x)$ est de classe C^1 sur $\mathbb{R}_+^*$ et, par parité de cette fonction, elle est C^1 sur $\mathbb{R}^*$.

Commentaires : *Même remarque que précédemment (même si la question est mieux réussie que les deux précédentes car les dominations justes sont plus aisées à obtenir).*

Q19. D'après la question précédente, on a pour tout réel strictement x :

$$\begin{aligned} g'(x) &= \int_{-\infty}^{+\infty} \frac{\partial f}{\partial x}(x, t) dt = -2x \int_{-\infty}^{+\infty} e^{-x^2(t^2 - i)} dt = -2x \int_{-\infty}^{+\infty} e^{-x^2 t^2} e^{x^2 i} dt \\ &= -2e^{ix^2} \int_{-\infty}^{+\infty} e^{-x^2 t^2} x dt \stackrel{\substack{u=xt \\ du=xdx}}{=} -2e^{ix^2} \int_{-\infty}^{+\infty} e^{-u^2} du = -2e^{ix^2} \sqrt{\pi} \end{aligned}$$

Commentaires : *Question de niveau standard. On évite de forcer le résultat car, comme il est donné, les correcteurs sont attentifs et exigeants sur les détails.*

Q20. Comme $i = e^{i\pi/2} = (e^{i\pi/4})^2$, les solutions de $z^2 = i$ sont $z = \pm e^{i\pi/4}$. La fraction rationnelle $F = \frac{1}{X^2 - i}$ se décompose alors en éléments simples sous la forme

$$F = \frac{1}{(X - e^{i\pi/4})(X + e^{i\pi/4})} = \frac{\alpha}{X - e^{i\pi/4}} + \frac{\beta}{X + e^{i\pi/4}}$$

avec α et β deux complexes. En multipliant cette égalité par $X - e^{i\pi/4}$ (resp. $X + e^{i\pi/4}$), en simplifiant puis en faisant tendre X vers $e^{i\pi/4}$ (resp. $-e^{i\pi/4}$), on obtient :

$$\alpha = \frac{1}{2e^{i\pi/4}}, \beta = -\frac{1}{2e^{i\pi/4}}$$

donc

$$\frac{1}{X^2 - i} = \frac{1}{2e^{i\pi/4}} \left(\frac{1}{X - e^{i\pi/4}} - \frac{1}{X + e^{i\pi/4}} \right).$$

Pour déterminer la valeur de l'intégrale proposée, on utilise un début de carré

$$t^2 - \sqrt{2}t + 1 = \left(t - \frac{\sqrt{2}}{2}\right)^2 + 1 - \left(\frac{\sqrt{2}}{2}\right)^2 = \left(t - \frac{\sqrt{2}}{2}\right)^2 + \frac{1}{2}.$$

On peut alors écrire

$$\begin{aligned} \frac{1}{t^2 - \sqrt{2}t + 1} &= \frac{1}{\left(t - \frac{\sqrt{2}}{2}\right)^2 + \frac{1}{2}} = \frac{1}{\frac{1}{2} \left(2 \left(t - \frac{\sqrt{2}}{2}\right)^2 + 1\right)} \\ &= \frac{2}{(\sqrt{2}t - 1)^2 + 1} = \frac{2}{\sqrt{2}} \left(\arctan(\sqrt{2}t - 1) \right)' \end{aligned}$$

ce qui nous donne la valeur de l'intégrale recherchée

$$\begin{aligned} \int_{-\infty}^{+\infty} \frac{dt}{t^2 - \sqrt{2}t + 1} &= \left[\sqrt{2} \arctan(\sqrt{2}t - 1) \right]_{t \rightarrow -\infty}^{t \rightarrow +\infty} \\ &= \sqrt{2} \left(\frac{\pi}{2} - \left(-\frac{\pi}{2}\right) \right) = \pi\sqrt{2}. \end{aligned}$$

En faisant le changement de variable $u = -t$, on obtient que

$$\int_{-\infty}^{+\infty} \frac{dt}{t^2 + \sqrt{2}t + 1} = \int_{+\infty}^{-\infty} \frac{-du}{u^2 - \sqrt{2}u + 1} = \int_{-\infty}^{+\infty} \frac{du}{u^2 - \sqrt{2}u + 1} = \pi\sqrt{2}.$$

En utilisant la décomposition en éléments simples de $\frac{1}{X^2 - i}$ sur $\mathbb{R}$ admise par le sujet, en remarquant que

$$\frac{2X \pm \sqrt{2}}{X^2 \pm \sqrt{2}X + 1} = \left(\ln(X^2 \pm \sqrt{2}X + 1) \right)'$$

donc que

$$\begin{aligned} &\frac{2X + \sqrt{2}}{X^2 + \sqrt{2}X + 1} - \frac{2X - \sqrt{2}}{X^2 - \sqrt{2}X + 1} \\ &= \left(\ln(X^2 + \sqrt{2}X + 1) - \ln(X^2 - \sqrt{2}X + 1) \right)' \\ &= \left(\ln \left(\frac{X^2 + \sqrt{2}X + 1}{X^2 - \sqrt{2}X + 1} \right) \right)'. \end{aligned}$$

Comme la fonction $\ln \left(\frac{t^2 + \sqrt{2}t + 1}{t^2 - \sqrt{2}t + 1} \right)$ tend vers $\ln(1) = 0$ quand $t \rightarrow \pm\infty$, on a

$$\begin{aligned} g(0) &= \int_{-\infty}^{+\infty} \frac{dt}{t^2 - i} = \frac{1-i}{4} \times \frac{\sqrt{2}}{2} \left[\ln \left(\frac{t^2 + \sqrt{2}t + 1}{t^2 - \sqrt{2}t + 1} \right) \right]_{t \rightarrow -\infty}^{t \rightarrow +\infty} \\ &\quad + \frac{1-i}{4} \times i \left(\int_{-\infty}^{+\infty} \frac{dt}{t^2 + \sqrt{2}t + 1} + \int_{-\infty}^{+\infty} \frac{dt}{t^2 - \sqrt{2}t + 1} \right) \\ &= \frac{1-i}{4} \times i \left(\pi\sqrt{2} + \pi\sqrt{2} \right) = \frac{1+i}{2} \pi\sqrt{2} = \frac{\pi(1+i)}{\sqrt{2}}. \end{aligned}$$

Commentaires : Si la décomposition en éléments simples est un classique du cours, la détermination des racines carrés d'un complexe s'avère en pratique sélective à ce concours. Cela fut encore le cas dans cette épreuve et cette question fut réussie uniquement dans les très bonnes copies.

Les deux autres questions sont des standards du programme et ne doivent pas poser de grands problèmes. Au final, ces deux calculs sont très discriminants (mais accessibles).

Q21. A la question 19, nous avons prouvé que

$$\forall x \in \mathbb{R}^*, g'(x) = -2e^{ix^2} \sqrt{\pi}.$$

Ainsi, il existe une constante C_+ telle que

$$\forall x > 0, g(x) = C_+ - 2\sqrt{\pi} \int_0^x e^{it^2} dt = C_+ - 2\sqrt{\pi} H(x).$$

En faisant tendre x vers 0, par continuité de g et de H , on obtient

$$C_+ = g(0) = \frac{1+i}{\sqrt{2}} \pi \Rightarrow \forall x > 0, g(x) = \frac{1+i}{\sqrt{2}} \pi - 2\sqrt{\pi} H(x).$$

Cette relation est manifestement vraie pour $x = 0$ puis sur $\mathbb{R}_+$ (par le même raisonnement, il y a une autre constante C_- puis on fait tendre x vers 0).

Comme g tend vers 0 en $\pm\infty$ (cf. question 17), en faisant tendre x vers $+\infty$ dans l'égalité de la question précédente, on obtient

$$2\sqrt{\pi} \int_0^{+\infty} e^{it^2} dt = \frac{1+i}{\sqrt{2}} \pi \Leftrightarrow \int_0^{+\infty} e^{it^2} dt = \frac{1+i}{2} \sqrt{\pi}.$$

En prenant les parties réelles et imaginaires de cette égalité, on aboutit aux formules :

$$\int_0^{+\infty} \cos(t^2) dt = \frac{\sqrt{\pi}}{2} = \int_0^{+\infty} \sin(t^2) dt.$$

Commentaires : Question destinée aux très bonnes copies.

Q22. Comme la suite $(b_n)_n$ est bornée, il existe un réel M tel que

$$\forall n \in \mathbb{N}, |b_n| \leq M.$$

Par conséquent, pour tout entier n , on a la majoration suivante :

$$|(a_n - a_{n+1}) b_n| = |a_n - a_{n+1}| |b_n| \leq |a_n - a_{n+1}| M = (a_n - a_{n+1}) M$$

(car la suite $(a_n)_n$ est décroissante donc $a_n - a_{n+1}$ est toujours positif). La suite $(a_n)_n$ étant convergente, la série $\sum_n (a_n - a_{n+1})$ converge donc, par théorème de comparaison, la série $\sum_n |(a_n - a_{n+1}) b_n|$ converge. Par suite, la série $\sum_n (a_n - a_{n+1}) b_n$ converge c'est-à-dire que la suite $\left(\sum_{n=1}^N (a_n - a_{n+1}) b_n \right)_N$ converge. La suite $(a_n)_n$ tendant vers 0 et la suite $(b_n)_n$ étant bornée, on peut affirmer que la suite $(a_{n+1} b_n)_n$ converge d'où la convergence de la suite

$$\sum_{n=1}^N (a_n - a_{n+1}) b_n + a_{N+1} b_N - a_1 b_0 = \sum_{n=1}^N a_n (b_n - b_{n-1}).$$

Autrement dit, la série $\sum_n a_n (b_n - b_{n-1})$ converge.

Commentaires : Question de niveau standard. Une connaissance solide du chapitre séries numériques est indispensable pour répondre correctement à cette question : étude de la convergence absolue (les termes sont des nombres réels arbitraire), les théorèmes de comparaison (on compare les termes généraux et on n'écrit pas des inégalités de la forme $\sum_n u_n \leq \sum_n v_n$ qui prouve les méconnaissances et/ou incompréhension du cas, etc). Au final, la question est très discriminante. A travailler si elle vous a posé problème car elle intervient relativement à ce concours (et aux autres) et elle vous oblige à consolider votre connaissance du chapitre associé.

Q23. A l'aide des sommes géométriques, si $e^{ix} \neq 1$, ce qui est le cas lorsque $x \in]0, 2\pi[$, on a

$$\begin{aligned} \sum_{k=1}^n e^{ikx} &= \sum_{k=1}^n (e^{ix})^k = e^{ix} \times \frac{e^{inx} - 1}{e^{ix} - 1} \\ &= e^{ix} \frac{e^{inx/2} (e^{inx/2} - e^{-inx/2})}{e^{ix/2} (e^{ix/2} - e^{-ix/2})} \quad (\text{factorisation par l'angle moitié}) \\ &= e^{i(n+1)x/2} \frac{2 \sin\left(\frac{nx}{2}\right)}{\sin\left(\frac{x}{2}\right)} \end{aligned}$$

Commentaires : Question de niveau standard. On évite de forcer le résultat car, comme il est donné, les correcteurs sont attentifs et exigeants sur les détails.

Q24. La suite $a_n = \frac{1}{\sqrt{n}}$ est décroissante de limite nulle. La suite $b_n = \sum_{k=1}^n e^{ikx}$ est bornée car, pour tout entier $n \geq 1$, on a :

$$|b_n| = \left| e^{i(n+1)x/2} \frac{\sin\left(\frac{nx}{2}\right)}{\sin\left(\frac{x}{2}\right)} \right| = \frac{\left| \sin\left(\frac{nx}{2}\right) \right|}{\left| \sin\left(\frac{x}{2}\right) \right|} \leq \frac{1}{\left| \sin\left(\frac{x}{2}\right) \right|}.$$

Puisque $b_n - b_{n-1} = e^{inx}$, la question 23 montre que la série $\sum_n \frac{e^{inx}}{\sqrt{n}}$ converge ce qui assure la définition de S sur $]0, 2\pi[$.

Commentaires : Question discriminante car, s'il est aisé de voir le lien avec les questions précédentes, les candidats butent généralement sur le caractère borné de b_n (des majorants dépendant de n , voire même tend vers $+\infty$ quand n tend vers $+\infty$).

Q25. D'après la relation de Chasles, on a

$$\int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt = \sum_{k=1}^{+\infty} \int_k^{k+1} \frac{e^{itx}}{\sqrt{t}} dt.$$

On a également

$$\begin{aligned} \frac{e^{ix} - 1}{ix} S(x) &= \frac{e^{ix} - 1}{ix} \sum_{k=1}^{+\infty} \frac{e^{ikx}}{\sqrt{k}} = \sum_{k=1}^{+\infty} \frac{(e^{ix} - 1) e^{ikx}}{ix\sqrt{k}} \\ &= \sum_{k=1}^{+\infty} \frac{e^{i(k+1)x} - e^{ikx}}{ix\sqrt{k}} \end{aligned}$$

Par linéarité du symbole $\sum$, on peut écrire

$$\begin{aligned} \left| \frac{e^{ix} - 1}{ix} S(x) - \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt \right| &= \left| \sum_{k=1}^{+\infty} \left(\frac{e^{i(k+1)x} - e^{ikx}}{ix\sqrt{k}} - \int_k^{k+1} \frac{e^{itx}}{\sqrt{t}} dt \right) \right| \\ &\leq \sum_{k=1}^{+\infty} \left| \frac{e^{i(k+1)x} - e^{ikx}}{ix\sqrt{k}} - \int_k^{k+1} \frac{e^{itx}}{\sqrt{t}} dt \right| \leq \sum_{k=1}^{+\infty} \frac{1}{4k^{3/2}} = C. \end{aligned}$$

Le nombre C existe bien car la série $\sum_k \frac{1}{k^{3/2}}$ converge (série de Riemann de paramètre $\frac{3}{2} > 1$).

Commentaires : Question destinée aux bonnes copies aptes à enchaîner dans le bon ordre les calculs et manipulant des expressions ayant un sens.

Q26. On effectue le changement de variable $u = tx$ donc $t = \frac{u}{x}$, $dt = \frac{du}{x}$, quand $t = 1$ alors $u = x$, quand $t \rightarrow +\infty$ alors $u \rightarrow +\infty$ (car $x > 0$). On obtient ainsi la formule

$$I(x) = \sqrt{x} \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt = \sqrt{x} \int_x^{+\infty} \frac{e^{iu}}{\sqrt{\frac{u}{x}}} \times \frac{du}{x} = \int_x^{+\infty} \frac{e^{iu}}{\sqrt{u}} du$$

qui tend, lorsque x tend vers 0, vers $\int_0^{+\infty} \frac{e^{iu}}{\sqrt{u}} du$ car, d'après les questions 10, 11 et 12, cette intégrale converge et, toujours par ces questions, elle vaut

$$\int_0^{+\infty} e^{iu^2} du = \frac{1+i}{2} \sqrt{\pi}$$

(d'après la question 21).

Commentaires : Question destinée aux bonnes copies aptes à synthétiser divers résultats et disposant d'une certaine autonomie.

Q27. D'après le développement en série entière de e^z , on peut écrire

$$e^z \underset{z \rightarrow 0}{=} 1 + z + o(z) \Rightarrow e^{ix} \underset{x \rightarrow 0}{=} 1 + ix + o(x) \Rightarrow \frac{e^{ix} - 1}{ix} = 1 + o(1) \underset{x \rightarrow 0}{\rightarrow} 1.$$

D'après la question 25, la fonction $x \mapsto \frac{e^{ix} - 1}{ix} S(x) - \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt$ est bornée donc

$$\frac{e^{ix} - 1}{ix} S(x) - \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt \underset{x \rightarrow 0}{=} O(1) \Leftrightarrow \frac{e^{ix} - 1}{ix} S(x) \underset{x \rightarrow 0}{=} \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt + O(1)$$

et d'après la question précédente, la fonction $x \mapsto \int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt$ est équivalente à $\frac{K}{\sqrt{x}}$ avec

$K = \frac{1+i}{2} \sqrt{\pi}$. En particulier, on peut écrire

$$\int_1^{+\infty} \frac{e^{itx}}{\sqrt{t}} dt \underset{x \rightarrow 0}{=} \frac{K}{\sqrt{x}} + o\left(\frac{1}{\sqrt{x}}\right)$$

d'où le développement suivant :

$$\frac{e^{ix} - 1}{ix} S(x) \underset{x \rightarrow 0}{=} \frac{K}{\sqrt{x}} + o\left(\frac{1}{\sqrt{x}}\right) + O(1) \underset{x \rightarrow 0}{\sim} \frac{K}{\sqrt{x}}.$$

D'après la question précédente, la fonction $\frac{e^{ix} - 1}{ix}$ tend vers 1 donc elle est équivalente à 1, ce qui nous donne

$$1 \underset{x \rightarrow 0}{\sim} \frac{e^{ix} - 1}{ix} \underset{\times S(x)}{\Rightarrow} S(x) \underset{x \rightarrow 0}{\sim} \frac{e^{ix} - 1}{ix} S(x) \underset{x \rightarrow 0}{\sim} \frac{K}{\sqrt{x}} = \frac{1+i}{2} \sqrt{\frac{\pi}{x}}.$$

Commentaires : La première question est élémentaire. Le reste est destinée aux meilleures copies.

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP**MATHÉMATIQUES 2****Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de deux exercices et d'un problème tous indépendants.

EXERCICE 1

Pour n entier, $n \geq 2$, on définit le déterminant de Vandermonde de n nombres complexes $x_1, x_2, \dots, x_n$ par :

$$V(x_1, x_2, \dots, x_n) = \begin{vmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_n \\ x_1^2 & x_2^2 & \dots & x_n^2 \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & \dots & x_n^{n-1} \end{vmatrix}.$$

L'objet de cet exercice est de démontrer par récurrence que l'on a : $V(x_1, x_2, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$.

Q1. Calculer $V(x_1, x_2)$. Expliquer pourquoi il suffit de faire la démonstration pour n nombres complexes $x_1, x_2, \dots, x_n$ deux à deux distincts.

Dans la suite, $x_1, x_2, \dots, x_n$ sont n nombres complexes deux à deux distincts.

Q2. On considère la fonction $t \mapsto P(t) = V(x_1, x_2, \dots, x_{n-1}, t)$.

Démontrer que P est une fonction polynomiale de degré au plus $n-1$ et justifier que le coefficient de t^{n-1} est un déterminant de Vandermonde.

Démontrer par récurrence que $V(x_1, x_2, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i)$.

Q3. Première application

Calculer le déterminant de la matrice $A = (i^j)_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n}}$ en faisant apparaître le déterminant de

Vandermonde $V(1, 2, \dots, n)$.

Q4. Deuxième application

Donner un exemple de n nombres complexes $a_1, a_2, \dots, a_n$ deux à deux distincts et tous non

nuls, tels que $\sum_{k=1}^n a_k^2 = 0$.

Soit n nombre complexes $x_1, x_2, \dots, x_n$ deux à deux distincts et tous non nuls, démontrer que

l'une au moins des sommes $\sum_{k=1}^n x_k, \sum_{k=1}^n x_k^2, \sum_{k=1}^n x_k^3, \dots, \sum_{k=1}^n x_k^n$ est non nulle.

On pourra utiliser un déterminant de Vandermonde non nul.

EXERCICE 2

Dans cet exercice, $\| \cdot \|$ désigne une norme d'algèbre sur $M_n(\mathbb{R})$, c'est-à-dire une norme vérifiant, pour tout couple (A, B) de matrices de $M_n(\mathbb{R})$, $\|A \cdot B\| \leq \|A\| \cdot \|B\|$.

Q5. Démontrer que pour toute matrice A de $M_n(\mathbb{R})$, la série $\sum_{k \geq 0} \frac{1}{k!} A^k$ converge. On notera e^A sa somme.

Q6. Démontrer que l'application $A \mapsto e^A$ est continue sur $M_n(\mathbb{R})$.

Q7. Si $H \in M_n(\mathbb{R})$ est une matrice non nulle de la boule de centre 0 et de rayon $r > 0$, déterminer

la limite de $\frac{1}{\|H\|} \sum_{k=2}^{+\infty} \frac{1}{k!} H^k$ lorsque H tend vers 0.

En déduire que l'application $A \mapsto e^A$ est différentiable en la matrice 0. On précisera sa différentielle en 0.

PROBLÈME

Pour toute matrice A de $M_n(\mathbb{R})$, on note $e^A = \sum_{k=0}^{+\infty} \frac{1}{k!} A^k$.

Dans ce problème, on note S_n l'espace vectoriel des matrices de $M_n(\mathbb{R})$ symétriques.

On dit que la matrice $A \in S_n$ est symétrique positive lorsque toutes ses valeurs propres sont positives ou nulles.

On note S_n^+ l'ensemble des matrices symétriques positives.

Partie I - Exponentielle d'une matrice symétrique

Pour a et b deux réels, on note :

$$A = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & a \end{pmatrix} \text{ et } J = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Q8. Démontrer, en détaillant les calculs, que $A \in S_3^+$, si et seulement si, $(a + 2b \geq 0 \text{ et } a \geq b)$.

Q9. Calculer J^k pour tout entier k non nul. Cette relation est-elle valable pour $k = 0$?

En utilisant la relation $A = (a - b)I_3 + bJ$, calculer et expliciter e^A .

On pourra utiliser sans démonstration que, si deux matrices A et B de $M_n(\mathbb{R})$ commutent, alors $e^{A+B} = e^A e^B$.

Vérifier que $e^A \in S_3^+$.

Q10. Soit P une matrice inversible de $M_n(\mathbb{R})$.

Justifier que l'application $M \mapsto PMP^{-1}$ est continue sur $M_n(\mathbb{R})$.

Si $A \in S_n^+$ est semblable à une matrice diagonale D , déterminer une matrice diagonale semblable à la matrice e^A .

En déduire que $e^A \in S_n^+$.

Partie II - Produit de Hadamard de deux matrices

Dans cette partie, pour une matrice $A = (a_{i,j})$ de $M_n(\mathbb{R})$, on note $E(A)$ la matrice de $M_n(\mathbb{R})$, de terme général $e^{a_{i,j}}$: $E(A) = (e^{a_{i,j}})$.

Nous allons démontrer que si $A \in S_n^+$, alors $E(A) \in S_n^+$.

On définit le produit de Hadamard de deux matrices $A = (a_{i,j})$ et $B = (b_{i,j})$ de $M_n(\mathbb{R})$ noté $*$ par :

$$A * B = (a_{i,j} b_{i,j}) \in M_n(\mathbb{R}).$$

On note le produit usuel de deux matrices A et B par AB .

On confond une matrice de $M_{1,1}(\mathbb{R})$ avec son terme réel.

Q11. Vérifier que, lorsque la matrice $A = \begin{pmatrix} a & b & b \\ b & a & b \\ b & b & a \end{pmatrix} \in S_3^+$, la matrice $E(A) \in S_3^+$.

Q12. Si D est une matrice diagonale dont tous les termes sont positifs ou nuls et si Y est matrice de $M_{n,1}(\mathbb{R})$, quel est le signe de ${}^tYDY$?

En déduire qu'une matrice A de S_n appartient à S_n^+ si, et seulement si, pour toute matrice $X \in M_{n,1}(\mathbb{R})$ on a ${}^tXAX \geq 0$.

Q13. Si A et B sont deux matrices de S_n^+ et α, β deux réels positifs, démontrer, en utilisant la question **Q12**, que $\alpha A + \beta B$ est une matrice de S_n^+ .

Si A et B sont deux matrices de S_n^+ a-t-on nécessairement $AB \in S_n^+$?

Q14. Si $A \in S_n^+$, démontrer qu'il existe une matrice $R \in S_n^+$ telle que $A = R^2$.

Q15. Si A et B sont deux matrices de S_n^+ , si on pose $A = U^2$ et $B = V^2$ avec $U = (u_{i,j}) \in S_n^+$, $V = (v_{i,j}) \in S_n^+$ et si $A * B = (c_{i,j})$, vérifier que, pour tout couple $(i,j) \in \llbracket 1, n \rrbracket^2$,

$$c_{i,j} = \left(\sum_{k=1}^n u_{k,i} u_{k,j} \right) \left(\sum_{l=1}^n v_{l,i} v_{l,j} \right).$$

En déduire que, si A et B sont deux matrices de S_n^+ , on a $A * B \in S_n^+$.

Q16. Pour toute matrice A de $M_n(\mathbb{R})$ et pour tout entier naturel $p \geq 2$, on note A^{*p} la matrice $A * A * \dots * A$ (p fois).

On note $A^{*0} = (1)$ la matrice dont tous les termes sont égaux à 1 et $A^{*1} = A$.

Soit une matrice A de $M_n(\mathbb{R})$, déterminer la limite de la suite de matrices (T_N) définie pour

$$\text{tout entier naturel } N \text{ non nul par } T_N = \sum_{p=0}^N \frac{1}{p!} A^{*p}.$$

Q17. Pour $X \in M_{n,1}(\mathbb{R})$, justifier que l'application $M \mapsto {}^tXMX$ est continue sur $M_n(\mathbb{R})$, puis démontrer que S_n^+ est une partie fermée de S_n .

En déduire que si $A \in S_n^+$ alors $E(A) \in S_n^+$.

FIN

SOLUTION

Q1. Par définition, on a :

$$V(x_1, x_2) = \prod_{1 \leq i < j \leq 2} (x_j - x_i) = x_2 - x_1$$

(car entre 1 et 2, le seul couple de nombres strictement croissant est $(1, 2)$ donc $i = 1, j = 2$). S'il existe $i_0 < j_0$ avec $x_{i_0} = x_{j_0}$ alors $x_{j_0} - x_{i_0} = 0$ est l'un des facteurs de $V(x_1, \dots, x_n)$ donc ce nombre est nul. Le déterminant de Vandermonde associé est aussi nul car ses colonnes d'indice i_0 et j_0 sont égales. Par conséquent, si deux des nombres $x_1, \dots, x_n$ sont égaux, on a

$$V(x_1, \dots, x_n) = 0 = \prod_{1 \leq i < j \leq n} (x_j - x_i).$$

Commentaires : La première question est très élémentaire. Pour la seconde, il est attendu de parler de l'annulation d'un déterminant lorsque deux colonnes sont égales et de justifier que les deux quantités sont alors égales ! Cette seconde question est discriminante.

Q2. En développant le déterminant $P(t) = V(x_1, \dots, x_{n-1}, t)$ selon sa dernière colonne, on obtient

$$P(t) = (-1)^{n+1} \Delta_{1,n} + (-1)^{n+2} t \Delta_{2,n} + \dots + (-1)^{n+n} t^{n-1} \Delta_{n,n}$$

où $\Delta_{i,n}$ est le cofacteur d'indice (i, n) (c'est à dire le déterminant obtenu en effaçant la dernière colonne et la i^{e} ligne). Comme ces nombres $\Delta_{i,n}$ sont indépendants de t , $P(t)$ est bien une fonction polynomiale de degré au plus $n - 1$. Le coefficient de t^{n-1} est

$$(-1)^{n+n} \Delta_{n,n} = \Delta_{n,n} = V(x_1, \dots, x_{n-1})$$

(car on a enlevé la dernière ligne et la dernière colonne du déterminant de Vandermonde $V(x_1, \dots, x_n)$).

Prouvons la formule attendue par récurrence sur $n \geq 2$.

Initialisation $n = 2$. Comme

$$V(x_1, x_2) = \begin{vmatrix} 1 & 1 \\ x_1 & x_2 \end{vmatrix} = x_2 - x_1 = \prod_{1 \leq i < j \leq 2} (x_j - x_i)$$

(d'après la question 1) donc la propriété est vérifiée au rang $n = 2$.

Hérédité. Supposons la propriété vérifiée pour un entier $n - 1$ avec $n \geq 3$. Si $x_1, \dots, x_n$ ne sont pas tous distincts alors, d'après la question 1, on a l'égalité

$$V(x_1, \dots, x_n) = \prod_{1 \leq i < j \leq n} (x_j - x_i).$$

Supposons maintenant les nombres $x_1, \dots, x_n$ deux à deux distincts. La fonction

$$g : t \mapsto P(x_1, \dots, x_{n-1}, t)$$

est polynomiale de degré $n - 1$ (car il est de degré au plus $n - 1$ et son coefficient de degré $n - 1$ vaut

$$V(x_1, \dots, x_{n-1}) = \prod_{1 \leq i < j \leq n-1} (x_j - x_i)$$

(par hypothèse de récurrence) qui n'est pas nul (car les nombres $x_1, \dots, x_{n-1}$ sont deux à deux distincts). Pour chaque $i \in \{1, \dots, n-1\}$, la fonction s'annule en $t = x_i$ (car les colonnes d'indice i et n sont identiques) donc $x_1, \dots, x_{n-1}$ sont $n-1$ racines distinctes de cette fonction polynomiale de degré $n-1$. Par conséquent, elle s'écrit

$$g(t) = C \prod_{i=1}^{n-1} (t - x_i)$$

où C est son coefficient dominant

$$V(x_1, \dots, x_{n-1}) = \prod_{1 \leq i < j \leq n-1} (x_j - x_i)$$

par hypothèse de récurrence. Nous avons ainsi établi la formule

$$\begin{aligned} V(x_1, \dots, x_n) &= g(x_n) = \prod_{1 \leq i < j \leq n-1} (x_j - x_i) \prod_{i=1}^{n-1} (x_n - x_i) \\ &= \prod_{1 \leq i < j \leq n-1} (x_j - x_i) \prod_{1 \leq i < j=n} (x_n - x_i) = \prod_{1 \leq i < j \leq n} (x_j - x_i) \end{aligned}$$

donc la propriété est vraie au rang n ce qui achève la récurrence.

Commentaires : Il s'agit d'une question de cours (de MPSI). La première question est un standard de deuxième année. Si elle vous a bloqué, retravaillez la, elle sert dans de nombreux concours (CCINP, Mines, Centrale) sur les questions d'espaces vectoriels normés. La seconde demande un peu de recul de la part du candidat. La troisième est un raisonnement de MPSI (lors du calcul de Vandermonde, qui est au programme officiel).

Au final, les bonnes copies parviennent à donner une réponse complète et correcte. Pour les autres, les réponses sont plus éparses.

Q3. Explicitons les coefficients de ce déterminant

$$\det(A) = \begin{vmatrix} i=1 & 1^1 & 1^2 & 1^3 & \dots & 1^n \\ i=2 & 2^1 & 2^2 & 2^3 & \dots & 2^n \\ i=3 & 3^1 & 3^2 & 3^3 & \dots & 3^n \\ \vdots & \vdots & \vdots & \vdots & & \vdots \\ i=n & n^1 & n^2 & n^3 & \dots & n^n \end{vmatrix}$$

Comme on peut factoriser chaque ligne numéro i ($2 \leq i \leq n$) par i , la linéarité du déterminant par rapport à chaque ligne nous donne

$$\begin{aligned} \det(A) &= 2 \times 3 \times \dots \times n \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ 1 & 2^1 & 2^2 & \dots & 2^{n-1} \\ 1 & 3^1 & 3^2 & \dots & 3^{n-1} \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & n^1 & n^2 & \dots & n^{n-1} \end{vmatrix} \\ &= n! V(1, 2, 3, \dots, n) = n! \prod_{1 \leq i < j \leq n} (j - i). \end{aligned}$$

Commentaires : Question sélective car elle demande de la part du candidat de la rigueur (écrire le bon déterminant initial) et une bonne connaissance du cours sur les déterminants (mettre en place la fameuse n -linéarité), etc.

Q4. Un exemple. On choisit $a_k = k$ pour chaque $k \in \{1, \dots, n-1\}$ (qui sont deux à deux distincts) et non nuls et a_n tel que

$$a_n = i \sqrt{\sum_{k=1}^{n-1} a_k^2} \Rightarrow a_n^2 = - \sum_{k=1}^{n-1} a_k^2 \Leftrightarrow \sum_{k=1}^n a_k^2 = 0.$$

Il est immédiat que a_n est non nul et distinct de tous les a_k , $1 \leq k \leq n-1$.

Soit $x_1, \dots, x_n$ deux à deux distincts et tous non nuls. Supposons que toutes les sommes $\sum_{k=1}^n x_i^q$, $1 \leq q \leq n$, sont nulles alors on a le système suivant :

$$\begin{cases} x_1 + x_2 + x_3 + \dots + x_n = 0 \\ x_1^2 + x_2^2 + x_3^2 + \dots + x_n^2 = 0 \\ x_1^3 + x_2^3 + x_3^3 + \dots + x_n^3 = 0 \\ \vdots \\ x_1^n + x_2^n + x_3^n + \dots + x_n^n = 0 \end{cases} \Leftrightarrow \underbrace{\begin{pmatrix} x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ x_1^3 & x_2^3 & x_3^3 & \dots & x_n^3 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^n & x_2^n & x_3^n & \dots & x_n^n \end{pmatrix}}_{=B} \underbrace{\begin{pmatrix} 1 \\ 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}}_{=X_0} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

La matrice B n'est pas inversible car elle possède un élément non nul dans son noyau (la colonne X_0 formée que de 1) donc son déterminant est nul. Or, en factorisant la colonne numéro i par x_i (pour $1 \leq i \leq n$), on obtient

$$\begin{aligned} \det(B) &= x_1 x_2 \dots x_n \begin{vmatrix} 1 & 1 & 1 & \dots & 1 \\ x_1 & x_2 & x_3 & \dots & x_n \\ x_1^2 & x_2^2 & x_3^2 & \dots & x_n^2 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_1^{n-1} & x_2^{n-1} & x_3^{n-1} & \dots & x_n^{n-1} \end{vmatrix} \\ &= x_1 x_2 \dots x_n V(x_1, \dots, x_n) \end{aligned}$$

qui est non nul (produit de nombres non nuls dont le Vandermonde car les x_i sont deux à deux distincts), ce qui est absurde. Par conséquent, au moins l'une des sommes $\sum_{k=1}^n x_i^q$, $1 \leq q \leq n$, est nulle.

Commentaires : Question difficile (donner un exemple non élémentaire) qui ne peut être réussi que par les meilleures copies.

Q5. Pour tout entier $k \geq 1$, on a

$$\left\| \frac{A^k}{k!} \right\| = \frac{1}{k!} \|A \times A \times \dots \times A\| \leq \frac{1}{k!} \|A\| \|A\| \dots \|A\| = \frac{\|A\|^k}{k!}.$$

Comme la série de nombres $\sum_k \frac{\|A\|^k}{k!}$ (série exponentielle de rayon de convergence $+\infty$), le théorème de comparaison des séries à termes positifs montre que la série $\sum_k \left\| \frac{A^k}{k!} \right\|$ converge.

Autrement dit, la série $\sum_k \frac{A^k}{k!}$ converge absolument dans l'espace vectoriel normé $\mathfrak{M}_n(\mathbb{R})$ qui est de dimension finie donc elle converge.

Commentaires : Il s'agit d'une question de cours mais elle s'avère en pratique sélective ! Si elle vous a posé problème (considérer l'absolue convergence, mentionné la convergence de la série exponentielle dans les réels, invoquer la dimension finie pour passer de l'absolue convergence à la convergence), reprenez cette preuve. Elle intervient souvent dans les écrits des concours (CCINP, Centrale, Mines) et aux oraux (dont plusieurs de la banque publique ...CCINP!).

Q6. Pour chaque entier k , on pose

$$f_k : A \in \mathfrak{M}_n(\mathbb{R}) \mapsto \frac{A^k}{k!} \in \mathfrak{M}_n(\mathbb{R}).$$

Justifions que chacune de ces fonctions sont continues sur $\mathfrak{M}_n(\mathbb{R})$.

Première justification. L'application

$$g : (M_1, \dots, M_k) \in (\mathfrak{M}_n(\mathbb{R}))^k \mapsto M_1 \cdots M_k \in \mathfrak{M}_n(\mathbb{R})$$

étant multilinéaire en dimension finie, elle est continue donc sa restriction au sous-ensemble $M_1 = M_2 = \cdots = M_k$ l'est aussi c'est-à-dire que la fonction $A \mapsto A^k$ est continue.

Deuxième justification. Pour chaque entier k , pour chaque indice (i, j) , $(A^k)_{i,j}$ est une somme et produit des coefficients de A (d'après le produit matriciel) donc c'est une fonction polynomiale en A d'où sa continuité. Comme ceci est vrai pour tous les coefficients (i, j) , la fonction $A \mapsto A^k$ est continue.

Soit $R \geq 0$, montrons que la série $\sum_k f_k$ converge normalement sur la boule

$$B(0, R) = \{A \in \mathfrak{M}_n(\mathbb{R}), \|A\| \leq R\}.$$

Pour tout entier k , on a :

$$\forall A \in B(0, R), \|f_k(A)\| = \left\| \frac{A^k}{k!} \right\| \leq \frac{\|A\|^k}{k!} \leq \frac{R^k}{k!}.$$

Comme le majorant est indépendant de A , on peut écrire :

$$\sup_{A \in B(0, R)} \|f_k(A)\| \leq \frac{R^k}{k!}.$$

Comme la série de nombres $\sum_k \frac{R^k}{k!}$ (série exponentielle de rayon de convergence $+\infty$), le théorème de comparaison des séries à termes positifs montre que la série $\sum_k \sup_{A \in B(0, R)} \|f_k(A)\|$

converge. Autrement dit, la série $\sum_k \frac{A^k}{k!}$ converge normalement sur la boule $B(0, R)$ donc elle y converge uniformément (car $\mathfrak{M}_n(\mathbb{R})$ est un espace vectoriel normé de dimension finie).

Par conséquent, sa somme $\sum_{k=0}^{+\infty} f_k(A) = e^A$ est continue sur toute boule $B(0, R)$ ($R \geq 0$) donc sur $\bigcup_{R \geq 0} B(0, R) = \mathfrak{M}_n(\mathbb{R})$.

Commentaires : Question difficile à ce concours, même s'il s'agit d'une question de cours (continuité de $A \mapsto \exp(A)$ de deuxième année, chapitre espaces vectoriels normés). Elle est destinée aux très bonnes copies.

Q7. Comme $\|\cdot\|$ est une norme sous-multiplicative (ou d'algèbre ou algébrique), on a

$$(*) : \left\| \sum_{k=2}^{+\infty} \frac{H^k}{k!} \right\| \leq \sum_{k=2}^{+\infty} \left\| \frac{H^k}{k!} \right\| \leq \sum_{k=2}^{+\infty} \frac{\|H\|^k}{k!} = e^{\|H\|} - 1 - \|H\|.$$

D'après le développement limité de $\exp$ en 0, on a

$$e^x = 1 + x + o(x) \Rightarrow \frac{e^x - 1 - x}{x} \underset{x \rightarrow 0}{=} o(1)$$

c'est-à-dire que

$$\lim_{x \rightarrow 0} \frac{e^x - 1 - x}{x} = 0 \Rightarrow \lim_{H \rightarrow 0} \frac{e^{\|H\|} - 1 - \|H\|}{\|H\|} = 0.$$

D'après le théorème d'encadrement, on peut affirmer grâce à la majoration (*) (divisée par $\|H\|$) que

$$\lim_{H \rightarrow 0} \frac{1}{\|H\|} \left\| \sum_{k=2}^{+\infty} \frac{H^k}{k!} \right\| = 0 \Leftrightarrow \lim_{H \rightarrow 0} \frac{1}{\|H\|} \sum_{k=2}^{+\infty} \frac{H^k}{k!} = 0$$

ce que l'on peut écrire comme

$$\sum_{k=2}^{+\infty} \frac{H^k}{k!} = o(H).$$

Par définition de l'exponentielle de matrice $f : A \mapsto e^A$, on a

$$f(H) = e^H = I + H + \sum_{k=2}^{+\infty} \frac{H^k}{k!} = I + H + o(H) = f(0) + \underbrace{H}_{=L(H)} + o(H).$$

Comme l'application $H \mapsto L(H) = H$ est linéaire, ceci est la définition de la différentiabilité de $f : A \mapsto e^A$ en 0 donc cette fonction est bien différentiable en 0 et

$$df(0) = L : H \mapsto H = \text{Id}.$$

Commentaires : La première question demande un peu de rigueur dans les calculs (vous manipuler des séries de matrices et non pas de nombres) donc elle est réussie par les très bonnes copies. Pour la seconde question, il s'agit d'une question de cours (du chapitre différentiabilité) donc elle est quand même sélective mais bien réussie par les candidat(e)s ayant traité la première question ... et par d'autres candidats quand même :-).

Q8. La matrice A est symétrique à coefficient réels donc elle est dans $\mathcal{S}_3^+(\mathbb{R})$ si et seulement si ses valeurs propres sont positives. Déterminons les via le polynôme caractéristique.

$$\chi_A(X) = \det(XI_3 - A) = \begin{vmatrix} X - a & -b & -b \\ -b & X - a & -b \\ -b & -b & X - a \end{vmatrix}$$

En additionnant toutes les lignes et en plaçant le résultat dans la première ligne, on obtient :

$$\begin{aligned}\chi_A(X) &= \begin{vmatrix} X-a-2b & X-a-2b & X-a-2b \\ -b & X-a & -b \\ -b & -b & X-a \end{vmatrix} \\ &= (X-a-2b) \begin{vmatrix} 1 & 1 & 1 \\ -b & X-a & -b \\ -b & -b & X-a \end{vmatrix}\end{aligned}$$

(car $X-a-2b$ se factorise dans la première ligne). En effectuant les opérations élémentaires $C_i \leftarrow C_i - C_1$, $2 \leq i \leq 3$, on peut écrire :

$$\begin{aligned}\chi_A(X) &= (X-a-2b) \begin{vmatrix} 1 & 0 & 0 \\ -b & X-a+b & 0 \\ -b & 0 & X-a+b \end{vmatrix} \\ &= (X-a-2b)(X-a+b)^2\end{aligned}$$

(déterminant d'une matrice triangulaire est le produit de ses coefficients diagonaux). Par conséquent, les valeurs propres de A sont les racines de ce polynôme c'est-à-dire son $a+2b$ et $a-b$ donc A appartient à $\mathcal{S}_3^+(\mathbb{R})$ si et seulement si

$$\begin{cases} a+2b \geq 0 \\ a-b \geq 0 \end{cases} \Leftrightarrow a+2b \text{ et } a \geq b.$$

Commentaires : Question de difficulté standard pour ce concours. Si le calcul du polynôme caractéristique vous a bloqué, reprenez le (pensez aux opérations sur les lignes et colonnes pour faire apparaître des lignes ou colonnes de 0).

Q9. Un calcul direct donne $J^2 = 3J$ donc

$$J^3 = JJ^2 = 3J^2 = 3^2J, J^k = JJ^3 = 3^2J^2 = 3^3J$$

et une récurrence immédiate montre que $J^k = 3^{k-1}J$ si $k \geq 1$. Si $k = 0$ alors $J^0 = I_0 \neq 3^{-1}J$ donc la formule est fautive pour $k = 0$.

Comme les matrices I et J commutent, les règles de calculs sur l'exponentielle de matrices nous donne :

$$e^A = e^{(a-b)I+bJ} = e^{(a-b)I_3} e^{bJ}.$$

Par définition, on a

$$\begin{aligned}e^{(a-b)I_3} &= \sum_{k=0}^{+\infty} \frac{((a-b)I_3)^k}{k!} = \sum_{k=0}^{+\infty} \frac{(a-b)^k}{k!} I_3 = e^{a-b} I_3 \\ e^{bJ} &= \sum_{k=0}^{+\infty} \frac{(bJ)^k}{k!} = I_3 + \sum_{k=1}^{+\infty} \frac{b^k J^k}{k!} = I_3 + \sum_{k=1}^{+\infty} \frac{b^k 3^{k-1} J}{k!} \\ &= I_3 + \frac{J}{3} \sum_{k=1}^{+\infty} \frac{(3b)^k}{k!} = I_3 + \frac{J}{3} (e^{3b} - 1).\end{aligned}$$

Par conséquent, on a prouvé que

$$\begin{aligned} e^A &= e^{a-b} I_3 \left(I_3 + \frac{J}{3} (e^{3b} - 1) \right) = e^{a-b} I_3 + \frac{J}{3} (e^{a+2b} - e^{a-b}) \\ &= \frac{1}{3} \begin{pmatrix} 2e^{a-b} + e^{a+2b} & e^{a+2b} - e^{a-b} & e^{a+2b} - e^{a-b} \\ e^{a+2b} - e^{a-b} & 2e^{a-b} + e^{a+2b} & e^{a+2b} - e^{a-b} \\ e^{a+2b} - e^{a-b} & e^{a+2b} - e^{a-b} & 2e^{a-b} + e^{a+2b} \end{pmatrix}. \end{aligned}$$

En remplaçant a par $\frac{2e^{a-b} + e^{a+2b}}{3}$ et b par $\frac{e^{a+2b} - e^{a-b}}{3}$ dans la question 7, on voit que e^A est dans $\mathcal{S}_3^+(\mathbb{R})$ si et seulement si

$$\left\{ \begin{array}{l} \frac{2e^{a-b} + e^{a+2b}}{3} + 2 \frac{e^{a+2b} - e^{a-b}}{3} \geq 0 \\ \frac{2e^{a-b} + e^{a+2b}}{3} \geq \frac{e^{a+2b} - e^{a-b}}{3} \end{array} \right\} \Leftrightarrow \left\{ \begin{array}{l} e^{a+2b} \geq 0 \\ 3e^{a-b} \geq 0 \end{array} \right.$$

ce qui est manifestement vrai donc e^A est bien dans $\mathcal{S}_3^+(\mathbb{R})$.

Commentaires : Cette matrice J est un grand classique de deuxième année. Pour le calcul de J^k , penser aux itérations de $J^2 = 3J$ pour faire émerger une formule (il n'est pas demandé de preuve ici).

Pour le calcul de l'exponentielle, il faut éviter de revenir à la définition et utiliser les règles de calculs sur l'exponentielle (si vous n'y avez pas songé, reconsolider vos connaissances sur l'exponentielle matricielle). Il est important ensuite d'être rigoureux dans la gestion des calculs. Le manque de rigueur sera alors lourdement sanctionné.

Q10. L'application $f : M \in \mathfrak{M}_n(\mathbb{R}) \mapsto PMP^{-1}$ est linéaire en dimension finie donc elle est continue. Si A est semblable à $D = \text{diag}(\lambda_1, \dots, \lambda_n)$, il existe P inversible tel que

$$A = PDP^{-1} \Rightarrow \forall k \in \mathbb{N}, A^k = PD^kP^{-1}$$

donc on peut écrire :

$$\begin{aligned} e^A &= \sum_{k=0}^{+\infty} \frac{A^k}{k!} = \sum_{k=0}^{+\infty} \frac{PD^kP^{-1}}{k!} = P \left(\sum_{k=0}^{+\infty} \text{diag} \left(\frac{\lambda_1^k}{k!}, \dots, \frac{\lambda_n^k}{k!} \right) \right) P^{-1} \\ &= P \text{diag} \left(\sum_{k=0}^{+\infty} \frac{\lambda_1^k}{k!}, \dots, \sum_{k=0}^{+\infty} \frac{\lambda_n^k}{k!} \right) P^{-1} = P \text{diag}(e^{\lambda_1}, \dots, e^{\lambda_n}) P^{-1}. \end{aligned}$$

Ainsi, e^A est semblable à $e^D = \text{diag}(e^{\lambda_1}, \dots, e^{\lambda_n})$.

Si $A \in \mathcal{S}_n^+(\mathbb{R})$, A est symétrique à coefficients réels donc A^k l'est aussi pour tout entier k donc $\sum_{k=0}^{+\infty} \frac{A^k}{k!} = e^A$ aussi. Comme ses valeurs propres sont celles de e^D c'est-à-dire les e^{λ_i} , $1 \leq i \leq n$, qui sont tous des nombres positifs, on peut affirmer que e^A appartient à $\mathcal{S}_n^+(\mathbb{R})$.

Commentaires : Les deux premières questions sont du cours donc si elles vous ont posé problème, vous devez impérativement les retravailler car elles interviennent très fréquemment à tous les concours (CCINP, Centrale, Mines) à l'écrit et à l'oral. Contre toute attente, elles se sont avérées très sélectives lors de cette épreuve.

La dernière question demande un peu de recul du candidat sur ce connaissance donc elle est sélective.

Q11. D'après la question 7, A est dans $\mathcal{S}_3^+(\mathbb{R})$ si et seulement si $a + 2b \geq 0$ et $a \geq b$. Dans ce cas, par définition de $E(A)$, on a

$$E(A) = \begin{pmatrix} e^a & e^b & e^b \\ e^b & e^a & e^b \\ e^b & e^b & e^a \end{pmatrix}$$

avec $e^a + 2e^b \geq 0$ et $e^a \geq e^b$ (par croissance de la fonction exponentielle). D'après la question 7, en remplaçant a par e^a et b par e^b , on peut affirmer que $E(A)$ appartient à $\mathcal{S}_3^+(\mathbb{R})$.

Commentaires : Question élémentaire ... pour les candidats ayant répondu correctement à la question 7. Pour les autres, elle est bien entendu inaccessible.

Q12. Si $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ avec $\lambda_i \geq 0$ pour tout $i \in \{1, \dots, n\}$ et $Y = \begin{pmatrix} y_1 \\ \vdots \\ y_n \end{pmatrix}$ alors

$$Y^T D Y = (y_1 \quad \dots \quad y_n) \begin{pmatrix} \lambda_1 y_1 \\ \vdots \\ \lambda_n y_n \end{pmatrix} = \underbrace{\lambda_1}_{\geq 0} \underbrace{(y_1)^2}_{\geq 0} + \dots + \underbrace{\lambda_n}_{\geq 0} \underbrace{(y_n)^2}_{\geq 0} \geq 0$$

(car c'est une somme et produit de nombres positifs).

Supposons que $A \in \mathcal{S}_n^+(\mathbb{R})$ alors toutes ses valeurs propres $\lambda_1, \dots, \lambda_n$ sont positives et, par le théorème spectral, elle est diagonalisable en base orthonormale c'est-à-dire qu'il existe une matrice orthogonale P et $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ telles que

$$A = P D P^{-1} = P D P^T$$

(car P est orthogonale donc $P^T P = I$ d'où $P^{-1} = P^T$). Pour toute colonne X , on a

$$X^T A X = X^T P D P^T X.$$

Si on pose $Y = P^T X$ qui est une matrice colonne alors $Y^T = X^T (P^T)^T = X^T P$, on a

$$X^T A X = Y^T D Y \geq 0$$

d'après le début de cette question.

Réciproquement, si A est symétrique à coefficients réels et, pour toute matrice colonne X , on a $X^T A X \geq 0$. Soit λ une valeur propre de A , il existe une colonne X_λ non nulle tel que $A X_\lambda = \lambda X_\lambda$ donc

$$X_\lambda^T A X_\lambda = X_\lambda^T \lambda X_\lambda = \lambda X_\lambda^T X_\lambda = \lambda \|X_\lambda\|^2$$

(car le produit scalaire canonique de deux matrices colonnes X et Y est $\langle X | Y \rangle = X^T Y$ donc la norme euclidienne canonique d'une matrice colonne est $\|X\|^2 = X^T X$). En divisant par $\|X_\lambda\|^2$ qui est non nul (car X_λ n'est pas nul), on obtient

$$\lambda = \frac{X_\lambda^T A X_\lambda}{\|X_\lambda\|^2} \geq 0$$

(comme quotient de deux nombres positifs). Ainsi, toutes les valeurs propres de A sont positives donc A appartient à $\mathcal{S}_n^+(\mathbb{R})$.

Commentaires : Il s'agit d'une question de cours. Si elle vous a posé problème, vous devez impérativement la retravailler car elle intervient très fréquemment à tous les concours (CCINP, Centrale, Mines) à l'écrit et à l'oral. L'implication réciproque est aisée (un peu astucieuse) ... mais il faut impérativement indiquer de le vecteur propre associé est non nul (pour effectuer la division). L'implication directe est plus difficile pour les candidats.

Au final, la question est très discriminante (mais largement accessible).

Q13. La matrice $M = \alpha A + \beta B$ est à coefficients réels (comme combinaison linéaire de telles matrices) et symétriques car, par linéarité de la transposition, on a :

$$M^T = \alpha A^T + \beta B^T = \alpha A + \beta B = M.$$

Pour toute colonne X , $X^T A X$ et $X^T B X$ sont des nombres positifs (d'après la question précédente) et on a :

$$X^T M X = \underbrace{\alpha}_{\geq 0} \underbrace{X^T A X}_{\geq 0} + \underbrace{\beta}_{\geq 0} \underbrace{X^T B X}_{\geq 0} \geq 0$$

(comme somme et produit de nombres positifs) donc, d'après la question précédente, M appartient à $\mathcal{S}_n^+(\mathbb{R})$.

Le produit de deux matrices symétriques n'est pas symétrique en général ! Par exemple, si on prend $A = \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ (symétrique à valeurs propres 0 et 2 car son déterminant vaut 0 donc

l'une des valeurs propres est nulle et sa trace vaut 2 donc l'autre vaut 2) et $B = \begin{pmatrix} 1 & 1 \\ 1 & 2 \end{pmatrix}$ (symétriques à valeurs propres positives car son déterminant vaut 1 donc le produit des valeurs propres est positif, ce qui assure qu'elles ont le même signe) et sa trace est positive, ce qui assure qu'elles sont toutes deux positives) alors

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

n'est pas une matrice symétrique !

Commentaires : La première question est de difficulté standard. La seconde est difficile comme toute question de contre-exemple.

Q14. Si $A \in \mathcal{S}_n^+(\mathbb{R})$, d'après le théorème spectral, il existe une matrice orthogonale P et une matrice diagonale $D = \text{diag}(\lambda_1, \dots, \lambda_n)$ telles que

$$A = P D P^{-1} \text{ avec } \forall i \in \{1, \dots, n\}, \lambda_i \geq 0.$$

Si on pose

$$R = P \Delta P^{-1} \text{ avec } \Delta = \text{diag}(\sqrt{\lambda_1}, \dots, \sqrt{\lambda_n})$$

alors

$$R^2 = P \Delta^2 P^{-1} = P D P^{-1} = A \text{ et } R^T = (P^{-1})^T \Delta^T P^T = P \Delta P^{-1} = R$$

(car P étant orthogonale, $P^{-1} = P^T$) donc R est bien symétrique à coefficients réels (comme produit de matrices à coefficients réels) et à valeurs propres positives (les $\sqrt{\lambda_i}$, $1 \leq i \leq n$) donc elle appartient bien à $\mathcal{S}_n^+(\mathbb{R})$.

Commentaires : Il s'agit d'une question de cours. Si elle vous a posé problème, vous devez impérativement la retravailler car elle intervient très fréquemment à tous les concours (CCINP, Centrale, Mines) à l'écrit et à l'oral. Le caractère symétrique de R résulte uniquement du caractère orthogonal de P . Si cela n'est pas mentionné par le candidat, il sera lourdement sanctionné.
La question est sélective (mais accessible).

Q15. Comme U est symétrique, on a $U^T = U$ et par définition du produit matriciel, on a pour tout couple (i, j) :

$$A_{i,j} = (UU)_{i,j} = (U^T U)_{i,j} = \sum_{k=1}^n (U^T)_{i,k} U_{k,j} = \sum_{k=1}^n U_{k,i} U_{k,j}.$$

Idem avec V donc

$$B_{i,j} = \sum_{l=1}^n V_{l,i} V_{l,j}.$$

ce qui permet d'écrire :

$$(A * B)_{i,j} = A_{i,j} B_{i,j} = \sum_{k=1}^n U_{k,i} U_{k,j} \sum_{l=1}^n V_{l,i} V_{l,j}.$$

Comme A et B sont symétriques pour tout couple (i, j) , on a :

$$(A * B)_{j,i} = A_{j,i} B_{j,i} = A_{i,j} B_{i,j} = (A * B)_{i,j}$$

donc $A * B$ est symétrique. Comme A et B sont à coefficients réels, $A * B$ l'est aussi (ses coefficients sont des produits de réels donc des réels). Pour montrer que $A * B$ est symétrique positive, nous allons utiliser la caractérisation de la question 12. Pour toute matrice colonne X et toute indice $i \in \{1, \dots, n\}$ (représentant les lignes), on a :

$$\begin{aligned} ((A * B) X)_i &= \sum_{j=1}^n (A * B)_{i,j} X_j = \sum_{j=1}^n \sum_{k=1}^n U_{k,i} U_{k,j} \sum_{l=1}^n V_{l,i} V_{l,j} X_j \\ &= \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n U_{k,i} U_{k,j} V_{l,i} V_{l,j} X_j \end{aligned}$$

donc

$$\begin{aligned} X^T (A * B) X &= \sum_{i=1}^n X_i ((A * B) X)_i = \sum_{i=1}^n X_i \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n U_{k,i} U_{k,j} V_{l,i} V_{l,j} X_j \\ &= \sum_{i=1}^n \sum_{j=1}^n \sum_{k=1}^n \sum_{l=1}^n (X_i U_{k,i} U_{k,j}) (X_j V_{l,i} V_{l,j}) \end{aligned}$$

En utilisant Fubini (en regroupant les indices i et k puis j et l), on obtient

$$\begin{aligned} X^T (A * B) X &= \sum_{(i,k) \in \{1, \dots, n\}^2} \sum_{(j,l) \in \{1, \dots, n\}^2} \underbrace{(X_i U_{k,i} U_{k,j})}_{\text{indépendant de } (j,l)} X_j V_{l,i} V_{l,j} \\ &= \underbrace{\sum_{(i,k) \in \{1, \dots, n\}^2} X_i U_{k,i} U_{k,j}}_{=S} \underbrace{\sum_{(j,l) \in \{1, \dots, n\}^2} X_j V_{l,i} V_{l,j}}_{=T}. \end{aligned}$$

Puisque les sommes S et T sont égales (indices (i, k) et (j, l) sont muets) et forment un nombre réel (comme somme et produit de tels nombres) on obtient que

$$X^T (A * B) X = S^2$$

est un nombre positif (puisque le carré d'un réel). D'après la question 12, $A * B$ appartient à $\mathcal{S}_n^+(\mathbb{R})$.

Commentaires : La première question est de difficulté standard (connaissance correcte des formules de calculs matriciels via les coefficients). La deuxième question est difficile et elle est destinée aux meilleures copies.

Q16. Pour tout couple d'indices (i, j) , on a $(A^{*p})_{i,j} = A_{i,j}^p$ (par une récurrence immédiate) donc

$$(T_N)_{i,j} = \sum_{p=0}^N \frac{1}{p!} (A^{*p})_{i,j} = \sum_{p=0}^N \frac{A_{i,j}^p}{p!}$$

converge vers $e^{A_{i,j}}$ donc la suite (T_N) converge vers la matrice $(e^{A_{i,j}})_{1 \leq i, j \leq n} = E(A)$.

Commentaires : Question de niveau standard et application directe du cours (rapeler par le sujet dans la question 5).

Q17. L'application $f_X : M \in \mathfrak{M}_n(\mathbb{R}) \mapsto X^T M X$ est linéaire en dimension finie donc elle est continue. D'après la question 12,

$$\mathcal{S}_n^+(\mathbb{R}) = \{M \in \mathcal{S}_n(\mathbb{R}), \forall X \in \mathfrak{M}_{n,1}(\mathbb{R}), X^T M X \geq 0\}$$

Soit $(M_k)_{k \geq 0}$ une suite de $\mathcal{S}_n^+(\mathbb{R})$ qui converge vers $M_\infty \in \mathcal{S}_n(\mathbb{R})$ (on veut un fermé de $\mathcal{S}_n(\mathbb{R})$). Soit $X \in \mathfrak{M}_{n,1}(\mathbb{R})$ alors, pour tout entier k , on a :

$$X^T M_k X \geq 0 \Leftrightarrow f_X(M_k) \geq 0.$$

Comme f est continue, on peut affirmer que $f(M_k) \xrightarrow{k \rightarrow +\infty} f(M)$ donc, en faisant tendre k vers $+\infty$ dans l'inégalité ci-dessous, on obtient

$$f_X(M) \geq 0 \Leftrightarrow X^T M X \geq 0$$

pour tout $X \in \mathfrak{M}_{n,1}(\mathbb{R})$ donc $M \in \mathcal{S}_n^+(\mathbb{R})$. Par conséquent, $\mathcal{S}_n^+(\mathbb{R})$ est bien un fermé de $\mathcal{S}_n(\mathbb{R})$.

Pour tout entier N , la suite $(T_N)_N$ de la question précédente appartient à $\mathcal{S}_n^+(\mathbb{R})$. En effet, comme $A \in \mathcal{S}_n^+(\mathbb{R})$, la question 15 et une récurrence immédiate montre $A^{*p} \in \mathcal{S}_n^+(\mathbb{R})$ pour tout entier p . Comme tous les réels $\frac{1}{p!}$ sont positifs, par la question 13 et une récurrence

immédiate $\sum_{p=0}^N \frac{A^{*p}}{p!} = T_N$ appartient à $\mathcal{S}_n^+(\mathbb{R})$. Comme cette suite converge vers $E(A)$ et que $\mathcal{S}_n^+(\mathbb{R})$ est fermé, sa limite $E(A)$ appartient à $\mathcal{S}_n^+(\mathbb{R})$.

Commentaires : La première question est une question de cours. Si elle vous a posé problème, la retravailler (utilisation courante tant à l'écrit qu'à l'oral à tous les concours CCINP, Mines, Centrale).

La deuxième question est une application directe du cours. On peut la traiter également par des images réciproques mais il faut alors remplacer le quantificateur $\forall X, F_X(M) \in \mathbb{R}_+$ par une union ensembliste $\bigcup_X F_X^{-1}(\mathbb{R}_+)$ et montrer que $F_X = \{M, X^T M X \geq 0\} = f_X^{-1}(\mathbb{R}_+)$ est un ouvert (image réciproque d'un fermé par une fonction continue) puis conclure via une réunion de fermés est un fermé. Si cette question vous a posé des difficultés et vous visez aussi des Mines-Telecom ou des écoles des concours Mines ou Centrale, retravaillez cette question (qui est un standard de ces concours, écrit comme oral).

La troisième question est destinée aux bonnes copies.

ÉPREUVE MUTUALISÉE AVEC E3A-POLYTECH**ÉPREUVE SPÉCIFIQUE - FILIÈRE MP**

INFORMATIQUE**Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de trois parties, toutes indépendantes.

Partie I - Automates

L'objectif de cette partie est de proposer quelques éléments autour d'un automate, dit augmenté, construit autour d'un automate fini non déterministe et d'un sous-ensemble d'états.

Définition 1 (Automate fini non déterministe)

Un *automate fini non déterministe* est un quintuplet $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ avec :

- Q un ensemble fini non vide d'états, de cardinal $|Q|$;
- Σ un alphabet;
- $I \subset Q$ l'ensemble des états initiaux;
- $\delta : Q \times \Sigma \rightarrow \mathcal{P}(Q)$ une fonction de transition : si $q \in Q$ et $a \in \Sigma$, $\delta(q, a)$ désigne l'ensemble des états q' de Q tels qu'il existe une transition étiquetée par a de q vers q' ;
- $F \subset Q$ l'ensemble des états finaux.

Définition 2 (Automate augmenté)

Soient $\mathcal{A} = (Q, \Sigma, I, \delta, F)$ un automate fini non déterministe et $O \subset Q$ un sous-ensemble d'états de Q . On appelle *automate augmenté* par O la paire $(\mathcal{A}, O)$, que l'on notera par la suite $\mathcal{A}_O$.

La notion de calcul est la même entre $\mathcal{A}$ et $\mathcal{A}_O$: un calcul dans $\mathcal{A}$ est une séquence d'états $q_1 \cdots q_n$ de Q telle qu'il existe toujours au moins une transition entre deux états successifs de cette séquence. Le mot construit par concaténation des étiquettes de chacune des transitions est alors reconnu par l'automate.

L'ensemble O introduit la notion de *calcul réussi au seuil s* .

Définition 3 (Calcul réussi au seuil s)

Soit $\mathcal{A}_O$ un automate augmenté. Un calcul passant par les états $q_i \in Q, i \in \llbracket 0, n \rrbracket$ est dit *réussi au seuil $s \in \mathbb{N}$* si :

- i) $q_0 \in I$;
- ii) $q_n \in F$;
- iii) pour tout $i \geq 0$ tel que $i + s \leq n$, $\{q_i \cdots q_{i+s}\} \cap O \neq \emptyset$.

Définition 4 (Langage reconnu par un automate augmenté au seuil s)

Soit $\mathcal{A}_O$ un automate augmenté et $s \in \mathbb{N}$. Le *langage reconnu par $\mathcal{A}_O$ au seuil s* , noté $L_s(\mathcal{A}_O)$, est l'ensemble des calculs réussis par $\mathcal{A}_O$ au seuil s .

On considère, pour les **questions 1 à 3**, l'automate de la **figure 1**, où $Q = \{q_0, q_1, q_2\}$, $\Sigma = \{a, b\}$, $I = \{q_0\}$ et $F = \{q_2\}$.

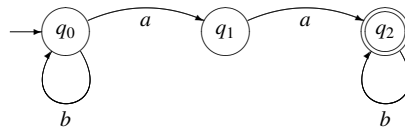


Figure 1 - Automate exemple

- Q1.** Donner sans justification $L_2(\mathcal{A}_{\{q_0\}})$.
- Q2.** Donner, en justifiant votre réponse, $L_2(\mathcal{A}_{\{q_1\}})$.
- Q3.** Soit $s \in \mathbb{N}$. Donner, en justifiant votre réponse, le cardinal de $L_s(\mathcal{A}_O)$ en fonction de s , où $\emptyset$ est l'ensemble vide.
- Q4.** Donner, sans justification, un majorant des calculs réussis au seuil s ne passant par aucun état de O .

- Q5.** Soit maintenant un calcul réussi au seuil s qui passe au moins par un état de O . On note $\tilde{q} = q_0 \cdots q_l$ la suite des états correspondants et $i_0 \cdots i_k \in \llbracket 0, l \rrbracket$ la suite des indices dans $\tilde{q}$ correspondant aux états de O . Donner, sans justification, en fonction de s :
- i) une majoration de i_0 ;
 - ii) un encadrement de i_k ;
 - iii) une majoration de $i_{j+1} - i_j$ pour $j \in \llbracket 0, k-1 \rrbracket$.
- Q6.** Soit $\mathcal{A}_O$ un automate augmenté, avec $|Q| = p$, où Q est l'ensemble des états de l'automate. Pour $s \in \mathbb{N}$, construire, en justifiant votre construction, un automate fini à $(s+1)p$ états reconnaissant $L_s(\mathcal{A}_O)$.

Partie II - Autour des tas

Cette partie comporte des questions de programmation qui seront abordées en utilisant **exclusivement le langage Python** (Informatique Pour Tous).

L'objectif est ici d'étudier et d'implémenter quelques outils autour d'une structure de données appelée *tas binomial*. Un tas binomial est une structure assez proche du tas binaire (utilisé par exemple pour réaliser une file de priorité), pour lequel la procédure de fusion de deux tas est efficace et peu complexe.

II.1 - Arbre binomial

Définition 5 (Arbre enraciné)

Un *arbre enraciné* est un graphe acyclique orienté possédant une unique racine et tel que tous les nœuds sauf la racine ont un unique parent.

La **figure 2** présente un exemple d'arbre enraciné dans lequel r est la racine.

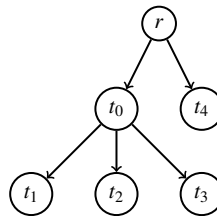


Figure 2 - Exemple d'arbre enraciné

On définit un arbre enraciné (non vide) par un couple $(r, [t_0, \dots, t_{n-1}])$, où r est la valeur de la racine et $[t_0, \dots, t_{n-1}]$ est la liste de ses fils, chaque t_i , $i \in \llbracket 0, n-1 \rrbracket$ étant un arbre. Un arbre vide est défini par `None`. Par abus de notation, on confond dans la suite la racine de l'arbre et sa valeur, ainsi que les fils d'un arbre avec leur racine.

- Q7.** Écrire les fonctions Python :
- `Vide(a)` qui renvoie `True` si l'arbre `a` est vide, `False` sinon ;
 - `Racine(a)` qui renvoie la racine de `a` si `a` est non vide ;
 - `Fils(a)` qui renvoie la liste des arbres, fils de la racine de `a`.

Définition 6 (Arbre binomial)

Un *arbre binomial* a_k d'ordre $k \geq 0$ est un arbre enraciné dans lequel les fils de chaque nœud sont ordonnés. Il est défini récursivement comme suit :

- i) $a_0 = (x, [])$ est constitué d'un nœud unique, la racine ;
- ii) pour $k \in \mathbb{N}$, soit $a_k = (x, [t_0, \dots, t_{n-1}])$ un arbre non vide. a_k est un arbre binomial d'ordre k si :
 - t_{n-1} est un arbre binomial d'ordre $(k-1)$;
 - $(x, [t_0, \dots, t_{n-2}])$ est un arbre binomial d'ordre $(k-1)$;
 - la racine de t_{n-1} a une valeur supérieure ou égale à x .

La **figure 3** donne un exemple d'arbre binomial d'ordre 3. Les valeurs dans l'arbre sont des entiers.

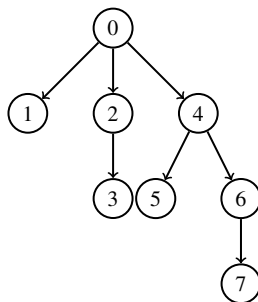


Figure 3 - Exemple d'arbre binomial d'ordre 3

- Q8.** Écrire une fonction Python `ArbreBinomial(a1,a2)` qui construit, à partir de deux arbres binomiaux a_1 et a_2 d'ordre $k-1$, un arbre binomial a_k d'ordre k contenant les mêmes valeurs que celles des arbres a_1 et a_2 . Dans la suite, on note $a_1 \oplus a_2$ cette opération.
- Q9.** Montrer par récurrence que la racine d'un arbre binomial d'ordre k a exactement k fils.
- Q10.** En déduire une fonction Python `Ordre(a)` qui renvoie l'ordre de l'arbre binomial a .
- Q11.** Montrer qu'un arbre binomial a d'ordre k possède 2^k nœuds.
- Q12.** Écrire une fonction récursive Python `EstUnArbreBinomial(a)` qui renvoie `True` si a est un arbre binomial, `False` sinon.

II.2 - Tas binomial

Un *tas* est une structure de données de type arbre qui permet en particulier de retrouver directement un élément qui doit être traité en priorité.

Définition 7 (Tas binomial)

Soient $k \geq 0$ et $T = \{a_0, \dots, a_k\}$ un ensemble d'arbres. T est un *tas binomial* de longueur $k+1$ si, pour tout $i \in \llbracket 0, k \rrbracket$, a_i est soit un arbre vide, soit un arbre binomial d'ordre i . Si $i = k$, a_i ne peut, de plus, pas être vide et a_k est donc un arbre binomial d'ordre k .

On note dans la suite $|T|$ le nombre de nœuds d'un tas T .

- Q13.** Quelle structure Python adopter pour coder un tas ?

Définition 8 (Signature d'un tas)

Soit $T = \{a_0, \dots, a_k\}$ un tas binomial de longueur $k+1$. On appelle *signature* de T la suite $s_0 \dots s_k$ telle que pour tout $i \in \llbracket 0, k \rrbracket$, $s_i = 0$ (respectivement $s_i = 1$) si l'arbre a_i est vide (respectivement n'est pas vide).

Q14. Soit T un tas binomial de longueur $k + 1$. En utilisant sa signature, calculer $|T|$ et montrer que $2^k \leq |T| < 2^{k+1}$. En déduire k en fonction de $|T|$.

Q15. Écrire une fonction Python `MinimumTas(T)` qui retourne la valeur minimum du tas T . En donner la complexité en fonction de $|T|$.

Les tas se construisent itérativement à partir de données. On est donc amené, pour un tas T , à ajouter un à un des éléments.

Soit p un élément que l'on souhaite ajouter à un tas binomial T non vide et déjà construit. L'insertion de la valeur p dans le tas T se fait alors selon l'algorithme 1.

Algorithme 1 - Insertion de p dans T .

Entrées : un tas $T = \{a_0 \dots a_k\}$, une valeur p

Sorties : un tas T augmenté de la valeur p

début

$i \leftarrow 0$

 Coder p dans un arbre binomial a d'ordre 0

tant que $i < k + 1$ **et** a non vide **faire**

si a_i est vide **alors**

$a_i \leftarrow a$

 Vider a

sinon

$a \leftarrow a \oplus a_i$ (***)

 Vider a_i

$i \leftarrow i + 1$

si a n'est pas vide **alors**

 Ajouter a au tas T .

Q16. Coder l'algorithme 1 sous la forme d'une fonction Python `Insertion(p, T)`.

Q17. Évaluer la complexité de cet algorithme en fonction de $|T|$. On suppose que l'étape marquée (***) s'effectue en temps constant.

Q18. Donner la signature du tas résultant de l'insertion de p dans T en fonction de la signature de T .

Q19. Donner, sans justification, un invariant de boucle pour la boucle de l'algorithme 1 permettant de prouver la correction de ce dernier.

Partie III - Autour de l'énumération des fractions positives

Cette partie comporte des questions nécessitant un **code OCaml**. Pour ces questions, **les réponses ne feront pas appel aux fonctionnalités impératives du langage** (en particulier pas de boucles, pas de références).

Notations

Dans la suite, on notera :

- $n \wedge d$ le PGCD (Plus Grand Commun Diviseur) de n et d ;
- $\lceil x \rceil$ la partie entière supérieure de x . Ainsi $\lceil 3.45 \rceil = 4$;
- $\lfloor x \rfloor$ la partie entière inférieure de x . Ainsi $\lfloor 3.45 \rfloor = 3$;
- $\log_2$ la fonction logarithme de base 2. C'est la fonction réciproque de la fonction $i \mapsto 2^i$.

Une *fraction*, ou *nombre rationnel*, est le quotient de deux nombres entiers, le dénominateur étant par définition non nul. On notera $\mathbb{Q}^+$ l'ensemble des fractions positives.

L'objectif de cette partie est d'étudier une structure de données permettant d'énumérer l'ensemble des fractions positives.

Plusieurs travaux se sont intéressés à l'énumération des nombres rationnels, le plus connu étant très certainement la méthode de Cantor. Cependant, il n'est très souvent pas possible, à moins d'un travail assez complexe, de connaître une formule générale donnant le i -ième terme de cette énumération, ni même de donner le successeur dans l'énumération d'une fraction donnée.

Nous proposons dans la suite de répondre à ces questions en construisant un arbre, dit de Calkin-Wilf, permettant de manipuler cette énumération.

Définition 9 (Arbre binaire infini)

Un *arbre binaire homogène* est un arbre binaire dont tous les nœuds ont 0 successeur ou 2 successeurs, appelés fils gauche et droit. La *hauteur* h de l'arbre est la profondeur maximale des nœuds de l'arbre, c'est-à-dire la plus grande longueur d'un chemin de la racine vers une feuille de l'arbre. Lorsque h est infinie, on parle d'*arbre infini*.

Définition 10 (Arbre de Calkin-Wilf)

L'*arbre de Calkin-Wilf* est un arbre binaire homogène infini dont les nœuds sont des fractions positives. La racine de l'arbre est la fraction $1/1$. Chaque sommet n/d a deux descendants : un fils gauche $n/(n+d)$ et un fils droit $(n+d)/d$.

Ainsi, si $N = n/d$, les deux fils de N sont $\frac{N}{1+N}$ (gauche) et $1+N$ (droit).

Q20. Dessiner l'arbre de Calkin-Wilf jusqu'à une profondeur de 3. Par convention, la racine de l'arbre est au niveau 0.

On propose le type record :

```
type fraction = {n:int; d:int}
```

pour définir une fraction de type n/d . La déclaration d'une telle fraction sera donc du type :

```
let f = {n=2;d=3};;
```

l'accès au numérateur (respectivement dénominateur) s'opérant par $f.n$; (resp. $f.d$;).

Q21. Proposer un type OCaml récursif permettant de décrire l'arbre de Calkin-Wilf.

Q22. Montrer par récurrence sur le niveau d'exploration de l'arbre que si n/d est un nœud de l'arbre, alors n et d sont premiers entre eux.

Q23. Écrire une fonction récursive OCaml de signature `pgcd : int * int -> int` qui calcule le PGCD de deux entiers naturels.

Q24. Écrire une fonction récursive OCaml de signature `fraction:int->int->fraction` qui construit une fraction positive irréductible à partir d'un numérateur n et un dénominateur d . La fonction vérifie que $n > 0$ et $d > 0$. Au besoin, elle simplifie la fraction par $n \wedge d$.

Par la suite, on ne construira des valeurs de type `fraction` que par l'intermédiaire de la fonction `fraction`, ce qui permettra de garantir l'invariant de type suivant : "pour toute valeur $f : \text{fraction}$, $f.n$ et $f.d$ sont premiers entre eux".

Q25. Soient deux nœuds k/l et n/d de l'arbre ayant des fils gauches (ou droits) identiques. Montrer qu'alors $k = n$ et $l = d$.

Q26. Soient N un nœud et $k \in \mathbb{N}$. Montrer que le nœud provenant d'une suite de k fils gauches de N est le nœud $\frac{N}{1+kN}$. Donner sans démonstration le nœud provenant d'une suite de k fils droits de N .

On définit alors une suite $(v_i)_{i \in \mathbb{N}}$, avec $v_0 = 0$, puis en effectuant un parcours en largeur, de gauche à droite, de l'arbre de Calkin-Wilf pour définir les termes de la suite.

Q27. Donner les huit premiers termes de la suite $(v_i)_{i \in \mathbb{N}}$.

Q28. Soient $n, d \in \mathbb{N}^*$ premiers entre eux. Montrer par récurrence sur $n + d$ que toute fraction n/d apparaît dans l'arbre.

Q29. Montrer qu'aucune fraction n'apparaît deux fois dans l'arbre.

Q30. Dédurre des questions précédentes que la suite $(v_i)_{i \in \mathbb{N}}$ est une bijection de $\mathbb{N}$ dans $\mathbb{Q}^+$.

La suite $(v_i)_{i \in \mathbb{N}}$ permet donc d'affirmer que $\mathbb{Q}^+$ est dénombrable. Nous allons utiliser $(v_i)_{i \in \mathbb{N}}$ pour déterminer, dans l'énumération des fractions produite par cette suite, la position de n'importe quelle fraction positive n/d . En d'autres termes, étant donnée une fraction positive n/d , on se propose de rechercher i tel que $n/d = v_i$.

Q31. Soit $i \in \mathbb{N}^*$. Montrer que le fils gauche du nœud de valeur v_i a pour valeur v_{2i} . Montrer de même que le fils droit a pour valeur v_{2i+1} .

Pour pouvoir énoncer le i -ième terme dans cette énumération, on introduit alors une suite auxiliaire, aux nombreuses propriétés arithmétiques et liens avec d'autres objets mathématiques.

La *suite diatomique de Stern* (ou *suite de Stern-Brocot*) doit son nom à Moritz Stern (1807-1894), élève de Gauss et Achille Brocot (1817-1878), horloger qui s'intéressait aux fractions pour la fabrication d'horloges avec des engrenages comportant peu de dents, donc simples à fabriquer.

Définition 11 (Suite diatomique de Stern ou suite de Stern-Brocot)

La *suite diatomique de Stern* $(s_i)_{i \in \mathbb{N}}$ est définie par $s_0 = 0$, $s_1 = 1$ et pour tout $i \geq 1$:

$$\begin{cases} s_{2i} &= s_i \\ s_{2i+1} &= s_i + s_{i+1} \end{cases}.$$

Q32. Donner les dix premiers termes de la suite $(s_i)_{i \in \mathbb{N}}$.

Q33. Écrire une fonction récursive OCaml de signature `stern : int -> int` permettant de calculer les termes de cette suite.

Q34. Dédurre par récurrence de la **Q31** que $\forall i \in \mathbb{N}$, $v_i = \frac{s_i}{s_{i+1}}$.

La suite diatomique de Stern permet d'exprimer le i -ième terme dans l'énumération des fractions positives qui est induite par le parcours en largeur de l'arbre de Calkin-Wilf décrit ci-dessus.

Voyons maintenant comment obtenir de manière rapide le successeur d'une fraction v_i donnée, sans connaître i , dans cette énumération. Trois cas peuvent se présenter :

- premier cas : les nœuds de valeur v_i et v_{i+1} sont à même profondeur k , fils d'un même nœud N ,
- deuxième cas : le nœud de valeur v_i est le dernier nœud à droite à la profondeur k ,
- troisième cas : les nœuds de valeur v_i et v_{i+1} sont à même profondeur k , mais ne sont pas fils d'un même nœud.

On pose pour tout $x > 0$, $f(x) = \frac{1}{1 + 2\lfloor x \rfloor - x}$.

Q35. Montrer que dans le premier cas, $v_{i+1} = f(v_i)$.

Q36. Montrer, en utilisant la **Q26**, que dans le deuxième cas on a encore $v_{i+1} = f(v_i)$.

On étudie enfin le dernier cas : les nœuds de valeur v_i et v_{i+1} sont sur une même profondeur k , mais ne sont pas les fils d'un même nœud. On va donc passer par la recherche d'un ancêtre commun de ces deux nœuds. Dans la suite, on s'intéressera toujours au premier ancêtre commun, c'est-à-dire celui de profondeur maximale.

En partant de la racine r , il est possible d'atteindre n'importe quel nœud $N = n/d$ de l'arbre par une suite de déplacements vers la gauche (G) ou vers la droite (D). Le *chemin* de r vers N peut donc être codé par un mot sur l'alphabet $\{G, D\}$.

Si un déplacement est représenté en OCaml par un type

```
type direction = G | D
```

alors un chemin est une liste `direction list`.

- Q37.** Écrire une fonction OCaml de signature `chemin : fraction -> direction list` qui calcule le chemin de la racine à un nœud quelconque de l'arbre. Cette fonction fera appel à une fonction auxiliaire récursive. Ainsi `chemin n d` calcule la liste des directions à prendre pour passer de `r` au nœud `n/d`. On pourra supposer l'existence d'une fonction de signature `rev : direction list -> direction list` qui renverse une liste.
- Q38.** Écrire une fonction OCaml de signature `noeud : direction list -> fraction` qui détermine le nœud obtenu en effectuant une série de déplacements depuis la racine. Ainsi `noeud [D;D;G;D]` renverra un couple d'entiers (n, d) correspondant au nœud de valeur n/d . Cette fonction fera appel à une fonction auxiliaire récursive.
- Q39.** Utiliser les deux fonctions précédentes pour écrire une fonction OCaml ancêtre de signature `ancetre : fraction -> fraction -> fraction` qui détermine le premier ancêtre commun entre deux nœuds. Ainsi `ancetre (n,d) (p,q)` détermine le premier ancêtre commun à n/d et p/q . Cette fonction pourra utiliser une fonction auxiliaire récursive.
- Q40.** On suppose que le nœud de valeur v_p , le premier ancêtre commun des nœuds de valeur v_i et v_{i+1} , est à k' niveaux au-dessus d'eux. Donner le chemin entre v_p et v_i sous la forme d'une liste de direction. Donner de même le chemin entre v_p et v_{i+1} .
- Q41.** À partir de l'expression des fils gauche et droit de v_p , montrer que l'on a encore $v_{i+1} = f(v_i)$.

FIN

SOLUTION

- Q1.** Il s'agit de déterminer tous les mots reconnus par $\mathcal{A}$ obtenus par tout chemin partant d'un état initial, finissant à un état final et dont tous les trois états consécutifs passent par q_0 . Cela a pour contrainte de ne lire aucune lettre en arrivant à l'état q_2 donc $L_2(\mathcal{A}_{\{q_0\}}) = b^*aa$.

Commentaires : Question ayant pour but de familiariser le candidat au nouveau concept introduit par le sujet. Il est important de traiter avec sérieux cette question sous peine d'avoir du mal à répondre aux questions ultérieures.

- Q2.** Il s'agit de déterminer tous les mots reconnus par $\mathcal{A}$ obtenus par tout chemin partant d'un état initial, finissant à un état final et dont tous les trois états consécutifs passent par q_1 . Cela a pour contrainte de ne lire qu'une fois au plus b depuis l'état q_0 et l'état q_2 . Les seuls mots reconnus par $\mathcal{A}$ vérifiant ces contraintes sont

$$L_2(\mathcal{A}_{\{q_1\}}) = \{aa, baa, aab, baab\}$$

Commentaires : Même remarque qu'à la question précédente.

- Q3.** Il s'agit de déterminer tous les mots reconnus par $\mathcal{A}$ obtenus par tout chemin partant d'un état initial, finissant à un état final et dont tous les $s+1$ états consécutifs rencontrent l'ensemble vide! Cette condition C'est impossible ... sauf si s dépasse strictement $n = 2$. Ainsi, $L_s(\mathcal{A}_\emptyset) = \emptyset$ si $s \leq n = 2$ et lorsque $s > n = 2$ alors

$$L_s(\mathcal{A}_\emptyset) = \{b^i aab^j, (i, j) \in \mathbb{N}^2 \text{ avec } i + 2 + j + 1 \leq s\}$$

(le mot $b^i aab^j$ possède $i + 2 + j$ lettres). Dans le premier cas, le nombre vaut 0. Pour i fixé, il faut que

$$j \leq s - i - 3$$

donc cela fait $s - i - 2$ choix possibles pour j donc, si $s > 2$, le nombre de cas est

$$\sum_{i=0}^{s-2} (s - i - 2) = \sum_{k=s-i-2}^{s-2} k = \frac{(s-1)(s-2)}{2}.$$

Commentaires : Ne pas penser à traiter le cas $s \leq n$ (vous ne serez pas le ou la seul(e)). Question de difficulté standard si le candidat a fait l'effort de comprendre la notion introduite.

- Q4.** Soit $q_0 \rightarrow q_1 \rightarrow \dots \rightarrow q_n$ un calcul réussi au seuil s donc il que tous les $s+1$ états consécutifs rencontrent O tout en exigeant qu'il ne passe pas par O . La seule possibilité est que la condition (iii) soit impossible à réaliser c'est-à-dire il faut que $s > n$ i.e. $n \leq s - 1$. Dans ce cas, pour un entier $n \in \{1, \dots, s-1\}$ donné, chaque transition $q_i \rightarrow q_{i+1}$ peut prendre au plus $|\Sigma|$ valeurs donc le calcul $q_0 \rightarrow q_1 \rightarrow \dots \rightarrow q_n$ peut prendre au plus $|\Sigma|^n$ valeurs. Le nombre total de calculs réussi au seuil s est au plus égal à $\sum_{n=1}^{s-1} |\Sigma|^n$.

Commentaires : Comme toute question de dénombrement, elle est discriminante mais de niveau standard.

Q5. Il faut que toute famille de $s + 1$ états consécutifs rencontrent O . Dans le pire des cas, les i_0 premiers états $q_0, \dots, q_{i_0-1}$ ne rencontrent pas O (cela fait au plus s états consécutifs) donc $i_0 \leq s$.

Par définition, i_k est majoré par ℓ et il est minoré par $\ell - s$ (car il doit être dans un intervalle de longueur au plus $s + 1$ où l'on trouve au moins un élément de O).

L'écart entre i_k et i_{k+1} ne peut pas excéder $s + 1$ (tous les $s + 1$ états consécutifs doivent rencontrer O) c'est-à-dire $i_{k+1} - i_k \leq s + 1$.

Commentaires : Question de niveau standard .. si le candidat a compris la notion de calcul réussi (d'où l'importance d'avoir bien réfléchi aux premières questions du sujet).

Q6. A chaque état q de l'automate initial, on ajoute un compteur $c \in \{0, \dots, s\}$ (qui comptera le nombre d'états qui n'ont pas encore rencontré O). Autrement dit, les états du nouvel automate sont les états (q, c) avec $q \in Q$ et $c \in \{0, \dots, s\}$ donc il possède $p(s + 1)$ états. Chaque transition $q \xrightarrow{x} q'$ de l'automate initial sera transcrite dans le nouvel automate en incrémentant d'une unité le compteur si on part d'un état qui n'est pas dans O . Sinon on le réinitialise à 0 si cet état est dans O (le fameux " parmi $s + 1$ états consécutifs, au moins un est dans O). Précisons cela. À toute transition $q \xrightarrow{x} q'$ de l'automate initial, on associe les transitions :

- pour chaque $c \in \{0, \dots, s - 1\}$, $(q, c) \xrightarrow{x} (q', c + 1)$ si q n'appartient pas à O ;
- pour chaque $c \in \{0, \dots, s\}$, $(q, c) \xrightarrow{x} (q', 0)$ si q appartient à O

dans le nouvel automate. Les états initiaux du nouveau automate sont les états $(i, 0)$ avec i état initial de l'automate initial. Les états finaux sont les états (f, c) avec f état final de l'automate initial et $c \in \{0, \dots, s\}$. Un calcul $(q_0, c_0) \xrightarrow{u_1} (q_1, c_1) \xrightarrow{u_2} \dots \xrightarrow{u_n} (q_n, c_n)$ est réussi dans le nouvel automate si et seulement si :

- (q_0, c_0) est un état initial du nouvel automate i.e. q_0 est un état initial de l'automate initial et $c_0 = 0$;
- (q_n, c_n) est un état final du nouvel automate i.e. q_n est un état final ;
- chaque transition $(q_i, c_i) \xrightarrow{u_{i+1}} (q_{i+1}, c_{i+1})$ existe c'est-à-dire que chaque transition $q_i \xrightarrow{u_i} q_{i+1}$ existe dans l'automate initial et toute famille de $s + 1$ états consécutifs $q_0, \dots, q_n$ rencontrent O (les compteurs c_i ne peuvent jamais excéder strictement s donc ils ne peuvent jamais prendre la valeur $s + 1$ c'est-à-dire il ne peut y avoir $s + 2$ états consécutifs qui ne rencontrent pas O).

si et seulement si le calcul $q_0 \xrightarrow{u_1} q_1 \xrightarrow{u_2} \dots \xrightarrow{u_n} q_n$ c'est-à-dire que le langage reconnu par le nouvel automate est exactement les calculs réussis au seuil s dans l'automate initial.

Commentaires : Question très difficile pour ce concours. Félicitations aux candidat(e)s ayant donné une réponse convenable.

Q7.

```
def Vide(a):
    return a == None

def Racine(a):
```

```

    return a[0]

def Fils(a):
    return a[1]

```

Commentaires : Question élémentaire.

Q8.

```

def ArbreBinomial(a1, a2):
    r1 = Racine(a1)
    r2 = Racine(a2)
    f1 = Fils(a1)
    f2 = Fils(a2)
    if r1 <= r2:
        f1.append(a2)
        return [r1, f1]
    else:
        f2.append(a1)
        return [r2, f2]

```

Commentaires : Question élémentaire .. si le candidat a intégré correctement le codage en Python de ces arbres.

Q9. On procède par récurrence sur k . Si $k = 0$ alors l'arbre est nécessairement $a = (r, [])$ qui possède 0 fils. Supposons la propriété vérifiée pour tout entier strictement inférieur à k avec $k \geq 1$. Soit a un arbre binomial d'ordre k . Par définition, il est de la forme $a = (r, [t_0, \dots, t_{n-1}])$ avec t_{n-1} un arbre binomial d'ordre $k-1$ et $a' = (r, [t_0, \dots, t_{n-2}])$ un arbre binomial d'ordre $k-1$. Par hypothèse de récurrence a' possède $k-1$ fils et $t_0, \dots, t_{n-2}$ donc a possède k fils $t_0, \dots, t_{n-2}, t_{n-1}$. Ainsi, la propriété est vraie au rang k ce qui achève la récurrence.

Commentaires : Question de difficulté standard mais elle s'avère en pratique discriminante. Entraînez vous à démontrer des résultats sur les arbres (par récurrence sur le nombre de noeuds ou la hauteur selon). Ce type de preuve est rudimentaire et vous serez bien récompensés pour y répondre.

Q10.

```

def ordre(a):
    return len(Fils(a))

```

Commentaires : Question élémentaire.

Q11. On procède par récurrence sur k . Si $k = 0$ alors l'arbre est nécessairement $a = (r, [])$ qui possède $1 = 2^0$ noeuds. Supposons la propriété vérifiée pour tout entier strictement inférieur à k avec $k \geq 1$. Soit a un arbre binomial d'ordre k . Par définition, il est de la forme $a = (r, [t_0, \dots, t_{k-1}])$ (car il possède k fils d'après la question 9) avec t_{k-1} un arbre binomial d'ordre $k-1$ donc il possède 2^{k-1} noeuds (par hypothèse de récurrence) et $a' = (r, [t_0, \dots, t_{k-2}])$ un

arbre binomial d'ordre $k-1$ donc il possède 2^{k-1} noeuds (par hypothèse de récurrence). Par conséquent, a possède

$$2^{k-1} + 2^{k-1} = 2 \times 2^{k-1} = 2^k$$

noeuds (on ajoute ceux de a' à ceux de t_{k-1}). Ainsi, la propriété est vraie au rang k ce qui achève la récurrence.

Commentaires : *Même remarque qu'à la question 9.*

Q12.

```
def EstUnArbreBinomial(a):
    r = Racine(a)
    f = Fils(a)
    n = len(f)
    if n == 0 :
        return True
    else :
        f1=[f[i] for i in range(n-1)]
        a1=(r,f1)
        a2 = f[n-1]
        return EstUnArbreBinomial(a1) and EstUnArbreBinomial(
            a2) and Racine(a1) <= Racine(a2) and Ordre(a1) ==
            Ordre(a2)
```

Commentaires : *Question de difficulté standard. Il faut être attentif à n'oubliez aucun critère (notamment la comparaison des racines et l'ordre). L'un de ces oublis sera sanctionné mais deux oublis fera que vous n'aurez aucun point à cette question.*

Q13. On les code via des listes d'arbres.

Commentaires : *Question élémentaire.*

Q14. $T = \{a_0, \dots, a_k\}$ un tas binomial de longueur $k+1$. Par définition, son nombre de noeuds est la somme des noeuds des arbres le composant. Comme le nombre de noeuds de a_i est 0 si a_i est vide et 2^i sinon (d'après la question 11), ce nombre vaut $s_i 2^i$ justement donc

$$|T| = \sum_{i=0}^k |a_i| = \sum_{i=0}^k s_i 2^i.$$

Comme chaque s_i est inférieur à 1, on a

$$|T| \leq \sum_{i=0}^k 2^i = \frac{2^{k+1} - 1}{2 - 1} = 2^{k+1} - 1.$$

Comme $s_k = 1$ (par définition que T est un tas binomial de longueur $k+1$), on a $|T| \geq |a_k| = 2^k$ d'où

$$2^k \leq |T| < 2^{k+1} \Leftrightarrow k \leq \log_2(|T|) < k+1 \Leftrightarrow k = \lfloor \log_2(|T|) \rfloor.$$

Commentaires : *Question sélective qui demande au candidat une assez bonne compréhension des notions importante de l'informatique théorique (écriture binaire, arbre, visualisation de cette notion de signature).*

Q15. Une première version en parcourant tous les noeuds du tas

```
def minimum(h):
    k = len(h)
    assert (k>0)
    m = racine(h[k])
    for i in range(k):
        if not Vide(h[i]) :
            m = min(m,minimum (racine (h[i])))
    return m
```

Sa complexité est en $O(|T|)$ (car on parcourt tous les noeuds composant le tas, chaque opération ayant un coût constant).

Mais si on réfléchit un peu, le minimum est forcément l'une des racines des arbres binomiaux composant cet arbre (par une récurrence immédiate), ce qui diminue drastiquement la complexité car il suffit juste de comparer les racines de chaque arbre le composant.

```
# version analysant tous les noeuds
def minimum(h):
    k = len(h)
    assert (k>0)
    m = racine(h[k])
    for i in range(k):
        if not Vide(h[i]) :
            m = min(m,minimum (racine (h[i])))
    return m
```

Si k est la longueur de ce tas, alors $k = O(\log_2(|T|))$ (d'après la question 14) alors la complexité de la fonction précédente est en $O(k) = O(\log_2(|T|))$.

Commentaires : *Il s'agit presque d'une question de cours. Elle ne doit pas vous poser de problème particulier. Si c'est le cas, consolider vos connaissances sur les arbres.*

Q16.

```
def insertion(p,T):
    i = 0
    a = (p,[])
    while i < len(T) and not Vide(a):
        if Vide(T[i]):
            T[i] = a
            a = None
        else :
            a = ArbreBinomial(a,T[i])
            T[i] = None
            i = i + 1
    if not Vide(a):
        T.append(a)
```


Q21. Je redéfinis le type *fraction* en *frac* pour éviter la confusion avec la fonction *fraction* qui devra être implémentée dans une question ultérieure. Les arbres de Calkin-Wilf sont implémentés sous le nom de *cw*

```
type frac = {n:int; d:int};;
type cw = V | N of cw * frac * cw;;
```

Commentaires : Question de cours qui ne doit pas vous poser de problème. Si c'est le cas, revoir votre cours pour la construction des structures du programme (notamment arbres et listes).

Q22. On procède par récurrence sur le niveau d'exploration de l'arbre. Si on est au niveau 0, la fraction considérée est $\frac{1}{1}$ donc $n = d = 1$ sont premiers entre eux. Supposons la propriété vérifiée à un certain niveau m d'exploration. Soit $\frac{n}{d}$ la fraction de l'un des noeuds de ce niveau. Par hypothèse, n et d sont premiers entre eux. Ses fils sont $\frac{n}{n+d}$ et $\frac{n+d}{d}$. Puisque le pgcd de n et $n+d$ est celui de n et $(n+d) - n = d$, il vaut 1 donc n et $n+d$ sont premiers entre eux. De même, le pgcd de $n+d$ et d est celui de $(n+d) - d = n$, il vaut 1 donc $n+d$ et d sont premiers entre eux. Ainsi, les deux fils de $\frac{n}{d}$ sont bien écrits sous la forme $\frac{u}{v}$ avec u et v premiers entre eux. Ceci est alors le cas pour tous les noeuds du niveau $m+1$ ce qui achève la récurrence.

Commentaires : Question difficile car il s'agit d'une question de pur mathématiques (arithmétique) et les candidats n'aiment pas cela en pratique. Au final, seules les très bonnes copies fournissent une réponse adéquate.

Q23.

```
let rec pgcd a b = if b == 0 then a else pgcd b (a mod b);;
```

Commentaires : Question de cours sans aucune difficulté.

Q24.

```
let fraction n d = match n,d with
| n,d when n <= 0 || d <= 0 -> failwith " les entiers doivent
être positifs "
| n,d -> let c = pgcd n d in {n=n/c;d=d/c};;
```

Commentaires : Question de cours sans aucune difficulté mais il faut maîtriser (un peu) la notion d'enregistrements (si ce n'est pas le cas, travaillez la car elle intervient dans de nombreux sujets, notamment avec les automates) et pensez aux distinctions de cas.

Q25. Supposons que $\frac{k}{l}$ et $\frac{n}{d}$ ont même fils gauche alors

$$\begin{aligned}\frac{k}{k+l} &= \frac{n}{d+n} \Leftrightarrow k(d+n) = n(k+l) \Leftrightarrow kd + kn = kn + nl \\ &\Leftrightarrow kd = nl \Leftrightarrow \frac{k}{l} = \frac{n}{d}\end{aligned}$$

donc $\frac{k}{k} = \frac{n}{d}$. De même, s'ils ont même fils droit, on a :

$$\begin{aligned}\frac{k+l}{l} &= \frac{n+d}{d} \Leftrightarrow (k+l)d = l(n+d) \Leftrightarrow kd + ld = nl + ld \\ &\Leftrightarrow kd = nl \Leftrightarrow \frac{k}{l} = \frac{n}{d}.\end{aligned}$$

Commentaires : Question difficile car il s'agit d'une question de pur mathématiques et les candidats n'aiment pas cela en pratique. Au final, seules les meilleures copies fournissent une réponse adéquate.

Q26. Le fils gauche d'une fraction $N = \frac{n}{d}$ consiste à ajouter à son dénominateur une fois son numérateur et le numérateur reste inchangé. En itérant cela k fois, le numérateur est inchangé et le dénominateur a été ajouté k de son numérateur donc il s'agit de la fraction

$$\frac{n}{d+kn} = \frac{n}{d\left(1+k\frac{n}{d}\right)} = \frac{\frac{n}{d}}{1+k\frac{n}{d}} = \frac{N}{1+kN}.$$

Pour le fils droit, le dénominateur reste inchangé alors que le numérateur est additionné une fois du dénominateur. En itérant k fois, le dénominateur est inchangé et le numérateur est additionné k fois du numérateur initial donc il s'agit de la fraction

$$\frac{n+kd}{d} = \frac{d}{d} \left(\frac{n}{d} + k \right) = N + k.$$

Commentaires : Si la première question est sélective car elle demande une preuve et non une observation, la seconde est aisée si vous observez quelques profondeurs consécutives de l'arbre. Vu que seule la bonne réponse est payée, vous devez savoir la deviner (d'où l'importance de la première question sur ce thème, c'est-à-dire la question 21).

Q27. $0, \frac{1}{1}, \frac{1}{2}, \frac{2}{1}, \frac{1}{3}, \frac{3}{2}, \frac{2}{3}, \frac{3}{1}.$

Commentaires : Question élémentaire.

Q28. Procédons par récurrence sur $n+d$. Sa plus petite valeur est obtenue pour $n=1$ et $d=1$. Comme $\frac{n}{d} = \frac{1}{1}$ apparaît bien dans l'arbre, la propriété est vérifiée au rang $n+d=2$. Supposons qu'elle soit vraie pour toute valeur de $n+d < k$ avec k un entier ≥ 3 . Considérons

une fraction $\frac{n}{d}$ avec $n + d = k$. Si $n < d$ alors $(n, d - n)$ est un couple d'entiers non nuls donc la somme vaut

$$n + (d - n) = d < n + d = k.$$

Par hypothèse de récurrence, la fraction $\frac{n}{d - n}$ apparaît dans l'arbre donc son fils gauche $\frac{n}{(d - n) + n} = \frac{n}{d}$ apparaît dans l'arbre. Si $n > d$ alors $(n - d, d)$ est un couple d'entiers non nuls donc la somme vaut

$$(n - d) + d = n < n + d = k.$$

Par hypothèse de récurrence, la fraction $\frac{n - d}{d}$ apparaît dans l'arbre donc son fils droit $\frac{(n - d) + d}{d} = \frac{n}{d}$ apparaît dans l'arbre. Le cas $n = d$ est impossible (car n et d sont premiers entre eux). Par conséquent, la propriété est vérifiée lorsque $n + d = k$ ce qui achève la récurrence.

Commentaires : Question difficile destinée en pratique aux meilleures copies.

Q29. Remarquons que tout noeud de l'arbre hormis la racine correspond à une fraction strictement inférieure à 1 (car elle est un fils gauche d'un noeud correspondant à une fraction $\frac{n}{d}$ donc elle est de la forme $\frac{n}{d + n} < 1$) ou strictement supérieure à 1 (car elle est un fils droit d'un noeud correspondant à une fraction $\frac{n}{d}$ donc elle est de la forme $\frac{n + d}{d} > 1$).

En particulier, on en déduit que la fraction $\frac{1}{1}$ n'apparaît qu'une seule fois dans l'arbre.

Considérons une fraction $\frac{n}{d}$ autre que 1. Supposons qu'il existe deux noeuds distincts f_1 et f_2 correspondant à cette fraction $\frac{n}{d}$. Si $\frac{n}{d} < 1$ (resp. $\frac{n}{d} > 1$) alors ces deux noeuds sont des fils gauche (resp. droit) de noeuds de l'arbre. D'après la question 25, leurs parents f'_1 et f'_2 correspondent à la même valeur qui vaut toujours $\frac{n}{d}$. On itère le procédé tant que les parents $f_1^{(i)}$ et $f_2^{(i)}$ existent pour les deux noeuds (après i itérations). Ce procédé s'arrête nécessairement (ces deux noeuds initiaux ont une profondeur finie qui majore le nombre d'itérations). Par conséquent, l'un des noeuds n'a pas de parent. Or, le seul noeud de l'arbre sans parent est la racine donc cette fraction est $\frac{1}{1}$, ce qui est absurde car $\frac{n}{d}$ est différent de 1. Par conséquent, un seul noeud de l'arbre correspond à $\frac{n}{d}$.

Commentaires : Même remarque qu'à la question précédente.

Q30. La question précédente montre l'injectivité de $i \in \mathbb{N} \mapsto v_i \in \mathbb{Q}_+$ (puisque l'on a ajouté artificiellement 0 en dehors de l'arbre et qu'il n'apparaît pas dans l'arbre) et la question 28 sa surjectivité d'où sa bijectivité.

Commentaires : Question de synthèse ... pour un candidat comprenant la notion de bijection donc elle est sélective.

Q31. Soit i un entier, v_i est la valeur du i^{e} noeud N de l'arbre dans son parcours en largeur et notons k sa profondeur. L'arbre est binaire complet donc, la profondeur k contient 2^k noeuds et son noeud le plus à gauche a pour numéro 2^k . Par conséquent, il existe $i - 2^k$ à gauche

du noeud N à la profondeur k et $2^{k+1} - i - 1$ noeuds à sa droite. Pour atteindre son fils gauche, il faudra parcourir tous les noeuds à droite numéro i puis tous les fils de ceux qui sont à sa gauche puis faire un pas de plus pour accéder au fils gauche. Ainsi, le numéro de son fils gauche aura pour numéro

$$i + (2^{k+1} - i - 1) + 2(i - 2^k) + 1 = 2^{k+1} + 2i - 2^{k+1} = 2i$$

qui correspond à la fraction v_{2i} . Son fils droite aura pour numéro $2i + 1$ qui correspond à la fraction v_{2i+1} .

Commentaires : Si la réponse est standard (dans le cours notamment concernant la structure de tas), elle est difficile pour de nombreux candidats car il s'agit d'un comptage précis et subtil.

Q32. 0, 1, 1, 2, 1, 3, 2, 3, 1, 4.

Commentaires : Question élémentaire.

Q33.

```
let rec stern n = match n with
| 0 -> 0 ;
| 1 -> 1 ;
| n when n mod 2 == 0 -> stern (n/2);
| n -> let i = n/2 in stern i + stern (i+1) ;;
```

Commentaires : Question élémentaire de programmation. Elle ne doit pas vous poser problème.

Q34. On procède par récurrence. Comme $s_0 = 0$, $s_1 = 1$, on a bien $\frac{s_0}{s_1} = 0 = v_0$ donc la propriété est vraie pour $i = 0$. Supposons la propriété vraie pour tout entier $j < i$ pour un certain entier $i \geq 1$. Si i est pair, il s'écrit $i = 2j$ avec $0 < j < i$ donc $v_i = v_{2j}$ est le fils gauche de v_j . Par hypothèse de récurrence, on a $v_j = \frac{s_j}{s_{j+1}}$. Son fils gauche v_{2j} est alors défini par :

$$v_i = v_{2j} = \frac{s_j}{s_j + s_{j+1}} = \frac{s_{2j}}{s_{2j+1}} = \frac{s_i}{s_{i+1}}.$$

Si i est impair, il s'écrit $i = 2j + 1$ avec $j < i$ donc $v_i = v_{2j+1}$ est le fils droit de v_j . Par hypothèse de récurrence, on a $v_j = \frac{s_j}{s_{j+1}}$. Son fils droit v_{2j+1} est alors défini par :

$$v_i = v_{2j+1} = \frac{s_j + s_{j+1}}{s_{j+1}} = \frac{s_{2j+1}}{s_{2(j+1)}} = \frac{s_{2j+1}}{s_{(2j+1)+1}} = \frac{s_i}{s_{i+1}}.$$

Ainsi, la propriété est vérifiée au rang i ce qui achève la récurrence.

Commentaires : Question de difficulté standard même si elle apparaît en pratique très sélective.

Q35. Comme v_i est le fils gauche de N , on a $v_i < 1$ donc $\lfloor v_i \rfloor = 0$ d'où $f(v_i) = \frac{1}{1 - v_i}$. Si on note $N = \frac{n}{d}$ alors $v_i = \frac{n}{d+n}$ donc

$$f(v_i) = \frac{1}{1 - v_i} = \frac{1}{1 - \frac{n}{d+n}} = \frac{1}{\frac{d+n-n}{d+n}} = \frac{d+n}{d}$$

est le fils droit de N donc $f(v_i) = v_{i+1}$.

Commentaires : *Même remarque qu'à la question précédente.*

Q36. Dans le deuxième cas, v_i est le noeud à droite à la profondeur k donc $v_i = 1 + k$ (cf. question 26 ou conjecture graphique) donc

$$\lfloor v_i \rfloor = v_i = 1 + k \Rightarrow f(v_i) = \frac{1}{1 + 2 \lfloor v_i \rfloor - v_i} = \frac{1}{1 + v_i}.$$

Dans ce cas, v_{i+1} est le noeud à gauche à la profondeur $k + 1$ donc $v_{i+1} = \frac{1}{1 + (k + 1)}$ (cf. question 26 ou par conjecture graphique) donc

$$v_{i+1} = \frac{1}{1 + v_i} = f(v_i).$$

Commentaires : *Même remarque qu'à la question précédente.*

Q37.

```
let chemin f =
  let rec aux f = match f.n, f.d with
    | 1,1 -> []
    | n,d when n < d -> G::(aux {n=n;d=d-n})
    | n,d -> D::(aux {n=n-d;d=d})
  in List.rev (aux f);;
```

Commentaires : *Le candidat doit bien avoir compris comment se fait la navigation dans l'arbre et, surtout de comprendre que le choix du fils à prendre dépend de la position de la fraction par rapport à 1 (ce qui se voit sur les premiers exemples, notamment la question 21. La sélection des candidats à cette question ne sera pas au niveau de la programmation (qui est relativement classique) mais sur cette capacité à avoir intégré le sens de la navigation dans l'arbre. Cette question est finalement réussie par les bonnes copies (sans être excellentes).*

Q38.

```
let noeud l =
  let rec aux l acc =
    match l with
    | [] -> acc
```

```

|G::q -> aux q {n = acc.n ; d = acc.n + acc.d}
|D::q -> aux q {n = acc.n + acc.d ; d = acc.d}
in aux 1 {n=1;d=1};;

```

Commentaires : *Même remarque de précédemment.*

Q39.

```

let ancetre f1 f2 =
  let rec identique l1 l2 = match l1,l2 with
    |a::q,b::r when a == b -> a::(identique q r)
    |_,_ -> []
  in let l1 = chemin f1 and l2 = chemin f2 in
  noeud (identique l1 l2);;

```

Commentaires : *Question destinée aux meilleures copies (ayant une bonne maîtrise de la programmation et de la navigation dans ce type d'arbre).*

Q40. Comme v_p est l'ancêtre commun de v_i et v_{i+1} , pour accéder à v_i , il faut commencer par un déplacement à gauche (fils gauche) et pour accéder à v_{i+1} , il faut commencer par un déplacement à droite (fils droit). Ensuite, il est indispensable que les descendants se rapprochent de plus en plus donc il faut toujours aller à droit pour accéder à v_i et aller à gauche pour atteindre v_{i+1} . Autrement dit, pour aboutir à v_i depuis v_p , il faut effectuer le mouvement $\underbrace{GD \cdots D}_{k-1 \text{ fois}}$ et pour aboutir à v_{i+1} depuis v_p , il faut effectuer le mouvement $\underbrace{DG \cdots G}_{k-1 \text{ fois}}$.

Commentaires : *Question destinée aux meilleures copies. Bravo à eux et à elles.*

Q41. Le fils gauche de v_p a pour valeur $\frac{v_p}{1+v_p} = N$ donc son $k' - 1$ ième fils droit consécutif vaut

$$v_i = N + (k' - 1) = \frac{v_p}{1+v_p} + (k' - 1)$$

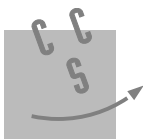
donc, comme $N < 1$, on a :

$$\begin{aligned}
[v_i] &= k' - 1 \Rightarrow f(v_i) = \frac{1}{1-v_i} = \frac{1}{1 - \frac{v_p}{1+v_p} + k' - 1} \\
&= \frac{1}{k' - \frac{v_p}{1+v_p}} = \frac{1+v_p}{k'(1+v_p) - v_p} = \frac{1+v_p}{k' + (k-1)v_p}.
\end{aligned}$$

Le fils droit de v_p a pour valeur $v_p + 1 = N'$ donc son $k' - 1$ ième fils gauche consécutif vaut

$$v_{i+1} = \frac{N'}{1 + (k' - 1)N'} = \frac{v_p + 1}{1 + (k' - 1)(v_p + 1)} = f(v_i).$$

Commentaires : *Question destinée aux meilleures copies .. s'ils ont encore la force et le temps d'arriver là :-).*



Mathématiques 1

MP

2022

CONCOURS CENTRALE-SUPÉLEC

4 heures

Calculatrice autorisée

Exemples de contraintes symplectiques linéaires

Notations

- Dans tout le problème, n et m désignent des entiers naturels non nuls.
- On note $\mathcal{M}_n(\mathbb{R})$ l'ensemble des matrices carrées de taille $n \times n$ à coefficients réels et I_n la matrice identité dans $\mathcal{M}_n(\mathbb{R})$.
- Pour tous entiers naturels non nuls p et q , on note $\mathcal{M}_{p,q}(\mathbb{R})$ l'ensemble des matrices à p lignes et q colonnes à coefficients réels. Ainsi, $\mathcal{M}_{n,1}(\mathbb{R})$ est l'ensemble des matrices colonnes à n lignes et $\mathcal{M}_{n,n}(\mathbb{R}) = \mathcal{M}_n(\mathbb{R})$.
- On note $\mathcal{S}_n(\mathbb{R})$ l'ensemble des matrices réelles symétriques de $\mathcal{M}_n(\mathbb{R})$.
- On note $\text{GL}_n(\mathbb{R})$ le groupe linéaire réel d'ordre n (matrices carrées inversibles dans $\mathcal{M}_n(\mathbb{R})$) et $\text{SL}_n(\mathbb{R})$ le sous-groupe des matrices de déterminant égal à 1 :

$$\text{SL}_n(\mathbb{R}) = \{A \in \mathcal{M}_n(\mathbb{R}) \mid \det(A) = 1\}.$$

- On note $A^\top$ la transposée d'une matrice A .
- Le produit scalaire canonique de $\mathbb{R}^n$ et la norme euclidienne associée sont notés respectivement $\langle \cdot, \cdot \rangle$ et $\|\cdot\|$.
- Le groupe orthogonal réel d'ordre n est noté $\text{O}_n(\mathbb{R})$:

$$\text{O}_n(\mathbb{R}) = \{A \in \mathcal{M}_n(\mathbb{R}) \mid A^\top A = I_n\}.$$

- Si E est un $\mathbb{R}$ -espace vectoriel, on note $\mathcal{L}(E)$ l'ensemble des endomorphismes de E et $\mathcal{L}(E, \mathbb{R})$ l'ensemble des applications linéaires de E dans $\mathbb{R}$.

Objectif

Ce problème a pour objectif de définir la notion d'espace symplectique réel et d'étudier certaines propriétés des endomorphismes symplectiques de $\mathbb{R}^n$.

La première partie établit quelques résultats utiles dans la suite.

La deuxième partie définit toutes les notions relatives aux objets symplectiques utilisés dans la suite du problème.

La troisième partie vise plus spécifiquement à montrer que toute matrice symplectique réelle a un déterminant égal à 1.

La quatrième partie aborde la définition des contraintes permettant d'injecter un objet dans un autre au moyen d'un endomorphisme symplectique.

Les deux dernières parties sont largement indépendantes.

I Préliminaires

- Q 1.** Soient A et B deux matrices de $\mathcal{M}_n(\mathbb{R})$ telles que

$$\forall (X, Y) \in (\mathcal{M}_{n,1}(\mathbb{R}))^2, \quad X^\top AY = X^\top BY.$$

Montrer que $A = B$.

- Q 2.** Soit $M \in \text{GL}_n(\mathbb{R})$. Montrer que les valeurs propres de $M^\top M$ sont toutes strictement positives. En déduire qu'il existe une matrice S symétrique à valeurs propres strictement positives telle que $S^2 = M^\top M$.

II Objets symplectiques

II.A – Structure d'espace vectoriel symplectique réel

Soit E un $\mathbb{R}$ -espace vectoriel de dimension finie n .

On appelle *forme symplectique sur E* toute application ω de E^2 dans $\mathbb{R}$ qui vérifie les trois propriétés suivantes :

- bilinéarité : $\forall (x, y, z) \in E^3, \forall \lambda \in \mathbb{R}, \omega(x + \lambda y, z) = \omega(x, z) + \lambda \omega(y, z)$ et $\omega(x, y + \lambda z) = \omega(x, y) + \lambda \omega(x, z)$;
- antisymétrie : $\forall (x, y) \in E^2, \omega(x, y) = -\omega(y, x)$;
- non dégénérescence : $\{x \in E \mid \forall y \in E, \omega(x, y) = 0\} = \{0_E\}$.

Un *espace vectoriel symplectique réel* (E, ω) est un $\mathbb{R}$ -espace vectoriel de dimension finie E muni d'une forme symplectique ω sur E .

Q 3. Montrer que, si ω est une forme symplectique sur E , alors pour tout vecteur x de E , $\omega(x, x) = 0$.

Pour tout sous-espace vectoriel F d'un espace symplectique (E, ω) , on appelle ω -orthogonal de F et on note F^ω l'ensemble

$$F^\omega = \{x \in E \mid \forall y \in F, \omega(x, y) = 0\}.$$

Soit F un sous-espace vectoriel d'un espace symplectique (E, ω) .

Q 4. Justifier que F^ω est un sous-espace vectoriel de E .

Q 5. Le sous-espace F^ω est-il nécessairement en somme directe avec F ?

Pour tout $x \in E$, on note $\omega(x, \cdot)$ l'application linéaire de E dans $\mathbb{R}$, $y \mapsto \omega(x, y)$ et on considère

$$d_\omega : \begin{cases} E & \rightarrow \mathcal{L}(E, \mathbb{R}) \\ x & \mapsto \omega(x, \cdot) \end{cases}$$

Q 6. Montrer que d_ω est un isomorphisme.

Pour $\ell \in \mathcal{L}(E, \mathbb{R})$, on note $\ell|_F$ la restriction de ℓ à F .

Q 7. Montrer que l'application de restriction $r_F : \begin{cases} \mathcal{L}(E, \mathbb{R}) & \rightarrow \mathcal{L}(F, \mathbb{R}) \\ \ell & \mapsto \ell|_F \end{cases}$ est surjective.

Q 8. Préciser le noyau de $r_F \circ d_\omega$. En déduire que $\dim F^\omega = \dim E - \dim F$.

Q 9. Montrer que la restriction ω_F de ω à F^ω définit une forme symplectique sur F si et seulement si $F \oplus F^\omega = E$.

II.B – Structure symplectique standard sur $\mathbb{R}^n$

On suppose qu'il existe une forme symplectique ω sur $\mathbb{R}^n$ et on note $\Omega \in \mathcal{M}_n(\mathbb{R})$ la matrice définie par

$$\Omega = (\omega(e_i, e_j))_{1 \leq i, j \leq n}$$

où $(e_1, \dots, e_n)$ désigne la base canonique de $\mathbb{R}^n$.

Q 10. Montrer que

$$\forall (x, y) \in \mathbb{R}^n \times \mathbb{R}^n, \quad \omega(x, y) = X^\top \Omega Y$$

où X et Y désignent les colonnes des coordonnées de x et y dans la base canonique de $\mathbb{R}^n$.

Q 11. En déduire que Ω est antisymétrique et inversible.

Q 12. Conclure que l'entier n est pair.

Jusqu'à la fin du problème, on suppose que n est pair et on note $m \in \mathbb{N}^*$ l'entier naturel tel que $n = 2m$.

On note $J \in \mathcal{M}_n(\mathbb{R}) (= \mathcal{M}_{2m}(\mathbb{R}))$ la matrice définie par blocs par

$$J = \begin{pmatrix} 0 & -I_m \\ I_m & 0 \end{pmatrix}$$

et on note j l'endomorphisme de $\mathbb{R}^n$ canoniquement associé à J .

Q 13. Montrer que l'application $b_s : \begin{cases} \mathbb{R}^n \times \mathbb{R}^n & \rightarrow \mathbb{R} \\ (x, y) & \mapsto \langle x, j(y) \rangle \end{cases}$ est une forme symplectique sur $\mathbb{R}^n$.

Il existe donc des formes symplectiques en dimension paire, et seulement en dimension paire.

La forme symplectique b_s est appelée la *forme symplectique standard sur $\mathbb{R}^n$* .

II.C – Endomorphismes et matrices symplectiques réels

On appelle *endomorphisme symplectique* d'un espace vectoriel symplectique réel (E, ω) tout endomorphisme $u \in \mathcal{L}(E)$ tel que

$$\forall (x, y) \in E^2, \quad \omega(u(x), u(y)) = \omega(x, y).$$

On note $\text{Symp}_\omega(E)$ l'ensemble des endomorphismes symplectiques de l'espace symplectique (E, ω) .

Soit $u \in \text{Symp}_\omega(E)$ un endomorphisme symplectique de E .

Soient λ, μ des valeurs propres réelles de u , et soient $E_\lambda(u)$, $E_\mu(u)$ les sous-espaces propres associés.

Q 14. Montrer que, si $\lambda\mu \neq 1$, alors les sous-espaces $E_\lambda(u)$ et $E_\mu(u)$ sont ω -orthogonaux, c'est-à-dire :

$$\forall x \in E_\lambda(u), \quad \forall y \in E_\mu(u), \quad \omega(x, y) = 0.$$

Soit $u \in \mathcal{L}(\mathbb{R}^n)$ un endomorphisme de $\mathbb{R}^n$. On note M la matrice de u dans la base canonique de $\mathbb{R}^n$.

Q 15. Montrer que u est un endomorphisme symplectique de l'espace symplectique standard $(\mathbb{R}^n, b_s)$ si et seulement si $M^\top JM = J$.

Une matrice $M \in \mathcal{M}_n(\mathbb{R})$ est dite *symplectique* si $M^\top JM = J$.

On note $\mathrm{Sp}_n(\mathbb{R})$ l'ensemble des matrices symplectiques réelles dans $\mathcal{M}_n(\mathbb{R}) = \mathcal{M}_{2m}(\mathbb{R})$:

$$\mathrm{Sp}_n(\mathbb{R}) = \mathrm{Sp}_{2m}(\mathbb{R}) = \{M \in \mathcal{M}_n(\mathbb{R}) \mid M^\top JM = J\}.$$

Q 16. Montrer que $\mathrm{Sp}_n(\mathbb{R})$ est un sous-groupe de $\mathrm{GL}_n(\mathbb{R})$, stable par transposition et contenant la matrice J . Ce groupe est appelé *groupe symplectique réel d'ordre $n = 2m$* .

Soient A, B, C, D dans $\mathcal{M}_m(\mathbb{R})$ et soit $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix} \in \mathcal{M}_{2m}(\mathbb{R})$ (décomposition par blocs).

Q 17. Montrer que $M \in \mathrm{Sp}_{2m}(\mathbb{R})$ si et seulement si

$$A^\top C \text{ et } B^\top D \text{ sont symétriques} \quad \text{et} \quad A^\top D - C^\top B = I_m.$$

III Déterminant d'une matrice symplectique réelle

L'objectif de cette partie est de montrer l'inclusion $\mathrm{Sp}_n(\mathbb{R}) \subset \mathrm{SL}_n(\mathbb{R})$ par deux méthodes différentes qui reposent chacune sur une propriété structurelle du groupe symplectique $\mathrm{Sp}_n(\mathbb{R})$ qu'on examine au préalable.

III.A – Le cas de la dimension 2

Q 18. Montrer que $\mathrm{Sp}_2(\mathbb{R}) = \mathrm{SL}_2(\mathbb{R})$.

III.B – Commutant de J

On note $\mathcal{C}_J = \{M \in \mathcal{M}_n(\mathbb{R}) \mid JM = MJ\}$ le commutant de la matrice J , c'est-à-dire l'ensemble des matrices de $\mathcal{M}_n(\mathbb{R})$ qui commutent avec J .

Q 19. Montrer que, pour toute matrice $M \in \mathcal{M}_n(\mathbb{R}) (= \mathcal{M}_{2m}(\mathbb{R}))$,

$$M \in \mathcal{C}_J \iff \exists (U, V) \in \mathcal{M}_m(\mathbb{R}) \times \mathcal{M}_m(\mathbb{R}), \quad M = \begin{pmatrix} U & -V \\ V & U \end{pmatrix}.$$

Q 20. En déduire que, pour toute matrice $M \in \mathcal{C}_J$, $\det(M) \geq 0$.

$$\text{On pourra considérer le produit de matrices par blocs } \begin{pmatrix} I_m & 0 \\ iI_m & I_m \end{pmatrix} \begin{pmatrix} U & -V \\ V & U \end{pmatrix} \begin{pmatrix} I_m & 0 \\ -iI_m & I_m \end{pmatrix}.$$

III.C – Décomposition polaire d'une matrice symplectique réelle

On note $\mathrm{OSp}_n(\mathbb{R}) = \mathrm{Sp}_n(\mathbb{R}) \cap \mathrm{O}_n(\mathbb{R})$ l'ensemble des matrices symplectiques et orthogonales réelles de $\mathcal{M}_n(\mathbb{R})$.

On munit $\mathcal{M}_n(\mathbb{R})$ de sa topologie d'espace vectoriel normé de dimension finie.

Q 21. Montrer que $\mathrm{OSp}_n(\mathbb{R})$ est un sous-groupe compact du groupe symplectique $\mathrm{Sp}_n(\mathbb{R})$.

Q 22. Montrer que $\mathrm{OSp}_n(\mathbb{R}) \subset \mathcal{C}_J$.

Q 23. En déduire que, pour toute matrice M de $\mathrm{OSp}_n(\mathbb{R})$, $\det(M) = 1$.

Jusqu'à la fin de la sous-partie III.C, on considère une matrice $M \in \mathrm{Sp}_n(\mathbb{R})$.

Soit $S \in \mathcal{S}_n(\mathbb{R})$ une matrice symétrique à valeurs propres strictement positives telle que $S^2 = M^\top M$.

Q 24. Montrer que S est symplectique.

On pourra considérer une base de vecteurs propres de l'endomorphisme s de $\mathbb{R}^n$ canoniquement associé à S , et montrer que s est un endomorphisme symplectique de l'espace standard $(\mathbb{R}^n, b_s)$.

Q 25. Justifier que S est inversible puis montrer que la matrice O définie par $O = MS^{-1}$ appartient au groupe $\mathrm{OSp}_n(\mathbb{R})$.

Q 26. Conclure que le déterminant de la matrice M est égal à 1.

III.D – Génération du groupe symplectique par les transvections symplectiques

Soit (E, ω) un espace vectoriel symplectique de dimension $n = 2m$.

On appelle *transvection* de E tout endomorphisme τ de E tel qu'il existe $\ell \in \mathcal{L}(E, \mathbb{R})$ et $a \in \ker(\ell)$ vérifiant

$$\forall x \in E, \quad \tau(x) = x + \ell(x)a.$$

III.D.1) Transvection symplectique

Q 27. Soit $a \in E$ un vecteur non nul et $\lambda \in \mathbb{R}$ un réel. Montrer que l'application τ_a^λ définie par

$$\forall x \in E, \quad \tau_a^\lambda(x) = x + \lambda \omega(a, x) a$$

est une transvection de E et qu'il s'agit d'un endomorphisme symplectique de ce même espace.

Les applications τ_a^λ pour $a \in E \setminus \{0_E\}$ et $\lambda \in \mathbb{R}$ sont appelées *transvections symplectiques de E* .

Q 28. Soit $a \in E$ un vecteur non nul et soient λ et μ des réels. Montrer que $\tau_a^\mu \circ \tau_a^\lambda = \tau_a^{\lambda+\mu}$.

Q 29. Soient $a \in E$ un vecteur non nul et λ un réel. Montrer que $\det(\tau_a^\lambda) > 0$.

Q 30. La réciproque $(\tau_a^\lambda)^{-1}$ est-elle encore une transvection symplectique ?

On se propose de montrer le théorème suivant :

Tout endomorphisme symplectique de E peut s'écrire comme la composée d'au plus $2n = 4m$ transvections symplectiques de E :
si $u \in \text{Symp}_\omega(E)$, il existe un entier $p \leq 4m$ et $\tau_1, \tau_2, \dots, \tau_p$ des transvections symplectiques de E telles que $u = \tau_p \circ \dots \circ \tau_2 \circ \tau_1$.

III.D.2) Un lemme

On commence par montrer le lemme suivant :

Pour tous vecteurs non nuls x et y de E , il existe une composée γ d'au plus deux transvections symplectiques de E telle que $\gamma(x) = y$.

On fixe x et y , non nuls, dans E .

Q 31. Supposons que $\omega(x, y) \neq 0$. Montrer qu'il existe $\lambda \in \mathbb{R}$ tel que $\tau_{y-x}^\lambda(x) = y$.

Q 32. Supposons que $\omega(x, y) = 0$. Montrer qu'il existe un vecteur $z \in E$ tel que $\omega(x, z) \neq 0$ et $\omega(y, z) \neq 0$.

Q 33. Montrer le lemme cité ci-dessus.

III.D.3) Le théorème

Soit $u \in \text{Symp}_\omega(E)$ un endomorphisme symplectique de E .

Soit $e_1 \in E$ un vecteur non nul.

Q 34. Justifier l'existence de $f_1 \in E$, non colinéaire à e_1 , tel que $\omega(e_1, f_1) = 1$.

On pose $P = \text{Vect}(e_1, f_1)$ le plan vectoriel engendré par les vecteurs e_1 et f_1 . On va montrer l'existence d'une composée δ d'au plus quatre transvections symplectiques de E telle que

$$\begin{cases} \delta(u(e_1)) = e_1 \\ \delta(u(f_1)) = f_1 \end{cases} \quad (\text{III.1})$$

Q 35. Pourquoi existe-t-il une composée δ_1 d'au plus deux transvections symplectiques de E telle que $\delta_1(u(e_1)) = e_1$?

Q 36. Notons $\tilde{f}_1$ le vecteur $\delta_1(u(f_1))$. Montrer qu'il existe une composée δ_2 d'au plus deux transvections symplectiques de E telle que

$$\begin{cases} \delta_2(e_1) = e_1 \\ \delta_2(\tilde{f}_1) = f_1 \end{cases}$$

On pourra adapter la démonstration du lemme précédent.

La composée $\delta = \delta_2 \circ \delta_1$ d'au plus quatre transvections symplectiques vérifie bien les conditions (III.1) souhaitées.

On pose $v = \delta \circ u$.

Q 37. Montrer que P est stable par v et déterminer v_P , endomorphisme induit par v sur P .

Q 38. Montrer que P^ω est stable par v .

Q 39. Montrer que la restriction ω_{P^ω} de ω à $P^\omega \times P^\omega$ munit P^ω d'une structure d'espace symplectique et que l'endomorphisme v_{P^ω} induit par v sur P^ω est un endomorphisme symplectique.

Q 40. À l'aide de ce qui précède, montrer le théorème annoncé.

III.D.4) Une conséquence topologique

On munit toujours l'espace $\mathcal{M}_n(\mathbb{R})$ de sa topologie d'espace vectoriel normé.

Q 41. Montrer que le groupe symplectique $\text{Sp}_n(\mathbb{R})$ est une partie connexe par arcs de cet espace.

III.D.5) Deuxième conséquence

Q 42. Utiliser les résultats de cette sous-partie III.D pour prouver l'inclusion $\text{Sp}_n(\mathbb{R}) \subset \text{SL}_n(\mathbb{R})$.

IV Exemples de problèmes de plongements symplectiques linéaires

Dans cette partie, on fixe $n = 2m \geq 4$.

On rappelle que $\text{Symp}_{b_s}(\mathbb{R}^{2m})$ désigne le groupe des endomorphismes symplectiques de l'espace symplectique standard $(\mathbb{R}^{2m}, b_s)$. On note de plus $\text{SL}(\mathbb{R}^{2m})$ le groupe des endomorphismes de $\mathbb{R}^{2m}$ de déterminant égal à 1. Pour $r \geq 0$, on considère les parties suivantes de $\mathbb{R}^{2m}$.

— La boule euclidienne fermée de rayon r :

$$B^{2m}(r) = \{(x_1, \dots, x_m, y_1, \dots, y_m) \in \mathbb{R}^{2m}, \quad x_1^2 + \dots + x_m^2 + y_1^2 + \dots + y_m^2 \leq r^2\}.$$

— Le cylindre symplectique de rayon r basé sur les axes de coordonnées numéro 1 et $m+1$:

$$Z^{2m}(r) = \{(x_1, \dots, x_m, y_1, \dots, y_m) \in \mathbb{R}^{2m}, \quad x_1^2 + y_1^2 \leq r^2\}.$$

Sur des exemples (boules ou cylindres) de parties A et B de $\mathbb{R}^{2m}$, on étudie l'existence d'un endomorphisme u tel que $u(A) \subset B$ lorsque $u \in \text{SL}(\mathbb{R}^{2m})$, puis lorsque $u \in \text{Symp}_{b_s}(\mathbb{R}^{2m})$.

IV.A – Injection par $u \in \text{SL}(\mathbb{R}^{2m})$ d'une boule dans un cylindre

Q 43. Montrer que, pour tout $r > 0$, il existe $u \in \text{SL}(\mathbb{R}^{2m})$ tel que $u(B^{2m}(1)) \subset Z^{2m}(r)$.

IV.B – Injection par $u \in \text{SL}(\mathbb{R}^{2m})$ d'une boule dans une autre

Soit $r > 0$ tel qu'il existe $u \in \text{SL}(\mathbb{R}^{2m})$ vérifiant $u(B^{2m}(1)) \subset B^{2m}(r)$.

Notons $U \in \mathcal{M}_{2m}(\mathbb{R})$ la matrice de u dans la base canonique de $\mathbb{R}^{2m}$.

Soit $\lambda \in \mathbb{C}$ une valeur propre complexe de la matrice U .

Q 44. Montrer que $|\lambda| \leq r$.

Pour le cas λ non réel, si P et Q dans $\mathcal{M}_{2m,1}(\mathbb{R})$ sont telles que $Z = P + iQ$ est une colonne propre de U pour la valeur propre λ , on pourra montrer que $\|UP\|^2 + \|UQ\|^2 = |\lambda|^2(\|P\|^2 + \|Q\|^2)$.

Q 45. En déduire que $1 \leq r$.

Q 46. À quelle condition nécessaire et suffisante sur $r > 0$ existe-t-il u appartenant à $\text{SL}(\mathbb{R}^{2m})$ tel que $u(B^{2m}(1)) \subset B^{2m}(r)$?

IV.C – Injection symplectique d'une boule dans un cylindre

Soit $r > 0$ tel qu'il existe un endomorphisme symplectique $\psi \in \text{Symp}_{b_s}(\mathbb{R}^{2m})$ vérifiant $\psi(B^{2m}(1)) \subset Z^{2m}(r)$.

On note $M \in \text{Sp}_{2m}(\mathbb{R})$ la matrice de ψ dans la base canonique $(e_1, \dots, e_m, f_1, \dots, f_m)$ de $\mathbb{R}^{2m}$ et $\psi^\top$ l'endomorphisme canoniquement associé à $M^\top$.

Q 47. Montrer que $|b_s(\psi^\top(e_1), \psi^\top(f_1))| = 1$ puis que $\|\psi^\top(e_1)\| \geq 1$ ou $\|\psi^\top(f_1)\| \geq 1$.

Q 48. Montrer que $1 \leq r$.

Q 49. Montrer le théorème de non-tassement linéaire :

Pour $R > 0$ et $R' > 0$, il existe $\psi \in \text{Symp}_{b_s}(\mathbb{R}^{2m})$ tel que $\psi(B^{2m}(R)) \subset Z^{2m}(R')$ si et seulement si $R \leq R'$.

• • • FIN • • •

SOLUTION

Préliminaires

1. On remarque que

$$X^T AY = \sum_{i=1}^n x_i (AY)_i = \sum_{i=1}^n \sum_{j=1}^n x_i y_j a_{i,j}$$

et en particulier, en notant $(E_1, \dots, E_n)$ la base canonique de $\mathbb{R}^n$, $a_{i,j} = E_i^T A E_j$. On en déduit donc que

$$(\forall X, Y \in \mathbb{R}^n, X^T AY = X^T BY) \implies A = B$$

Remarque : l'expression de $X^T AY$ donnée plus haut doit être connue. On l'utilisera à plusieurs reprises dans ce corrigé.

2. $M^T M$ est symétrique réelle et toutes ses valeurs propres sont donc réelles. Soit λ une valeur propre de $M^T M$ et X un vecteur propre associé (en particulier, $X \neq 0$). On a

$$\lambda \|X\|^2 = X^T (\lambda X) = X^T M^T M X = \|MX\|^2 \geq 0$$

Comme $\|X\|^2 > 0$, on en déduit que $\lambda \geq 0$. Comme M est supposée inversible, $M^T M$ l'est aussi et $\lambda \neq 0$. On a donc montré (le spectre réel et le spectre complexes sont égaux et l'écriture ci-dessous n'est pas ambiguë)

$$\forall M \in GL_n(\mathbb{R}), \operatorname{Sp}(M^T M) \subset \mathbb{R}^{+*}$$

$M^T M$ étant symétrique, elle est orthodiagonalisable par théorème spectral. En notant $d_1, \dots, d_n$ les valeurs propres (> 0) de $M^T M$, il existe $P \in O_n(\mathbb{R})$ telle que $P^{-1} M^T M P = \operatorname{diag}(d_1, \dots, d_n)$.

On peut poser $S = P \operatorname{diag}(\sqrt{d_1}, \dots, \sqrt{d_n}) P^{-1}$. C'est une matrice symétrique (car P est orthogonale) et ses valeurs propres sont > 0 (ce sont les $\sqrt{d_i}$). De plus,

$$S^2 = P \operatorname{diag}(\sqrt{d_1}, \dots, \sqrt{d_n})^2 P^{-1} = P \operatorname{diag}(d_1, \dots, d_n) P^{-1} = M^T M$$

$$\forall M \in GL_n(\mathbb{R}), \exists S \in S_n^{++}(\mathbb{R}), M^T M = S^2$$

Remarque : on a ici seulement prouvé une existence. Il y a en fait aussi unicité. Pour le prouver, on peut raisonner par conditions nécessaires et montrer que si S convient alors elle agit forcément comme $\sqrt{d_i} \operatorname{Id}$ sur le sous-espace propre $E_{d_i}(M^T M)$. Sans l'inversibilité de M , on a encore existence et unicité de $S \in S_n^+(\mathbb{R})$ telle que $M^T M = S^2$. Ces résultats classiques sont à connaître et permettent, par exemple, de prouver le résultat de décomposition polaire (si $A \in \mathcal{M}_n(\mathbb{R})$, il existe $O \in O_n(\mathbb{R})$ et $S \in S_n^+(\mathbb{R})$ telles que $A = OS$).

Objets symplectiques

A. Structure d'espace vectoriel symplectique réel

3. Si ω est une forme symplectique sur E alors pour tout $x \in E$, on a $\omega(x, x) = -\omega(x, x)$ par l'antisymétrie et donc $\omega(x, x) = 0$.

$$\boxed{\text{Si } \omega \text{ est symplectique sur } E, \forall x \in E, \omega(x, x) = 0}$$

4. On montre que F^ω est un sous-espace de E .
Il est non vide puisqu'il contient 0 et est inclus dans E .
Si $x_1, x_2 \in F^\omega$ et $\lambda \in \mathbb{R}$, alors pour tout $y \in F$, on a

$$\omega(x_1 + \lambda x_2, y) = \omega(x_1, y) + \lambda \omega(x_2, y) = 0$$

et donc $x_1 + \lambda x_2 \in F^\omega$. Notre ensemble est stable par combinaisons linéaires.

$$\boxed{F^\omega \text{ est un sous-espace vectoriel de } E}$$

Remarque : en ce début de problème, le correcteur attend des arguments complets et précis. Le rapport note ainsi que la non vacuité de F^ω devait être notée et justifiée.

5. Considérons le cas où $F = \text{Vect}(e)$ avec $e \neq 0$. On a alors $e \in F^\omega$ et $e \in F$. Les espaces ne sont pas en somme directe.

$$\boxed{F \text{ et } F^\omega \text{ ne sont pas toujours en somme directe}}$$

Remarque : on a ici donné un contre-exemple valable pour tout espace de dimension ≥ 1 et toute forme symplectique. On pourrait sans doute se contenter de donner un contre-exemple dans un espace particulier mais cela ne semble pas simplifier les choses. Notez que dans ce type de question, une simple réponse (vrai/faux) ne suffit pas et qu'il faut donner une preuve ou contre-exemple le plus explicite possible.

6. d_ω est bien définie de E dans $\mathcal{L}(E, \mathbb{R})$ car ω est linéaire par rapport à la seconde variable (i.e. pour tout x , $d_\omega(x)$ est bien une forme linéaire) :

$$\forall x \in E, \forall z_1, z_2 \in E, \forall \lambda \in \mathbb{R}, d_\omega(x)(z_1 + \lambda z_2) = d_\omega(x)(z_1) + \lambda d_\omega(x)(z_2)$$

La linéarité de d_ω est une conséquence de la linéarité de ω par rapport à la première variable :

$$\forall x, y \in E, \forall \lambda \in \mathbb{R}, \forall z \in E, d_\omega(x + \lambda y)(z) = d_\omega(x)(z) + \lambda d_\omega(y)(z)$$

Si $d_\omega(x) = 0$ alors $x \in E^\omega$ et ceci entraîne $x = 0$ par non dégénérescence de ω . Ceci donne $\ker(d_\omega) = \{0\}$ et l'injectivité.

Comme E et $\mathcal{L}(E, \mathbb{R})$ ont même dimension n , on conclut que

$$\boxed{d_\omega \text{ est un isomorphisme de } E \text{ dans } \mathcal{L}(E, \mathbb{R})}$$

Ce résultat est semblable à celui de représentation des formes linéaires d'un espace euclidien. Il ne faut rien oublier dans l'argumentation, en particulier le fait que $\omega(x, \cdot)$ est une forme linéaire, ce que l'énoncé semble affirmer et qu'il faut justifier. Notez aussi que l'argument de dimension doit être complet (être en dimension finie ne suffit pas et il faut la même dimension au départ et à l'arrivée).

7. Soit G un supplémentaire de F dans E (il en existe bien un en dimension finie, par exemple à l'aide du théorème de la base incomplète). Soit $\varphi \in \mathcal{L}(F, \mathbb{R})$.

Par théorème de recollement linéaire, il existe une unique application linéaire $\ell \in \mathcal{L}(E, \mathbb{R})$ égale à φ sur F et nulle sur G (puisque φ et l'application nulle sont linéaires). Sa restriction à F est égale à φ et donc $r_F(\ell) = \varphi$. Ainsi

$$\boxed{r_F \text{ est surjective}}$$

Le théorème de recollement linéaire stipule que si $E = E_1 \oplus \dots \oplus E_k$, une application linéaire de E dans F est parfaitement définie par son action sur les E_i . Autrement dit, pour définir une application linéaire, il suffit de la définir sur une famille de supplémentaires. C'est une généralisation du résultat indiquant qu'une application linéaire définie sur un espace E de dimension finie est parfaitement définie par son action sur une base.

8. Soit $x \in \ker(r_F \circ d_\omega)$. On a

$$\forall y \in F, (d_\omega(x))(y) = 0$$

c'est à dire $\omega(x, y) = 0$ pour tout $y \in F$ ou encore $x \in F^\omega$. La réciproque est immédiate. Ainsi

$$\boxed{\ker(r_F \circ d_\omega) = F^\omega}$$

Pour prouver une inclusion entre ensembles, on doit prouver une double inclusion. Nous vous déconseillons de raisonner par équivalence. En revanche, il faut évoquer une réciproque même si elle semble évidente.

Par théorème du rang, $\dim(E) = \dim(\ker(r_F \circ d_\omega)) + \text{rg}(r_F \circ d_\omega)$.

Mais $r_F \circ d_\omega$ est surjective (composée de surjections avec les questions 6 et 7) et donc son rang est égal à $\dim(\mathcal{L}(F, \mathbb{R})) = \dim(F)$. On en conclut que

$$\boxed{\dim(F^\omega) = \dim(E) - \dim(F)}$$

9. Remarquons que ω_F est bilinéaire et antisymétrique sur F car ω l'est sur E . Le problème est celui de la non dégénérescence. Il s'agit donc de montrer que ω_F est non dégénérée si et seulement si F et F^ω sont supplémentaires.

Supposons que $F \oplus F^\omega = E$ et soit $x \in F$ tel que $\forall y \in F, \omega_F(x, y) = 0$. Soit $z \in E$ que l'on peut décomposer en $z = z_1 + z_2$ avec $z_1 \in F$ et $z_2 \in F^\omega$. On a (dernière égalité car $x \in F$ et $z_2 \in F^\omega$)

$$\omega(x, z) = \omega(x, z_1) + \omega(x, z_2) = \omega_F(x, z_1) + \omega(x, z_2) = \omega(x, z_2) = 0$$

Ainsi $x \in E^\omega = \{0\}$ et ω_F est non dégénérée.

Réciproquement, on contrapose et on suppose que F et F^ω ne sont pas supplémentaires. Comme les dimensions sont les bonnes (question 8), on peut en déduire que F et F^ω ne sont pas en somme directe. Il existe un élément e non nul dans l'intersection. On a alors $\omega(z, e) = 0$ pour tout $z \in F$ (car $e \in F^\omega$) et donc $e \in \{x \in F \mid \forall z \in F, \omega_F(x, z) = 0\}$ ce qui montre que ω_F est dégénérée.

$$\boxed{\omega_F \text{ est une forme symplectique sur } F \text{ ssi } F \oplus F^\omega = E}$$

La structure logique de votre raisonnement (sens direct, réciproque) doit clairement apparaître. Il faut aussi préciser le type de raisonnement utilisé (ici, une contraposition dans la réciproque). Dans le même esprit, si vous menez un raisonnement par récurrence ou par l'absurde, il faut l'annoncer.

B. Structure symplectique standard sur $\mathbb{R}^n$

10. Par bilinéarité de ω , pour tous x et y de $\mathbb{R}^n$, on a

$$\begin{aligned}\omega(x, y) &= \omega\left(\sum_{i=1}^n x_i e_i, \sum_{j=1}^n y_j e_j\right) \\ &= \sum_{i=1}^n \sum_{j=1}^n x_i y_j \omega(e_i, e_j)\end{aligned}$$

Avec le calcul de la question 1, on reconnaît $X^T \Omega Y$.

$$\forall x, y \in \mathbb{R}^n, \omega(x, y) = X^T \Omega Y$$

11. L'antisymétrie donne $X^T \Omega Y = -Y^T \Omega X$ pour tous X et Y . Comme cette quantité est réelle, elle est égale à sa transposée et donc

$$\forall X, Y, X^T \Omega Y = -X^T \Omega^T Y$$

La question 1 donne alors l'antisymétrie de Ω ($\Omega = -\Omega^T$).

Par ailleurs, si $Y \in \ker(\Omega)$ alors pour tout x , $\omega(x, y) = X^T 0 = 0$ et donc aussi $\omega(y, x) = 0$.

Ainsi $y = 0$ par caractère non dégénéré de ω et Ω est inversible.

$$\Omega \text{ est antisymétrique et inversible}$$

12. On a $\det(\Omega) = \det(\Omega^T) = \det(-\Omega) = (-1)^n \det(\Omega)$. Comme $\det(\Omega) \neq 0$, $1 = (-1)^n$ et

$$n \text{ est pair}$$

13. La bilinéarité de b_s est conséquence de celle du produit scalaire et de la linéarité de j .

L'antisymétrie de b_s découle de celle de J :

$$b_s(x, y) = X^T J Y \text{ et } b_s(y, x) = Y^T J X$$

Comme $b_s(y, x)$ est un réel il est égal à sa transposée et vaut donc $X^T J^T Y$. On a ainsi $b_s(y, x) = -b_s(x, y)$.

Supposons que $b_s(x, y) = 0$ pour tout $y \in E$. On a alors $X^T J Y = 0$ pour tout Y . Or, J est inversible ($J^2 = -I_n$ et donc $J^4 = I_n$ et l'inverse de J est J^3). En prenant $Y = J^{-1}X$, on trouve $\|X\|^2 = 0$ et donc $x = 0$. b_s est non dégénérée.

$$b_s \text{ est une forme symplectique sur } \mathbb{R}^n$$

C. Endomorphismes et matrices symplectiques réels

14. On a

$$\lambda\mu\omega(x, y) = \omega(\lambda x, \mu y) = \omega(u(x), u(y)) = \omega(x, y)$$

Si $\lambda\mu \neq 1$, $\omega(x, y) = 0$.

Si λ et μ sont tels que $\lambda\mu \neq 1$, alors $E_\lambda(u)$ et $E_\mu(u)$ sont ω -orthogonaux

15. Pour tous $x, y \in \mathbb{R}^n$,

$$\omega(u(x), u(y)) = (MX)^T J(MY) = X^T M^T JMY$$

Si u est b_s -symplectique, alors $X^T M^T JMY = X^T JY$ pour tous X et Y et la question 1 indique que $M^T JM = J$. La réciproque est immédiate.

u est b_s -symplectique ssi $M^T JM = J$

16. Si $M \in \text{Sp}_n(\mathbb{R})$ alors $\det(M)^2 \det(J) = \det(J)$ et donc $\det(M) = \pm 1 \neq 0$ (puisque J est inversible). On a donc $\text{Sp}_n(\mathbb{R}) \subset GL_n(\mathbb{R})$.

Si $M, N \in \text{Sp}_n(\mathbb{R})$ alors $(MN)^T JMN = N^T (M^T JM)N = N^T JN = J$ et $MN \in \text{Sp}_n(\mathbb{R})$.

Si $M \in \text{Sp}_n(\mathbb{R})$ alors $M^T JM = J$ et en multipliant à droite et gauche par M^{-1} et sa transposée, on obtient $M^{-1} \in \text{Sp}_n(\mathbb{R})$.

$\text{Sp}_n(\mathbb{R})$ est donc, par caractérisation, un sous-groupe de $GL_n(\mathbb{R})$.

Si $M \in \text{Sp}_n(\mathbb{R})$ alors $M^T JM = J$ et en passant à l'inverse, on obtient $M^{-1} J^{-1} (M^T)^{-1} = J^{-1}$ et comme $J^{-1} = -J$, on trouve

$$M^{-1} J (M^T)^{-1} = J$$

En remarquant que M^{-1} est la transposée de $(M^T)^{-1}$, on a montré que $(M^{-1})^T = (M^T)^{-1}$ est dans $\text{Sp}_n(\mathbb{R})$ et avec ce qui précède (stabilité par passage à l'inverse), $M^T \in \text{Sp}_n(\mathbb{R})$.

Enfin, on remarqué plus haut que $J^2 = -I_n$ et donc $-J^3 = J$ ou encore $J \in \text{Sp}_n(\mathbb{R})$.

$\text{Sp}_n(\mathbb{R})$ est un sous-groupe de $GL_n(\mathbb{R})$ stable par transposition et contenant J

17. Le calcul par blocs donne

$$M^T JM = \begin{pmatrix} A^T & C^T \\ B^T & D^T \end{pmatrix} \begin{pmatrix} -C & -D \\ A & B \end{pmatrix} = \begin{pmatrix} -A^T C + C^T A & -A^T D + C^T B \\ -B^T C + D^T A & -B^T D + D^T B \end{pmatrix}$$

Si M est symplectique alors les blocs correspondent et donc

$$-A^T C + C^T A = -B^T D + D^T B = 0, \quad A^T D - C^T B = B^T C + D^T A = I_m$$

Les deux premières égalités donnent la symétrie de $A^T C$ et $B^T D$ (rappelons que $(UV)^T = V^T U^T$). On a ainsi

$$(A^T C \text{ et } B^T D \text{ symétriques et } A^T D - C^T B = I_m)$$

Réciproquement, si ces conditions sont vérifiées alors les quatre blocs de $M^T J M$ et J sont égaux (en particulier car $B^T C + D^T A$ est la transposée de $A^T D - C^T B$).

On a donc montré que

$$M \in \mathrm{Sp}_{2m}(\mathbb{R}) \text{ ssi } (A^T C \text{ et } B^T D \text{ symétriques et } A^T D - C^T B = I_m)$$

A nouveau, il est vivement conseillé de ne pas raisonner par équivalence, même si c'est sans doute possible ici. Notez que le programme ne dit pas grand chose sur le calcul par blocs. On doit savoir faire une somme ou un produit par blocs (en vérifiant que les tailles correspondent, c'est à dire que les sommes ou produits existent. On peut aussi faire des opérations élémentaires par blocs (par exemple des transvections par blocs). Quant aux déterminants, seuls les déterminants bloc-triangulaires sont supposés connus (voir la question 20).

Déterminant d'une matrice symplectique réelle

A. Le cas de la dimension 2

18. On utilise la question 17 dans le cas $m = 1$. $\begin{pmatrix} a & b \\ c & d \end{pmatrix}$ est symplectique seulement si $ad - bc = 1$ (la condition de symétrie pour un bloc de taille 1 est vérifiée). On reconnaît le déterminant et donc

$$\mathrm{Sp}_2(\mathbb{R}) = \mathrm{SL}_2(\mathbb{R})$$

Si on revient à la définition de $\mathrm{Sp}_2(\mathbb{R})$, on procédera par double inclusion. Ici, on obtient directement l'équivalence car on a utilisé le résultat de la question 17.

B. Commutant de J

19. On écrit $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ par blocs comme en question 17 et on trouve

$$MJ = \begin{pmatrix} B & -A \\ D & -C \end{pmatrix} \quad \text{et} \quad JM = \begin{pmatrix} -C & -D \\ A & B \end{pmatrix}$$

La condition nécessaire et suffisante de commutation est donc $B = -C$ et $D = A$. Avec les notations de l'énoncé, on a donc

$$\mathcal{C}_J = \left\{ \begin{pmatrix} U & -V \\ V & U \end{pmatrix}, U, V \in \mathcal{M}_m(\mathbb{R}) \right\}$$

20. Le calcul par blocs donne

$$\begin{pmatrix} I_m & 0 \\ iI_m & I_m \end{pmatrix} \begin{pmatrix} U & -V \\ V & U \end{pmatrix} \begin{pmatrix} I_m & 0 \\ -iI_m & I_m \end{pmatrix} = \begin{pmatrix} U + iV & -V \\ 0 & U - iV \end{pmatrix}$$

Par propriété de morphisme du déterminant et puisqu'un déterminant bloc-triangulaire est le produit des déterminants des blocs,

$$\det \begin{pmatrix} U & -V \\ V & U \end{pmatrix} = \det(U + iV) \det(U - iV)$$

Par propriété de morphisme du passage au conjugué ($z \mapsto \bar{z}$ est un automorphisme de corps), le conjugué du déterminant est le déterminant du conjugué (puisque le déterminant est une somme ou différence de produit des coefficients). Et puisque U et V sont réelles, on a donc

$$\overline{\det(U + iV)} = \det(U - iV)$$

On en déduit que

$$\det \begin{pmatrix} U & -V \\ V & U \end{pmatrix} = |\det(U + iV)|^2 \in \mathbb{R}^+$$

Avec la question 19, on a donc

$$\forall M \in \mathcal{C}_J, \det(M) \geq 0$$

C. Décomposition polaire d'une matrice symplectique réelle

21. $\text{OSp}_n(\mathbb{R})$ est un sous-groupe de $GL_n(\mathbb{R})$ comme intersection de sous-groupes. Comme il est inclus dans le groupe $\text{Sp}_n(\mathbb{R})$, c'est un sous-groupe de ce groupe.

$\text{Sp}_n(\mathbb{R})$ est fermé comme image réciproque du fermé $\{J\}$ par l'application continue $M \mapsto M^T J M$.

$O_n(\mathbb{R})$ est compact (fermé comme image réciproque de $\{I_n\}$ par $M \mapsto M^T M$ qui est continue, inclus dans la sphère de centre 0 de rayon $\sqrt{n}$ pour la norme euclidienne canonique de $\mathcal{M}_n(\mathbb{R})$, le tout en dimension finie).

Un fermé dans un compact étant compact,

$$\text{OSp}_n(\mathbb{R}) \text{ est un sous-groupe compact de } \text{Sp}_n(\mathbb{R})$$

Remarque : les arguments de fermeture, de bornitude, de dimension finie sont ceux qui sont attendus. Il y a d'autres façons de les justifier. Notez que la compacité de $O_n(\mathbb{R})$ n'est pas au programme. Même à ce niveau du problème, le rapport nous indique qu'elle doit être justifiée. On peut penser que dans une bonne copie la seule mention de cette compacité ne sera pas sanctionnée mais cela indique bien le niveau de précision attendu.

22. Soit $M \in \text{OSp}_n(\mathbb{R})$. On a d'une part $M^T J M = J$ et d'autre part $M M^T = M^T M = I_n$. En multipliant la première identité par M à gauche, on trouve $J M = M J$.

$$\text{OSp}_n(\mathbb{R}) \subset \mathcal{C}_J$$

23. Si $M \in \text{OSp}_n(\mathbb{R})$ alors $\det(M) \geq 0$ (car $M \in \mathcal{C}_J$) et $\det(M)^2 = 1$ (car M orthogonale). Ainsi, $\det(M) = 1$.

$$\text{OSp}_n(\mathbb{R}) \subset SL_n(\mathbb{R})$$

24. S étant symétrique réelle, on peut trouver une b.o.n. $(f_1, \dots, f_n)$ formée de vecteurs propres. On note d_i la valeur propre associée à f_i ($d_i > 0$ par hypothèse). M étant symplectique, sa transposée l'est aussi et S^2 de même (stabilité par produit et passage à la transposée). Pour tous $i, j \in \llbracket 1, n \rrbracket$, on a donc

$$b_s(f_i, f_j) = b_s(s^2(f_i), s^2(f_j)) = d_i^2 d_j^2 b_s(f_i, f_j)$$

On a aussi

$$b_s(s(f_i), s(f_j)) = d_i d_j b_s(f_i, f_j)$$

- Si $d_i d_j = 1$ alors, on a immédiatement $b_s(f_i, f_j) = b_s(s(f_i), s(f_j))$.
- Si $d_i d_j \neq 1$ alors $d_i^2 d_j^2 \neq 1$ (car les valeurs propres sont positives) et alors $b_s(f_i, f_j) = 0$. Et la deuxième égalité donne alors aussi $b_s(s(f_i), s(f_j)) = 0$.

Pour tous i, j , on a donc $b_s(f_i, f_j) = b_s(s(f_i), s(f_j))$. En décomposant x et y de $\mathbb{R}^n$ sur la base $(f_1, \dots, f_n)$, on obtient

$$b_s(s(x), s(y)) = \sum_{1 \leq i, j \leq n} x_i y_j b_s(s(f_i), s(f_j)) = \sum_{1 \leq i, j \leq n} x_i y_j b_s(f_i, f_j) = b_s(x, y)$$

et ceci pour tous x, y . s est donc un endomorphisme symplectique pour b_s et donc

S est symplectique

25. S étant à valeurs propres > 0 est inversible (0 n'est pas valeur propre). MS^{-1} est symplectique car l'ensemble de telles matrices est un groupe. Enfin

$$(MS^{-1})^T (MS^{-1}) = (S^{-1})^T M^T MS^{-1} = S^{-1} S^2 S^{-1} = I_n$$

et MS^{-1} est orthogonale.

$$O = MS^{-1} \in \text{OSp}_n(\mathbb{R})$$

26. On a donc $1 = \det(O) = \det(M) \det(S^{-1})$ et donc $\det(M) = \det(S) \geq 0$ (S est à valeurs propres positives et diagonalisable : son déterminant est le produit de ses valeurs propres comptées avec multiplicité).

De $M^T J M = J$, on obtient (puisque $\det(J) \neq 0$) que $\det(M)^2 = 1$. On peut donc conclure que $\det(M) = 1$. On a montré que

$$\text{Sp}_n(\mathbb{R}) \subset \text{SL}_n(\mathbb{R})$$

D. Génération du groupe symplectique

D.1. Transvection symplectique

27. $\ell : x \mapsto \lambda \omega(a, x)$ est une forme linéaire et $\ell(a) = \lambda \omega(a, a) = 0$ (question 3) et donc $a \in \ker(\ell)$. τ_a^λ est donc une transvection. De plus (développement par bilinéarité)

$$\omega(\tau_a^\lambda(x), \tau_a^\lambda(y)) = \omega(x, y) + \lambda \omega(a, x) \omega(a, y) + \lambda \omega(a, y) \omega(x, a) + \lambda^2 \omega(a, x) \omega(a, y) \omega(a, a)$$

Comme $\omega(a, a) = 0$ et $\omega(a, x) = -\omega(x, a)$, ce terme est égal à $\omega(x, y)$ et τ_a^λ est symplectique.

$$\tau_a^\lambda \text{ est une transvection qui est un endomorphisme symplectique}$$

28. Par linéarité,

$$\begin{aligned} \tau_a^\mu \circ \tau_a^\lambda(x) &= \tau_a^\mu(x) + \lambda \omega(x, a) \tau_a^\mu(a) \\ &= x + \mu \omega(a, x) a + \lambda \omega(x, a) (a + \mu \omega(a, a) a) \\ &= x + (\lambda + \mu) \omega(a, x) a \end{aligned}$$

On a donc (ce qui précède étant vrai pour tout x)

$$\tau_a^\mu \circ \tau_a^\lambda = \tau_a^{\lambda+\mu}$$

29. a est un élément non nul du noyau de $d_a : x \mapsto \omega(a, x)$.
 Si $d_a = 0$ alors $\tau_a^\lambda = \text{Id}$ est de déterminant $1 > 0$.
 Sinon, $\ker(d_a)$ est un hyperplan de E et on peut par théorème de la base incomplète trouver une base $(a, f_3, \dots, f_n)$ de celui-ci. On peut aussi compléter en $(f_1, a, f_3, \dots, f_n)$ base de E . En notant $E_{2,1}$ la matrice dont tous les coefficients sont nuls sauf celui en position $(2, 1)$ qui vaut 1, τ_a^λ est représenté dans cette base, par la matrice

$$I_n + \lambda \omega(a, f_1) E_{2,1}$$

qui est de déterminant 1 .

$$\det(\tau_a^\lambda) = 1 > 0$$

Remarque : le passage par une matrice est tout à fait naturel, c'est en général ainsi que l'on obtient le déterminant d'un endomorphisme.

30. Avec la question 28, $\tau_a^{-\lambda} \circ \tau_a^\lambda = \tau_a^0 = \text{Id}$ et de même pour $\tau_a^\lambda \circ \tau_a^{-\lambda}$ donc (comme on est dimension finie, on aurait pu se contenter d'un des deux égalités)

$$(\tau_a^\lambda)^{-1} = \tau_a^{-\lambda} \text{ est une transvection symplectique}$$

D.2. Un lemme

31. $\tau_{y-x}^\lambda(x) = x + \lambda \omega(x - y, x)(x - y) = x - \lambda \omega(y, x)(x - y)$ et ainsi

$$\tau_{y-x}^\lambda(x) - y = (1 + \lambda \omega(x, y))(x - y)$$

Si $\omega(x, y) \neq 0$, il suffit de choisir $\lambda = -1/\omega(x, y)$ pour que x soit envoyé sur y .

$$\text{Si } \omega(x, y) \neq 0, \text{ il y a une transvection symplectique qui transforme } x \text{ en } y$$

Remarque : on a mené un raisonnement par conditions nécessaires qui nous montre même l'unicité de λ . Celle-ci n'étant pas demandé, la présentation ne parle pas d'analyse et synthèse mais seulement de condition suffisante.

32. On suppose $\omega(x, y) = 0$. Notons $F = \text{Vect}(x, y)$. ω_F est alors nulle et n'est pas une forme symplectique sur F (puisque $\dim(F) \geq 1$). Par non dégénérescence de ω , il existe z_1, z_2 tels que $\omega(x, z_1) \neq 0$ et $\omega(y, z_2) \neq 0$. Distinguons trois cas.

- Si $\omega(y, z_1) \neq 0$ alors on pose $z = z_1$ et on a $\omega(x, z)$ et $\omega(y, z)$ non nul.
- Si $\omega(x, z_2) \neq 0$ alors on pose $z = z_2$ et on a $\omega(x, z)$ et $\omega(y, z)$ non nul.
- Sinon, $\omega(x, z_2) = \omega(y, z_1) = 0$. On pose $z = z_1 + z_2$ et on a $\omega(x, z) = \omega(x, z_1) \neq 0$ et de même $\omega(y, z) \neq 0$.

$$\text{Si } \omega(x, y) = 0, \text{ il existe } z \in E \text{ tel que } \omega(x, z)\omega(y, z) \neq 0$$

33. Si $\omega(x, y) = 0$, on introduit z comme ci-dessus. On trouve une transvection symplectique τ_1 telle que $\tau_1(x) = z$ avec la question 31. Et aussi une autre τ_2 telle que $\tau_2(z) = y$ (puisque $\omega(z, y) = -\omega(y, z) \neq 0$). $\gamma = \tau_2 \circ \tau_1$ transforme x en y .
 Si $\omega(x, y) \neq 0$, la question 31 donne directement une transvection envoyant x sur y .

$$\text{Si } x, y \neq 0, \text{ il existe une composée d'au plus deux transvections symplectique qui envoie } x \text{ sur } y$$

D.3. Le théorème

34. Comme ω est non dégénérée, il existe f tel que $\omega(e_1, f) \neq 0$. $f_1 = \frac{1}{\omega(e_1, f)} f$ vérifie $\omega(e_1, f_1) = 1$.

Il existe f_1 tel que $\omega(e_1, f_1) = 1$

f_1 est non colinéaire à e_1 sinon $\omega(e_1, f_1) = 0$ avec la question 3.

35. On a $\omega(u(e_1), u(f_1)) = \omega(e_1, f_1) = 1$ car u est symplectique. On en déduit que $u(e_1) \neq 0$. On peut alors utiliser le lemme avec $y = e_1$ et $x = u(e_1)$ et affirmer que

Il existe δ_1 composée d'au plus deux transvections symplectiques tel que $\delta_1(u(e_1)) = e_1$

36. *Question délicate ! Il faut comprendre comment généraliser le lemme de III.D.2. C'est une question qui rapportera sans doute un peu plus de points mais au vu de la longueur du sujet, on n'y passera pas trop de temps.*

On montre le lemme suivant : si $\omega(v, w_1) = \omega(v, w_2) = 1$, alors il existe γ composée d'au plus deux transvections symplectiques tel que $\gamma(v) = v$ et $\gamma(w_1) = w_2$.

- Si $\omega(w_1, w_2) \neq 0$, on peut poser $a = w_1 - w_2$ et $\lambda = -\frac{1}{\omega(a, w_1)}$ car $\omega(a, w_1) = -\omega(w_2, w_1) \neq 0$. On a $\omega(a, v) = \omega(w_1, v) - \omega(w_2, v) = 0$ et donc

$$\tau_a^\lambda(v) = v + \lambda\omega(a, v)a = v$$

$$\tau_a^\lambda(w_1) = w_1 + \lambda\omega(a, w_1)a = w_1 - (w_1 - w_2) = w_2$$

- Sinon, $\omega(v, v + w_1) = \omega(v, w_1) = 1$ et $\omega(w_1, v + w_1) = \omega(w_1, v) = -1 \neq 0$ et le premier cas donne τ_1 tel que $\tau_1(v) = v$ et $\tau_1(w_1) = v + w_1$.

De même $\omega(v + w_1, w_2) = \omega(v, w_2) = 1$ et le premier cas donne τ_2 tel que $\tau_2(v) = v$ et $\tau_2(v + w_1) = w_2$.

$\gamma = \tau_2 \circ \tau_1$ convient alors

Il suffit alors d'utiliser le lemme avec $v = e_1$, $w_1 = \tilde{f}_1$ et $w_2 = f_1$ en remarquant que $\omega(e_1, f_1) = 1$ mais aussi (u étant symplectique)

$$\omega(e_1, \tilde{f}_1) = \omega(\delta_1(u(e_1)), \delta_1(u(f_1))) = \omega(e_1, f_1) = 1$$

Il existe δ_2 convenable

37. Par construction de δ , on a $v(e_1) = e_1$ et $v(f_1) = f_1$. Ainsi (l'action d'une application linéaire est caractérisée par l'image d'une base)

P est stable par v et $v_P = \text{Id}_P$

38. Soit $x \in P^\omega$: on a donc $\omega(x, e_1) = \omega(x, f_1) = 0$. On a alors (on utilise $v = \delta \circ u$ puis le fait que δ et u sont symplectiques)

$$\omega(v(x), e_1) = \omega(\delta(u(x)), \delta(u(e_1))) = \omega(u(x)u(e_1)) = \omega(x, e_1) = 0$$

et de même $\omega(v(x), f_1) = 0$. Puis par combinaisons linéaires, $\omega(x, y) = 0$ dès que $y \in P$.

P^ω est stable par v

39. Si F est un sous-espace de E alors il est immédiat que $F \subset (F^\omega)^\omega$ (si $x \in F$ alors pour tout $y \in F^\omega$, $\omega(x, y) = 0$).

Avec la question 8 utilisée avec F et F^ω , $(F^\omega)^\omega$ et F ont même dimension et ainsi, $(F^\omega)^\omega = F$.

D'après la question 9, pour montrer que ω_{P^ω} est symplectique, il suffit de montrer que P^ω et $(P^\omega)^\omega = P$ sont supplémentaires. Et comme ils ont les bonnes dimensions, il suffit de montrer que $P \cap (P^\omega)^\omega = \{0\}$.

Soit donc $x \in P \cap P^\omega$. On a $\omega(x, e_1) = \omega(x, f_1) = 0$ (appartenance à P^ω) et l'existence de α, β tels que $x = \alpha e_1 + \beta f_1$. Ainsi

$$0 = \omega(x, f_1) = \alpha \omega(e_1, f_1) = \alpha \quad \text{et} \quad 0 = \omega(x, e_1) = \beta \omega(f_1, e_1) = -\beta$$

ce qui donne bien $x = 0$.

 ω_{P^ω} est une forme symplectique sur P^ω

Le caractère symplectique de v sur tout l'espace entraîne immédiatement (c'est vrai partout et donc sur P^ω) celui de la restriction. Ainsi

 v_{P^ω} est un endomorphisme symplectique de P^ω

40. On peut maintenant prouver par récurrence sur m que : "tout endomorphisme symplectique d'un espace symplectique de dimension $2m$ peut s'écrire comme la composée d'au plus $4m$ transvections symplectiques".

- Si $m = 1$, on se donne un endomorphisme symplectique u de E de dimension 2. Les questions 34 à 36 permettent de construire δ composée d'au plus quatre transvections symplectiques telle que $\delta \circ u = \text{Id}$ (envoie les éléments d'une base sur eux mêmes). $u = \delta^{-1}$ est la composée d'au plus quatre transvections (avec la question 30).
- Supposons le résultat vrai pour des valeurs plus petites que $m \geq 1$ (c'est une récurrence forte mais cela est en fait inutile ici). Soit E symplectique de dimension $2(m+1)$ et u endomorphisme symplectique de E . On construit avec les questions 34 à 36 un plan P et un endomorphisme $v = \delta \circ u$. On peut appliquer l'hypothèse de récurrence à v_{P^ω} et le décomposer en produit $\tau_p \circ \dots \circ \tau_1$ d'au plus $4m$ transvections symplectiques de P^ω .

Il faut maintenant construire à partir des τ_i des transvections symplectiques de E . Soit $i \in \llbracket 1, p \rrbracket$. Il existe λ_i réel et $a_i \in P^\omega \setminus \{0_E\}$ tels que

$$\forall x \in P^\omega, \tau_i(x) = x + \lambda_i \omega_{P^\omega}(a_i, x) a_i = x + \lambda_i \omega(a_i, x) a_i$$

On prolonge la transvection symplectique sur E par la même expression que l'on note de la même manière. Comme $a_i \in P^\omega$, elle agit comme l'identité sur P . Ainsi, l'égalité $\delta \circ u = \tau_p \circ \dots \circ \tau_1$ reste vraie sur E avec les nouvelles transvections (on a l'égalité sur deux supplémentaires).

On a alors $u = \delta^{-1} \circ \tau_p \circ \dots \circ \tau_1$ et δ^{-1} est la composée d'au plus quatre transvection ce qui conclut la récurrence.

Les transvections symplectiques engendrent le groupe symplectique

Remarque : le fond du raisonnement est le raisonnement par récurrence (qu'il faut expliciter). Dans cette récurrence, il faut bien comprendre qu'on obtient d'abord des transvections symplectiques sur un espace symplectique plus petit et que l'on doit prolonger celles-ci en des transvections symplectiques sur l'espace tout entier. C'est presque la seule subtilité du raisonnement si on a pensé à procéder par récurrence.

D.4. Une conséquence topologique

41. On identifie ici endomorphisme de $\mathbb{R}^n$ et matrice dans la base canonique afin d'utiliser librement ce qui précède.

Soit $u \in \mathrm{Sp}_n(\mathbb{R})$ et $u = \tau_p \circ \dots \circ \tau_1$ une décomposition de u en produit de transvections. A chaque τ_i , on peut associer un réel λ_i et un vecteur a_i . Posons

$$\tau_i(t) : x \mapsto x + t\lambda_i\omega(a_i, x)a_i$$

et $\varphi : t \mapsto \tau_p(t) \circ \dots \circ \tau_1(t)$. φ est définie de $[0, 1]$ dans $\mathrm{Sp}_n(\mathbb{R})$ et vérifie $u(0) = \mathrm{Id}$ et $\varphi(1) = u$. Par ailleurs, φ est continue (matriciellement, les coefficients de la matrice sont des polynômes de degré ≤ 1 en t). On a donc un chemin continu dans $\mathrm{Sp}_n(\mathbb{R})$ de Id vers u . Tous les éléments de $\mathrm{Sp}_n(\mathbb{R})$ sont donc dans la classe de connexité par arcs de l'identité et

$\mathrm{Sp}_n(\mathbb{R})$ est connexe par arcs

Remarque : formellement, il aurait fallu à partir d'une matrice symplectique introduire l'endomorphisme canoniquement associé. Nous avons ici privilégié la simplicité de la rédaction en faisant l'identification annoncée au début. De même, la continuité de l'application étant assez simple, nous sommes passés rapidement sur celle-ci, l'argument donné nous semblant suffisant. A ce niveau du problème (question 41 !), il faut savoir faire des choix.

D.5. Deuxième conséquence

42. En question 29, on a montré que les transvections symplectiques sont de déterminant 1 et donc appartiennent à $SL_n(\mathbb{R})$. Comme ce dernier ensemble est un groupe, il contient le groupe engendré par les transvections symplectiques et ainsi

$$\mathrm{Sp}_n(\mathbb{R}) \subset SL_n(\mathbb{R})$$

Exemples de problèmes de plongements symplectiques linéaires

A. Injection par $u \in SL(\mathbb{R}^{2m})$ d'une boule dans un cylindre

43. Considérons l'application

$$u = (x_1, \dots, x_m, y_1, \dots, y_m) \mapsto (rx_1, x_2/r, x_3, \dots, x_m, ry_1, y_2/r, y_3, \dots, y_m)$$

que l'on peut bien définir car $m \geq 2$. Cette application est linéaire et représentée dans la base canonique par une matrice diagonale avec pour coefficient deux r , deux $1/r$ et des 1. Le déterminant de u vaut donc 1 et $u \in SL(\mathbb{R}^{2m})$.

Si $(x_1, \dots, x_m, y_1, \dots, y_m) \in B^{2m}(1)$, alors $x_1^2 + y_1^2 \leq x_1^2 + \dots + x_m^2 + y_1^2 + \dots + y_m^2 = 1$ et donc $(rx_1)^2 + (ry_1)^2 \leq r^2$. On a donc $u(B^{2m}(1)) \subset Z^{2m}(r)$.

B. Injection par $u \in SL(\mathbb{R}^{2m})$ d'une boule dans une autre

44. Si $\lambda \in \mathbb{R}$, on peut considérer $x \in \mathbb{R}^{2m}$ un vecteur propre associé et quitte à le diviser par sa norme (cela reste un vecteur propre), on peut supposer $\|x\| = 1$ et donc $x \in B^{2m}(1)$. On a alors $\lambda x = u(x) \in B^{2m}(r)$ et donc $\|\lambda x\| \leq r$ ce qui donne $|\lambda| \leq r$.

Si $\lambda \in \mathbb{C} \setminus \mathbb{R}$, on travaille matriciellement. λ est valeur propre de U et on peut trouver un vecteur propre $X \in \mathbb{C}^{2m}$ que l'on écrit $P + iQ$ avec $P, Q \in \mathbb{R}^{2m}$. On écrit aussi $\lambda = a + ib$. En identifiant parties réelles et imaginaires dans $UZ = \lambda Z$, on obtient

$$aP - bQ = UP \quad \text{et} \quad aQ + bP = UQ$$

On en déduit que

$$\|UP\|^2 = (aP - bQ)^T(aP - bQ) = a^2\|P\|^2 + b^2\|Q\|^2 - abP^TQ - baQ^TP$$

$$\|UQ\|^2 = (aQ + bP)^T(aQ + bP) = a^2\|Q\|^2 + b^2\|P\|^2 + abQ^TP + baP^TQ$$

et donc

$$\|UP\|^2 + \|UQ\|^2 = (a^2 + b^2)(\|P\|^2 + \|Q\|^2) = |\lambda|^2(\|P\|^2 + \|Q\|^2)$$

Supposons P et Q non nuls. Comme $\frac{P}{\|P\|}$ et $\frac{Q}{\|Q\|}$ sont dans $B^{2m}(1)$, leurs images par U sont dans $B^{2m}(r)$ et l'égalité précédente donne

$$|\lambda|^2(\|P\|^2 + \|Q\|^2) \leq r^2(\|P\|^2 + \|Q\|^2)$$

Comme $\|P\|^2 + \|Q\|^2 > 0$, on peut diviser pour obtenir

$$|\lambda| \leq r$$

Dans le cas où $P = 0$ (ou $Q = 0$) alors $Q \neq 0$ (ou $P \neq 0$) et on peut reprendre le raisonnement sans peine.

45. Le produit des valeurs propres complexes (comptées avec multiplicité) est égal au déterminant (trigonalisabilité dans $\mathbb{C}$) et vaut donc 1. On en déduit que $1 \leq r^{2m}$ et donc que

$$1 \leq r$$

46. Une condition nécessaire vient d'être vue : c'est $r \geq 1$. La réciproque est immédiate car l'identité convient alors.

$$\text{L'injection par } SL(\mathbb{R}^{2m}) \text{ existe ssi } r \geq 1$$

C. Injection symplectique d'une boule dans un cylindre

47. M^T étant symplectique comme M , ψ^T est un endomorphisme symplectique et (en notant E_1 et F_1 les colonnes associées à e_1 et f_1)

$$b_s(\psi^T(e_1), \psi^T(f_1)) = b_s(e_1, f_1) = E_1^T J F_1$$

$$\boxed{|b_s(\psi^T(e_1), \psi^T(f_1))| = 1}$$

On a également à l'aide de Cauchy-Schwarz :

$$\left| b_s(\psi^T(e_1), \psi^T(f_1)) \right| = \left| \langle \psi^T(e_1), j(\psi^T(f_1)) \rangle \right| \leq \|\psi^T(e_1)\| \times \|j(\psi^T(f_1))\|$$

or $J \in O_n(\mathbb{R})$ car ses colonnes forment une base orthonormée de $\mathbb{R}^n$.

Ainsi $j \in O(\mathbb{R}^n)$ et $\|j(\psi^T(f_1))\| = \|\psi^T(f_1)\|$

D'où $1 \leq \|\psi^T(e_1)\| \cdot \|\psi^T(f_1)\|$

On peut conclure que $\boxed{\|\psi^T(e_1)\| \geq 1 \text{ ou } \|\psi^T(f_1)\| \geq 1}$

48. On remarque que

$$\forall x, y \in \mathbb{R}^n, \langle x, \psi^T(y) \rangle = \langle \psi(x), y \rangle$$

en utilisant l'écriture matricielle du produit scalaire usuel de $\mathbb{R}^n$.

Par ailleurs on a $\forall x \in \psi(B^{2m}(1)), \psi(x) \in Z^{2m}(r)$

donc en regardant à la condition sur la première coordonnée, on a

$$\forall x \in \psi(B^{2m}(1)), |\langle \psi(x), e_1 \rangle| \leq r$$

En combinant les inégalités, on a

$$\forall x \in B^{2m}(1), |\langle x, \psi^T(e_1) \rangle| \leq r$$

En appliquant cela à $\frac{\psi^T(e_1)}{\|\psi^T(e_1)\|} \in B^{2m}(1)$, on obtient $r \geq \|\psi^T(e_1)\|$

De manière analogue, on établit que $r \geq \|\psi^T(f_1)\|$.

A l'aide de la question précédente, on conclut que $\boxed{1 \leq r}$

49. On prouve d'abord le résultat dans le cas $R = 1$ et $R' = r$. il s'agit donc de montrer que pour $r > 0$

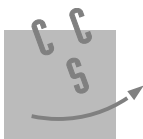
il existe $\psi \in \text{Symp}_{b_s}(\mathbb{R}^{2m})$ tel que $\psi(B^{2m}(1)) \subset Z^{2m}(r)$ si et seulement si $1 \leq r$

- Si $r \geq 1$, il suffit de prendre $\psi = \text{Id}_{\mathbb{R}^n}$.
- Réciproquement, on suppose qu'il existe $\psi \in \text{Symp}_{b_s}(\mathbb{R}^{2m})$ tel que $\psi(B^{2m}(1)) \subset Z^{2m}(r)$. On a alors $r \geq 1$ selon Q48.

Le cas général s'en déduit par linéarité et changement d'échelle puisque si u est un endomorphisme de $\mathbb{R}^n$ et $R, R' > 0$, on a

$$u(B^{2m}(R)) \subset Z^{2m}(R') \iff u(B^{2m}(1)) \subset Z^{2m}(R'/R)$$

$$\boxed{(\exists \psi \in \text{Symp}_{b_s}(\mathbb{R}^{2m}), \psi(B^{2m}(R)) \subset Z^{2m}(R')) \iff R' \geq R}$$



Mathématiques 2

MP

2022

CONCOURS CENTRALE-SUPÉLEC

4 heures

Calculatrice autorisée

Objectifs

Ce problème étudie la dérivation des sommes de séries de fonctions $\sum f_n$ de deux façons différentes : un point de vue déterministe et un point de vue probabiliste. Pour conclure à une formule du type $\left(\sum_{n=0}^{+\infty} f_n\right)^{(K)} = \sum_{n=0}^{+\infty} f_n^{(K)}$ avec K entier supérieur ou égal à 2, les théorèmes usuels contiennent généralement au moins une hypothèse sur les dérivées intermédiaires $f'_n, \dots, f_n^{(K-1)}$ (par exemple de convergence simple sur tout l'intervalle ou même en un seul point). Le sujet montre que l'on peut affaiblir l'hypothèse de contrôle des dérivées intermédiaires par une hypothèse de convergence de séries numériques de la forme $\sum f_n(x)$ où x parcourt un ensemble fini. Cette dernière hypothèse sera de nouveau affaiblie dans la partie probabiliste consacrée à la dérivation de séries aléatoires de fonctions.

Le sujet est divisé en quatre parties :

- la partie I étudie une inégalité, qualifiée d'inégalité d'interpolation, qui permet de contrôler les dérivées intermédiaires d'une fonction de classe $\mathcal{C}^K$;
- la partie II utilise la partie I pour démontrer un résultat de transfert du caractère $\mathcal{C}^K$ à une somme de série de fonctions ;
- la partie III, qui est indépendante des parties I et II, étudie la convergence des séries aléatoires numériques de la forme $\sum X_n a_n$, où (X_n) est une suite de variables aléatoires mutuellement indépendantes de Rademacher et (a_n) une suite réelle telle que la série $\sum a_n^2$ converge ;
- la partie IV utilise les résultats des parties précédentes pour donner une application au caractère $\mathcal{C}^K$ de la somme d'une série aléatoire de fonctions de la forme $\sum X_n f_n$.

Notations

- Pour tous entiers i et j vérifiant $i \leq j$, la notation $\llbracket i, j \rrbracket$ désigne l'intervalle d'entiers $[i, j] \cap \mathbb{N}$.
- La lettre K désigne systématiquement un entier naturel non nul.
- Le symbole $\mathbb{R}_{K-1}[X]$ désigne le $\mathbb{R}$ -espace vectoriel des polynômes de degré inférieur ou égal à $K-1$ à coefficients réels.
- Pour tout intervalle I , on note $\mathcal{C}^K(I)$ le $\mathbb{R}$ -espace vectoriel des fonctions $f : I \rightarrow \mathbb{R}$ de classe $\mathcal{C}^K$. Pour tous $f \in \mathcal{C}^K(I)$ et $k \in \llbracket 0, K \rrbracket$, on note $f^{(k)}$ la dérivée d'ordre k (et donc $f^{(0)} = f$, $f^{(1)} = f'$, $f^{(2)} = f''$).
- Dans le cas particulier $I = [0, 1]$, pour toute fonction bornée $f : [0, 1] \rightarrow \mathbb{R}$, on note $\|f\|_\infty = \sup_{x \in [0, 1]} |f(x)|$.

I Inégalités d'interpolation des dérivées

Soit K réels distincts $x_1 < \dots < x_K$ de l'intervalle $[0, 1]$. Le but de cette partie est de montrer le résultat suivant : il existe une constante $C > 0$ (dépendant des réels $x_1, \dots, x_K$) telle que

$$\forall f \in \mathcal{C}^K([0, 1]), \quad \max_{0 \leq k \leq K-1} \|f^{(k)}\|_\infty \leq \|f^{(K)}\|_\infty + C \sum_{\ell=1}^K |f(x_\ell)|. \quad (\text{I.1})$$

Une inégalité du type précédent est appelée *inégalité d'interpolation* à l'ordre K .

I.A – Cas particulier $K = 1$

On fixe $x_1 \in [0, 1]$ et on étudie une inégalité d'interpolation à l'ordre 1,

$$\forall f \in \mathcal{C}^1([0, 1]), \quad \|f\|_\infty \leq \|f'\|_\infty + C |f(x_1)|. \quad (\text{I.2})$$

Q 1. Montrer l'inégalité d'interpolation (I.2) avec $C = 1$.

Q 2. Soit $C \in]0, 1[$. À l'aide d'un exemple simple de fonction f , montrer que l'inégalité d'interpolation (I.2) est fautive.

I.B – Cas particulier $K = 2$

On fixe deux réels distincts $x_1 < x_2$ de $[0, 1]$. On veut construire une constante $C > 0$ telle qu'on ait l'inégalité d'interpolation à l'ordre 2,

$$\forall f \in \mathcal{C}^2([0, 1]), \quad \max(\|f\|_\infty, \|f'\|_\infty) \leq \|f''\|_\infty + C(|f(x_1)| + |f(x_2)|). \quad (\text{I.3})$$

Q 3. Pour tous $x \in [0, 1]$ et $f \in \mathcal{C}^2([0, 1])$, démontrer l'inégalité

$$\left| f'(x) - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \right| \leq \|f''\|_\infty.$$

Q 4. En déduire que, pour toute fonction $f \in \mathcal{C}^2([0, 1])$, on a $\|f'\|_\infty \leq \|f''\|_\infty + \frac{|f(x_1)| + |f(x_2)|}{x_2 - x_1}$.

Q 5. Conclure le cas $K = 2$ en montrant l'inégalité d'interpolation (I.3) avec $C = 1 + \frac{1}{x_2 - x_1}$.

I.C – Cas général par interpolation de Lagrange

On revient à l'étude du cas général d'inégalité d'interpolation à l'ordre K , donnée par (I.1). On fixe $K \in \mathbb{N}^*$.

Q 6. Démontrer que l'application

$$\Psi : \begin{cases} \mathbb{R}_{K-1}[X] & \rightarrow \mathbb{R}^K \\ P & \mapsto (P(x_1), \dots, P(x_K)) \end{cases}$$

est un isomorphisme d'espaces vectoriels.

Q 7. Montrer qu'il existe K polynômes $L_1, \dots, L_K$ de $\mathbb{R}_{K-1}[X]$ tels que, pour toute fonction $f \in \mathcal{C}^K([0, 1])$, le polynôme $P = \sum_{j=1}^K f(x_j) L_j$ vérifie

$$\forall \ell \in \llbracket 1, K \rrbracket, \quad P(x_\ell) = f(x_\ell).$$

Dans les deux questions suivantes Q8 et Q9, on fixe $f \in \mathcal{C}^K([0, 1])$ et on note P le polynôme déterminé dans la question Q7.

Q 8. Pour tout $k \in \llbracket 0, K-1 \rrbracket$, montrer qu'il existe au moins $K-k$ réels distincts de $[0, 1]$ en lesquels la fonction $f^{(k)} - P^{(k)}$ s'annule.

Q 9. En déduire l'inégalité $\|f^{(k)} - P^{(k)}\|_\infty \leq \|f^{(k+1)} - P^{(k+1)}\|_\infty$ pour tout $k \in \llbracket 0, K-1 \rrbracket$.

Q 10. Montrer qu'il existe une constante $C > 0$ pour laquelle l'inégalité d'interpolation (I.1) est vérifiée.

II Dérivation $\mathcal{C}^K$ pour les séries de fonctions

II.A – Énoncé général

On se propose maintenant de démontrer le résultat annoncé dans le préambule. Soit $K \in \mathbb{N}^*$, on considère

- des réels distincts $x_1 < \dots < x_K$ d'un intervalle $[a, b]$ (avec $a < b$) ;
- une suite de fonctions (f_n) de classe $\mathcal{C}^K$ sur $[a, b]$ à valeurs réelles et vérifiant les deux hypothèses

(H1) la série de fonctions $\sum f_n^{(K)}$ converge normalement sur $[a, b]$;

(H2) pour tout $\ell \in \llbracket 1, K \rrbracket$ la série numérique $\sum f_n(x_\ell)$ est absolument convergente.

Q 11. Dans le cas particulier $[a, b] = [0, 1]$, justifier que la série $\sum f_n^{(k)}$ converge normalement sur $[a, b]$ pour tout $k \in \llbracket 0, K-1 \rrbracket$.

Q 12. Traiter la question précédente dans le cas général d'un segment $[a, b]$ avec $a < b$.

On pourra examiner $f_n \circ \sigma$ où $\sigma : [0, 1] \rightarrow [a, b]$ est définie par $\sigma(t) = (1-t)a + tb$ pour tout $t \in [0, 1]$.

D'après le résultat de la question précédente, on peut poser $F_k(x) = \sum_{n=0}^{+\infty} f_n^{(k)}(x)$ pour tout $x \in [a, b]$.

Q 13. Démontrer que F_0 est de classe $\mathcal{C}^K$ sur $[a, b]$ et que $F_0^{(k)} = F_k$ pour tout $k \in \llbracket 1, K \rrbracket$.

II.B – Application sur un exemple

Dans cette sous-partie, on considère un exemple où les dérivées intermédiaires ne s'expriment pas avec les fonctions usuelles.

Q 14. Pour tout $n \in \mathbb{N}^*$, justifier qu'il existe une unique fonction $f_n \in \mathcal{C}^2(]0, +\infty[)$ vérifiant $f_n(1) = 0$, $f_n(2) = 0$ et $f_n''(x) = (-1)^n 2^{-n} x^2$ pour tout $x > 0$.

Q 15. Montrer que la série de fonctions $\sum f_n(x)$ converge normalement sur tout segment inclus dans $]0, +\infty[$ et que la fonction $F : x \mapsto \sum_{n=1}^{+\infty} f_n(x)$ est de classe $\mathcal{C}^2$ sur $]0, +\infty[$.

Q 16. Expliciter $F''(x)$.

Q 17. Montrer que $|F(x)| \leq \frac{1}{3}$ pour tout $x \in [1, 2]$.

III Convergence d'une série aléatoire de Rademacher

Le but de cette partie est de montrer que, si la série $\sum a_n^2$ converge, alors la série aléatoire $\sum X_n a_n$ converge avec probabilité 1.

Notations

— $(X_n)_{n \in \mathbb{N}}$ désigne une suite de variables aléatoires mutuellement indépendantes d'un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$ vérifiant

$$\forall n \in \mathbb{N}, \quad \mathbb{P}(X_n = -1) = \mathbb{P}(X_n = 1) = \frac{1}{2};$$

— $(a_n)_{n \in \mathbb{N}}$ est une suite réelle telle que la série $\sum a_n^2$ converge ;

— pour tout $N \in \mathbb{N}$, on note $S_N = \sum_{n=0}^N X_n a_n$ la somme partielle au rang N de la série $\sum X_n a_n$;

— si $(\phi(j))_{j \in \mathbb{N}}$ est une suite strictement croissante d'entiers naturels, pour tout entier $j \in \mathbb{N}$ et tout entier $m \in [\phi(j) + 1, \phi(j+1)]$, on note les événements

$$\begin{aligned} A_j &= \{|S_{\phi(j+1)} - S_{\phi(j)}| > 2^{-j}\}, \\ B_j &= \left\{ \max_{\phi(j)+1 \leq n \leq \phi(j+1)} |S_n - S_{\phi(j)}| > 2^{-j} \right\}, \\ B_{j,m} &= \{|S_m - S_{\phi(j)}| > 2^{-j} \text{ et } \forall n \in [\phi(j), m-1], |S_n - S_{\phi(j)}| \leq 2^{-j}\}. \end{aligned}$$

La réalisation de l'évènement $B_{j,m}$ signifie que m est le plus petit entier de l'intervalle $[\phi(j), \phi(j+1)]$ vérifiant $|S_m - S_{\phi(j)}| > 2^{-j}$.

III.A – Construction de la suite $(\phi(j))_{j \in \mathbb{N}}$ et majoration de $\mathbb{P}(A_j)$

Q 18. Justifier l'existence d'une suite strictement croissante d'entiers naturels $(\phi(j))_{j \in \mathbb{N}}$ vérifiant

$$\forall j \in \mathbb{N}, \quad \sum_{n > \phi(j)}^{+\infty} a_n^2 \leq \frac{1}{8^j}.$$

On fixe désormais une telle suite $(\phi(j))_{j \in \mathbb{N}}$.

Q 19. Exprimer l'espérance et la variance de $S_{\phi(j+1)} - S_{\phi(j)}$ en fonction des termes de la suite $(a_n)_{n \in \mathbb{N}}$.

Q 20. Dédurre des deux questions précédentes la majoration $\mathbb{P}(A_j) \leq 2^{-j}$.

III.B – Inégalité maximale de Lévy $\mathbb{P}(B_j) \leq 2\mathbb{P}(A_j)$

Q 21. Pour tout $j \in \mathbb{N}$, démontrer que les événements $B_{j,m}$, pour m parcourant $[\phi(j) + 1, \phi(j+1)]$, sont disjoints deux à deux et qu'on a l'égalité d'évènements

$$B_j = \bigcup_{\phi(j) < m \leq \phi(j+1)} B_{j,m}.$$

Q 22. Expliquer comment en déduire la formule $\mathbb{P}(A_j) = \sum_{m=\phi(j)+1}^{\phi(j+1)} \mathbb{P}(A_j \cap B_{j,m})$.

Q 23. Soit $m \in [\phi(j) + 1, \phi(j+1)]$, montrer que la fonction

$$\begin{aligned} \mathbb{R} &\rightarrow \mathbb{R} \\ \alpha &\mapsto 2^{\phi(j+1)-\phi(j)} \mathbb{P}\left(\left\{|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}\right\} \cap B_{j,m}\right) \end{aligned}$$

est à valeurs dans $\mathbb{N}$ et est paire.

Q 24. Prouver que si l'évènement B_j se réalise, alors il existe $m \in [\phi(j) + 1, \phi(j+1)]$ et $\alpha \in \{-1, +1\}$ tels que l'évènement

$$\left\{|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}\right\} \cap B_{j,m}$$

se réalise également.

On pourra exprimer $S_m - S_{\phi(j)}$ en fonction des deux nombres $\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}$ avec $\alpha = \pm 1$.

Q 25. En déduire que

$$\mathbb{P}(B_j) \leq 2\mathbb{P}(A_j)$$

III.C – Convergence de la série aléatoire $\sum X_n a_n$

Q 26. On note B l'évènement $\bigcap_{J \in \mathbb{N}} \bigcup_{j \geq J} B_j$. Montrer l'égalité $\mathbb{P}(B) = 0$.

Q 27. Montrer que l'évènement

$$\{\exists J \in \mathbb{N}, \quad \forall j \geq J, \quad \forall n \in [\phi(j) + 1, \phi(j + 1)], \quad |S_n - S_{\phi(j)}| \leq 2^{-j}\}$$

se réalise avec probabilité 1.

Q 28. En déduire que l'évènement

$$\{\text{la suite } (S_{\phi(j)})_{j \in \mathbb{N}} \text{ est convergente}\}$$

a également une probabilité 1.

On pourra examiner la série $\sum |S_{\phi(j+1)} - S_{\phi(j)}|$.

Q 29. Conclure que l'évènement

$$\{\text{la série } \sum X_n a_n \text{ est convergente}\}$$

a une probabilité 1.

IV Dérivation $\mathcal{C}^K$ pour des séries aléatoires de fonctions

On fixe $K \in \mathbb{N}^*$ et on considère

- une suite de variables aléatoires $(X_n)_{n \in \mathbb{N}}$ vérifiant les hypothèses de la partie précédente ;
- des réels distincts $x_1 < \dots < x_K$ de $[0, 1]$;
- une suite de fonctions (f_n) de classe $\mathcal{C}^K$ sur $[0, 1]$ à valeurs réelles et vérifiant les deux hypothèses
 - (H1) la série de fonctions $\sum f_n^{(K)}$ converge normalement sur $[0, 1]$;
 - (H2') pour tout $\ell \in \llbracket 1, K \rrbracket$, la série numérique $\sum f_n(x_\ell)^2$ est convergente.

Q 30. Montrer que l'une des deux hypothèses (H2') ou (H2) (étudiée dans la partie II) implique l'autre.

Q 31. Montrer que l'évènement

$$\{\text{pour tout } \ell \in \llbracket 1, K \rrbracket, \text{ la série } \sum X_n f_n(x_\ell) \text{ est convergente}\}$$

a une probabilité 1.

Q 32. On note $P_n \in \mathbb{R}_{K-1}[X]$ un polynôme vérifiant $P_n(x_\ell) = f_n(x_\ell)$ pour tout $\ell \in \llbracket 1, K \rrbracket$ (cf. question 7), montrer que l'évènement

$$\left\{ \begin{array}{l} \text{pour tout } k \in \llbracket 0, K \rrbracket, \text{ la série de fonctions } \sum X_n (f_n - P_n)^{(k)} \text{ est uniformément convergente sur } [0, 1], \\ \text{la fonction } \sum_{n=0}^{+\infty} X_n (f_n - P_n) \text{ est de classe } \mathcal{C}^K, \\ \text{pour tout } k \in \llbracket 0, K \rrbracket, \left(\sum_{n=0}^{+\infty} X_n (f_n - P_n) \right)^{(k)} = \sum_{n=0}^{+\infty} X_n (f_n - P_n)^{(k)} \end{array} \right\}$$

a une probabilité 1.

Q 33. Montrer que l'évènement

$$\left\{ \begin{array}{l} \text{pour tout } k \in \llbracket 0, K \rrbracket, \text{ la série de fonctions } \sum X_n f_n^{(k)} \text{ est uniformément convergente sur } [0, 1], \\ \text{la fonction } \sum_{n=0}^{+\infty} X_n f_n \text{ est de classe } \mathcal{C}^K, \\ \text{pour tout } k \in \llbracket 0, K \rrbracket, \left(\sum_{n=0}^{+\infty} X_n f_n \right)^{(k)} = \sum_{n=0}^{+\infty} X_n f_n^{(k)} \end{array} \right\}$$

a une probabilité 1.

Q 34. Donner un exemple d'entier $K \in \mathbb{N}^*$ pour lequel l'évènement précédent se réalise avec les fonctions f_n définies par

$$\begin{cases} f_0 = 0, \\ f_n(x) = \ln \left(1 + \sin \left(\frac{x}{n} \right) \right) \quad \forall n \in \mathbb{N}^*, \forall x \in [0, 1]. \end{cases}$$

• • • FIN • • •

SOLUTION

Inégalités d'interpolation des dérivées

A. Cas particulier $K = 1$

1. Soit $f \in C^1([0, 1])$. Soit $x \in [0, 1]$. Comme f est de classe C^1 sur $[0, 1]$ on peut appliquer l'inégalité des accroissements finis à f entre x et x_1 :

$$|f(x) - f(x_1)| \leq |x - x_1| \|f'\|_\infty \leq \|f'\|_\infty$$

Par inégalité triangulaire (seconde forme), on a ainsi

$$|f(x)| - |f(x_1)| \leq \|f'\|_\infty$$

c'est à dire

$$|f(x)| \leq |f(x_1)| + \|f'\|_\infty$$

La majorant étant valable pour tout x , un passage à la borne supérieure donne (on a trouvé un majorant qui est donc plus grand que la borne supérieure qui est le plus petit des majorants)

$$\boxed{\|f\|_\infty \leq \|f'\|_\infty + |f(x_1)|}$$

On a ici utilisé les notations $\|f\|_\infty$ et $\|f'\|_\infty$ sans justifier que ces quantités sont bien définies. Selon le rapport, il était apprécié que l'on précise l'existence de ces quantités par un théorème de compacité (une fonction continue sur un compact, ici un segment, est bornée). L'oubli de cette précision (il nous semble que c'est un oubli très bénin) n'était cependant pas vraiment sanctionné.

2. Si on considère la fonction f constante égale à 1, on a $|f(x_1)| = \|f\|_\infty = 1$ et $\|f'\|_\infty = 0$.

$$\boxed{\text{L'inégalité (I.2) est fausse avec } C \in]0, 1[\text{ dans le cas général}}$$

B. Cas particulier $K = 2$

3. Par égalité des accroissements finis appliquée à f (de classe C^1) entre x_1 et x_2 :

$$\text{il existe } y \in]x_1, x_2[\text{ tel que } \frac{f(x_2) - f(x_1)}{x_2 - x_1} = f'(y)$$

Par inégalité des accroissements finis appliquée à f' (de classe C^1) entre x et y :

$$|f'(x) - f'(y)| \leq |x - y| \|f''\|_\infty \leq \|f''\|_\infty$$

On a ainsi montré, en remplaçant $f'(y)$ par son expression, que

$$\boxed{\left| f'(x) - \frac{f(x_2) - f(x_1)}{x_2 - x_1} \right| \leq \|f''\|_\infty}$$

4. Par seconde forme de l'inégalité triangulaire,

$$|f'(x)| - \left| \frac{f(x_2) - f(x_1)}{x_2 - x_1} \right| \leq \|f''\|_\infty$$

Puis par inégalité triangulaire

$$|f'(x)| \leq \frac{|f(x_2)| + |f(x_1)|}{x_2 - x_1} + \|f''\|_\infty$$

On a un majorant de $|f'|$ sur $[0, 1]$ qui est donc plus grand que la borne supérieure :

$$\|f'\|_\infty \leq \frac{|f(x_2)| + |f(x_1)|}{x_2 - x_1} + \|f''\|_\infty$$

On cherche dans cette question à majorer $\|f'\|_\infty$, c'est à dire à majorer une borne supérieure. La méthode générale pour majorer $\sup(A)$ consiste à trouver une constante majorant tous les éléments de A . C'est bien ce que l'on fait ici en majorant $|f'(x)|$ par une quantité indépendante de x . Cela donne simultanément l'existence de la borne supérieure et sa majoration.

5. Posons $C = 1 + \frac{1}{x_2 - x_1}$. La question précédente donne

$$\|f'\|_\infty \leq \|f''\|_\infty + (C - 1)(|f(x_1)| + |f(x_2)|)$$

Mais avec la question 1, $\|f\|_\infty \leq \|f'\|_\infty + |f(x_1)| \leq \|f'\|_\infty + (|f(x_1)| + |f(x_2)|)$. On a donc

$$\|f\|_\infty \leq \|f''\|_\infty + C(|f(x_1)| + |f(x_2)|)$$

Comme on a aussi (a fortiori avec ce qui précède) $\|f'\|_\infty \leq \|f''\|_\infty + C(|f(x_1)| + |f(x_2)|)$ (on a ajouté un terme positif au majorant précédent), on en déduit que

$$\max(\|f\|_\infty, \|f'\|_\infty) \leq \|f''\|_\infty + C(|f(x_1)| + |f(x_2)|) \text{ avec } C = 1 + \frac{1}{x_2 - x_1}$$

C. Cas général par interpolation de Lagrange

6. Ψ est immédiatement linéaire ($\Psi(\lambda P + Q) = \lambda \Psi(P) + \Psi(Q)$).

De plus, si $P \in \ker(\Psi)$ alors P est de degré $\leq K - 1$ et possède au moins K racines (les x_i) et est donc nul. Ψ est injective (on a montré que $\ker(\Psi) \subset \{0\}$ et l'autre inclusion est toujours vraie).

Comme $\mathbb{R}_{K-1}[X]$ et $\mathbb{R}^K$ ont même dimension K , on peut conclure que

Ψ est un isomorphisme d'espaces vectoriels

Il fallait bien sûr mentionner la linéarité avant d'utiliser le noyau. Le rapport nous dit que l'argument " Ψ est trivialement linéaire" était insuffisant. Nous ne voyons pas vraiment ce qu'apporterait des justification supplémentaire (la linéarité est vraiment immédiate). L'argument de dimension doit lui être clairement exposé ("on est en dimension finie" ne suffit à l'évidence pas). Notez qu'il est possible de prouver la surjectivité en exhibant un antécédent à l'élément $(a_1, \dots, a_K)$ à l'aide des polynômes de Lagrange :

$$\Psi^{-1}(a_1, \dots, a_K) = \sum_{j=1}^K \left(a_j \prod_{i \in \llbracket 1, K \rrbracket \setminus \{j\}} \frac{X - x_i}{x_j - x_i} \right)$$

Au vu de la seconde question, c'est cependant plutôt la preuve théorique qui était attendue.

7. Posons

$$\forall j \in \llbracket 1, K \rrbracket, L_j = \prod_{\substack{1 \leq i \leq K \\ i \neq j}} \frac{X - x_i}{x_j - x_i}$$

On vérifie immédiatement que $\forall i, j \in \llbracket 1, K \rrbracket, L_j(x_i) = \delta_{i,j}$ (symbole de Kronecker valant 1 si $i = j$ et 0 sinon) c'est à dire que $\Psi(L_j)$ est le j -ème vecteur de la base canonique de $\mathbb{R}^K$. On a alors avec les notations de l'énoncé

$$P(x_\ell) = \sum_{j=1}^K f(x_j) L_j(x_\ell) = f(x_\ell)$$

$$\forall \ell \in \llbracket 1, K \rrbracket, P(x_\ell) = f(x_\ell) \text{ où } P = \sum_{j=1}^K f(x_j) L_j$$

Dans l'esprit de l'énoncé, on pourrait se contenter de dire que l'on note $(L_1, \dots, L_K)$ l'image réciproque par Ψ de la base canonique de $\mathbb{R}^K$ et que c'est une base de $\mathbb{R}_{K-1}[X]$ car Ψ est un isomorphisme. On n'a pas besoin de donner une expression explicite des L_k . Cependant, il faut connaître l'expression de ces polynômes de Lagrange associés aux x_i . Voir la remarque en fin de question 6.

8. Montrons par récurrence finie (sur k) que

$$\forall k \in \llbracket 0, K-1 \rrbracket, f^{(k)} - P^{(k)} \text{ s'annule en au moins } K-k \text{ éléments de } [0, 1]$$

- Par choix de P , la question précédente montre que $f^{(0)} - P^{(0)} = f - P$ est nulle en tous les x_i et a donc au moins K zéros dans $[0, 1]$. Ceci montre le résultat pour $k = 0$.
- Supposons le résultat acquis à un rang $k \in \llbracket 0, K-2 \rrbracket$ (récurrence forte même si cela est inutile). $(f - P)^{(k)} = f^{(k)} - P^{(k)}$ admet au moins $K-k$ zéros $0 \leq y_1 < \dots < y_{K-k} \leq 1$ par hypothèse de récurrence.
Par théorème de Rolle, la dérivée de cette fonction s'annule sur chaque intervalle $]y_i, y_{i+1}[$ (f est de classe C^1 sur $[y_i, y_{i+1}]$ et prend la même valeur aux deux bornes). Ainsi $(f - P)^{(k+1)} = f^{(k+1)} - P^{(k+1)}$ admet au moins $K-k-1$ zéros dans $[0, 1]$ (les intervalles $]y_i, y_{i+1}[$ sont disjoints). Ceci montre le résultat au rang $k+1$.

Ici, on met en oeuvre une récurrence finie. Lors de l'étape héréditaire, quand on montre que l'hypothèse au rang k entraîne celle au rang $k+1$, il faut supposer $k \leq K-2$. Notez que l'on a ici utilisé une récurrence généralisée (on a supposé le résultat vrai jusqu'au rang k). C'est inutile dans cette question mais cela peut être un réflexe à adopter. Pour rassurer le correcteur, je précise cela dans la rédaction. Plus important : si vous raisonnez par récurrence, vous devez commencer par l'annoncer et donner clairement et complètement l'hypothèse de récurrence.

9. Soit $k \in \llbracket 0, K-1 \rrbracket$. La fonction $f^{(k)} - P^{(k)}$ est de classe C^1 sur $[0, 1]$ et s'annule en au moins un point (question 8).

La relation I.2 utilisée avec cette fonction et ce point donne $\|f^{(k)} - P^{(k)}\|_\infty \leq \|f^{(k+1)} - P^{(k+1)}\|_\infty$.

$$\forall k \in \llbracket 0, K-1 \rrbracket, \|f^{(k)} - P^{(k)}\|_\infty \leq \|f^{(k+1)} - P^{(k+1)}\|_\infty$$

10. La question précédente montre que la suite finie $(\|f^{(k)} - P^{(k)}\|_\infty)_{0 \leq k \leq K}$ est croissante. Ses termes sont donc tous plus petits que celui au rang K .
Or, $P^{(K)} = 0$ car P est de degré $\leq K - 1$ et ce terme au rang K vaut $\|f^{(K)}\|_\infty$. On a donc

$$\forall k \in \llbracket 0, K \rrbracket, \|f^{(k)} - P^{(k)}\|_\infty \leq \|f^{(K)}\|_\infty$$

Par inégalité triangulaire pour la norme infinie,

$$\forall k \in \llbracket 0, K \rrbracket, \|f^{(k)}\|_\infty \leq \|P^{(k)}\|_\infty + \|f^{(K)}\|_\infty$$

Par ailleurs, on a aussi

$$\begin{aligned} \forall k \in \llbracket 0, K - 1 \rrbracket, \|P^{(k)}\|_\infty &= \left\| \sum_{j=1}^K f(x_j) L_j^{(k)} \right\|_\infty \leq \sum_{j=1}^K |f(x_j)| \|L_j^{(k)}\|_\infty \\ &\leq \max_{1 \leq j \leq K} \|L_j^{(k)}\|_\infty \sum_{j=1}^K |f(x_j)| \end{aligned}$$

En notant C le maximum des $\|L_j^{(k)}\|_\infty$ pour $j \in \llbracket 1, K \rrbracket$ et $k \in \llbracket 0, K - 1 \rrbracket$, on a a fortiori

$$\forall k \in \llbracket 0, K - 1 \rrbracket, \|f^{(k)}\|_\infty \leq \|f^{(K)}\|_\infty + C \sum_{j=1}^K |f(x_j)|$$

et la constante C ne dépend pas de f .

$$\exists C, \forall f \in C^K([0, 1]), \max_{0 \leq k \leq K-1} \|f^{(k)}\|_\infty \leq \|f^{(K)}\|_\infty + C \sum_{j=1}^K |f(x_j)|$$

Remarque : d'autres constantes C sont bien sûr envisageables. Il importe de bien noter que celle obtenue est valable pour toute fonction f , c'est à dire qu'elle ne dépend pas de f (de manière directe ou cachée). Ici, C ne dépend que des L_j qui ne dépendent eux même que des x_i .

Dérivation C^K pour les séries de fonctions

A. Enoncé général

11. Soit $k \in \llbracket 0, K - 1 \rrbracket$. L'inégalité d'interpolation prouvée en première partie donne

$$\|f_n^{(k)}\|_\infty \leq \|f_n^{(K)}\|_\infty + C \sum_{j=1}^K |f_n(x_j)|$$

Dans le membre de droite, $\|f_n^{(K)}\|_\infty$ et les $|f_n(x_j)|$ sont les termes généraux de séries positives convergentes. Une combinaison de telles séries converge absolument. Par comparaison des séries positives, $\sum \|f_n^{(k)}\|_\infty$ converge aussi. Ceci signifie que

toutes les séries dérivées convergent normalement sur $[0, 1]$

12. On pose $g_n : t \in [0, 1] \mapsto f_n((1-t)a + tb)$ qui est bien définie car $(1-t)a + tb \in [a, b]$ pour $t \in [0, 1]$. Quand t décrit $[0, 1]$, $(1-t)a + tb$ décrit $[a, b]$ (c'est même la définition géométrique de ce segment). Ainsi

$$\|g_n\|_\infty = \|f_n\|_{\infty, [a, b]}$$

Remarquons que

$$\forall x \in [a, b], f_n(x) = g_n\left(\frac{x-a}{b-a}\right)$$

et posons

$$\forall \ell \in \llbracket 1, K \rrbracket, t_\ell = \frac{x_\ell - a}{b - a}$$

On a ainsi $0 \leq t_1 < \dots < t_K \leq 1$ et $\forall \ell, f_n(x_\ell) = g_n(t_\ell)$. De plus,

- Pour tout n , g_n est de classe C^K sur $[0, 1]$ comme composée d'une fonction de classe C^K et d'une fonction de classe C^∞ . De plus

$$\forall k \in \llbracket 0, K \rrbracket, g_n^{(k)}(t) = (b-a)^k f_n^{(k)}((1-t)a + tb)$$

- Pour tout $\ell \in \llbracket 1, K \rrbracket, \sum(g_n(t_\ell)) = \sum(f_n(x_\ell))$ converge absolument.
- $\|g_n^{(K)}\|_\infty = (b-a)^K \|f_n^{(K)}\|_{\infty, [a, b]}$ est le terme général d'une série convergente. $\sum(g_n^{(K)})$ converge donc normalement sur $[0, 1]$.

La question précédente donne la convergence normale sur $[0, 1]$ de toutes les séries dérivées de g_n . Mais comme $\|g_n^{(k)}\|_\infty = (b-a)^k \|f_n^{(k)}\|_{\infty, [a, b]}$, on a aussi la convergence normale sur $[a, b]$ de toutes les séries dérivées de f_n .

Le résultat est vrai dans le cas d'un segment $[a, b]$ quelconque

Remarque : nous avons utilisé l'indication de l'énoncé mais sans introduire σ . Par ailleurs, l'expression de $g_n^{(k)}$ à l'aide de $f_n^{(k)}$ a été donnée sans justification. C'est raisonnable à ce niveau du problème (une récurrence aurait pu être évoquée) d'autant plus que cette expression n'est pas donnée par l'énoncé. Cependant, si vous avez un doute sur la formule, il faut la vérifier au moins au brouillon.

13. On a a fortiori la convergence simple de $\sum(f_n), \dots, \sum(f_n^{(K-1)})$ sur $[a, b]$ et la convergence normale donc uniforme de $\sum(f_n^{(K)})$ sur $[a, b]$. Le théorème du cours de régularité des sommes de séries de fonctions indique que

$$F_0 \in C^K([a, b]) \text{ et } \forall k \in \llbracket 1, K \rrbracket, F_0^{(k)} = F_k$$

B. Application sur un exemple

14. Existence : notons $h_n : x > 0 \mapsto (-1)^n 2^{-nx^2}$. Par théorèmes d'opérations, h_n est continue sur $\mathbb{R}^{+*}$ et admet une primitive i_n qui s'annule en 1 (par théorème fondamental appliqué à h_n sur l'intervalle $\mathbb{R}^{+*}$ qui contient 1). A son tour, i_n admet une primitive j_n qui s'annule en 1. Pour tout choix de α et β , la fonction

$$k_{\alpha, \beta} : x \mapsto j_n(x) + \alpha(x-1) + \beta$$

vérifie $k''_{\alpha,\beta} = h_n$. En outre, $k_{\alpha,\beta}(1) = \beta$ et $k_{\alpha,\beta}(2) = j_n(2) + \alpha + \beta$. Si on choisit $\beta = 0$ et $\alpha = -j_n(2)$, on trouve une fonction $f_n = k_{-j_n(2),0}$ qui respecte les conditions demandées.

Unicité : supposons que g_n soit une autre fonction convenable. $d_n = f_n - g_n$ est alors à dérivée seconde nulle sur $\mathbb{R}^{+*}$ et c'est donc une fonction affine sur cet intervalle. La nullité en 1 et 2 entraîne la nullité de cette fonction affine et $f_n = g_n$ ce qui donne l'unicité de f_n convenable.

Il existe une unique $f_n \in C^2(\mathbb{R}^{+*})$, $f_n(1) = f_n(2) = 0$ et $\forall x > 0$, $f_n'' 2^{-nx^2}$

Remarque : pour prouver un résultat du type existence et unicité, il y a a priori deux méthodes possibles. La première est celle utilisée ci-dessus (on trouve miraculeusement un objet convenable puis on suppose que deux objets conviennent et on montre qu'ils sont égaux). La seconde utilise un raisonnement par conditions nécessaires puis suffisantes (on montre que si un objet convient il est forcément égal à quelquechose, c'est l'unicité si existence, puis il suffit de vérifier que ce quelquechose fonctionne). Il est très fortement déconseillé de tenter de raisonner par équivalence. Et il convient de respecter l'une des deux méthodes qui précède.

15. Soient a, b tels que $0 < a < 1 < 2 < b$. f_n s'annule en $x_1 = 1$ et $x_2 = 2$ pour tout n et $\sum(f_n(x_1))$ ainsi que $\sum(f_n(x_2))$ convergent. De plus

$$\forall x \in [a, b], |f_n'' - nx^2| \leq 2^{-na^2}$$

et ainsi $\|f_n''\|_{\infty, [a, b]} \leq 2^{-na^2}$ qui est le terme général d'une série convergente (géométrique de raison $2^{-a^2} \in [0, 1[$).

On a donc convergence normale de $\sum(f_n'')$ sur $[a, b]$. La partie II.A s'applique et indique que $\sum(f_n)$ converge normalement sur $[a, b]$ et que sa somme est de classe C^2 sur $[a, b]$. Tout segment de $\mathbb{R}^{+*}$ pouvant être inclus dans un segment du type précédent on en déduit que

$\sum(f_n)$ converge normalement sur tout segment de $\mathbb{R}^{+*}$ et $F = \sum_{k=1}^{\infty} f_k \in C^2(\mathbb{R}^{+*})$

Remarque : être de classe C^2 sur tout segment de $\mathbb{R}^{+*}$ entraîne être de classe C^2 sur $\mathbb{R}^{+*}$ car dérivabilité et continuité sont des notions locales (dire que f est continue sur un intervalle I , c'est dire qu'elle est continue en chaque point et il suffit de regarder ce qui se passe sur un petit voisinage du point). Peut-être pourrait-on insister sur ce point dans la rédaction. On a quand même bien précisé que l'on ne travaillait pas avec un segment quelconque mais contenant 1 et 2, ce qui suffit avec la remarque "tout segment de $\mathbb{R}^{+*}$ pouvant être inclus dans un segment du type précédent".

16. La même partie II.A donne

$$\forall x > 0, F''(x) = \sum_{n=1}^{\infty} (-2^{-x^2})^n$$

et on reconnaît une somme géométrique. Ainsi

$$\forall x > 0, F''(x) = -\frac{2^{-x^2}}{1+2^{-x^2}} = -1 + \frac{1}{1+2^{-x^2}}$$

17. Si $x \in [1, 2]$, $2^{-x^2} = \exp(-x^2 \ln(2)) \in [1/16, 1/2]$ puis $1 + 2^{-x^2} \in [17/16, 3/2]$. Ainsi

$$\forall x \in [1, 2], \quad -\frac{1}{3} \leq F''(x) \leq -\frac{1}{17}$$

On en déduit que $\|F''\|_{\infty, [1, 2]} \leq \frac{1}{3}$. Pour pouvoir utiliser (I.1), on se ramène au segment $[0, 1]$ en considérant $G : x \mapsto F(x + 1)$. G est de classe C^2 sur $[0, 1]$, nulle en 0 et 1 et comme $G'''(x) = F'''(x + 1)$, on a $\|G'''\|_{\infty} \leq 1/3$. Ainsi, $\|G\|_{\infty} \leq 1/3$ avec (I.1). En revenant à F , on a montré que

$$\boxed{\forall x \in [1, 2], \quad |F(x)| \leq \frac{1}{3}}$$

Le passage par la fonction auxiliaire G est indispensable pour se ramener à (I.1).

Convergence d'une série aléatoire de Rademacher

A. Construction de la suite $(\phi(j))$ et majoration de $\mathbb{P}(A_j)$

18. On construit la suite $(\phi(j))$ par récurrence. Pour cela, on remarque que comme $\sum(a_n^2)$ converge, la suite de terme général $R_p = \sum_{n=p+1}^{\infty} a_n^2$ est bien définie et de limite nulle quand $p \rightarrow +\infty$.

- La convergence précédente donne un rang $\phi(0)$ à partir duquel $0 \leq R_n \leq 1 = \frac{1}{8^0}$. On a en particulier $R_{\phi(0)} \leq \frac{1}{8^0}$.
- Supposons construits $\phi(0) < \dots < \phi(j)$ pour un $j \geq 0$. La convergence évoquée donne un rang N tel que si $p \geq N$, $R_p \leq \frac{1}{8^{j+1}}$. On choisit alors $\phi(j+1) = \max(N, \phi(j) + 1)$ (ce qui assure la stricte croissance et la propriété de majoration).

$$\boxed{(\phi(j))_{j \in \mathbb{N}} \text{ est telle que } \forall j \in \mathbb{N}, \quad \sum_{n=\phi(j)+1}^{+\infty} a_n^2 \leq \frac{1}{8^j}}$$

Remarque : dans la construction, il faut penser à imposer la stricte croissance. Cela explique l'utilisation d'un maximum dans l'étape de récurrence.

19. Remarquons que pour tout j , on a

$$\mathbb{E}(X_j) = -\mathbb{P}(X_i = -1) + \mathbb{P}(X_i = 1) = 0$$

De plus X_j^2 est constante égale à 1 et donc d'espérance 1. Ainsi (formule de König)

$$\mathbb{V}(X_j) = \mathbb{E}(X_j^2) - \mathbb{E}(X_j)^2 = 1$$

Par linéarité de l'espérance, on a

$$\mathbb{E}(S_{\phi(j+1)} - S_{\phi(j)}) = \sum_{n=\phi(j)+1}^{\phi(j+1)} a_n \mathbb{E}(X_j) = 0$$

La variance d'une somme de variables indépendantes (deux à deux suffit, ici on a même indépendance mutuelle) étant la somme des variances, on a aussi

$$\mathbb{V}(S_{\phi(j+1)} - S_{\phi(j)}) = \sum_{n=\phi(j)+1}^{\phi(j+1)} \mathbb{V}(a_n X_j) = \sum_{n=\phi(j)+1}^{\phi(j+1)} a_n^2 \mathbb{V}(X_j)$$

$$\forall j, \mathbb{E}(S_{\phi(j+1)} - S_{\phi(j)}) = 0 \text{ et } \mathbb{V}(S_{\phi(j+1)} - S_{\phi(j)}) = \sum_{n=\phi(j)+1}^{\phi(j+1)} a_j^2$$

20. On applique l'inégalité de Tchebychev à la variable $Z_j = S_{\phi(j+1)} - S_{\phi(j)}$ qui est d'espérance nulle et admet un moment d'ordre 2 :

$$\mathbb{P}(A_j) = \mathbb{P}(|Z_j - \mathbb{E}(Z_j)| > 2^{-j}) \leq \frac{\mathbb{V}(Z_j)}{2^{-2j}}$$

On remarque que

$$\mathbb{V}(Z_j) = \sum_{n=\phi(j)+1}^{\phi(j+1)} a_j^2 \leq \sum_{n=\phi(j)+1}^{+\infty} a_j^2 \leq \frac{1}{8j}$$

On en déduit donc que

$$\forall j, \mathbb{P}(A_j) \leq 2^{-j}$$

Remarque : il est souvent rappelé dans les rapports de concours que l'utilisation des inégalités de Markov ou Tchebychev doit être précédée d'une vérification des hypothèses.

B. Inégalité maximale de Lévy $\mathbb{P}(B_j) \leq 2\mathbb{P}(A_j)$.

Remarque : dans la définition de l'évènement $B_{j,m}$, l'énoncé parle de l'intervalle $[\phi(j), m-1]$ en incluant $\phi(j)$. On peut tout aussi bien l'exclure car $|S_{\phi(j)} - S_{\phi(j)}| = 0 \leq 2^{-j}$. Dans la suite, on ne reviendra plus sur cette remarque.

21. On veut prouver une égalité entre ensembles et on procède donc par double inclusion.

- Soit $\omega \in B_j$. On a alors

$$\max_{\phi(j)+1 \leq n \leq \phi(j+1)} |S_n(\omega) - S_{\phi(j)}(\omega)| > 2^{-j}$$

Ainsi, l'ensemble $E = \{n \in [\phi(j) + 1, \phi(j + 1)], |S_n(\omega) - S_{\phi(j)}(\omega)| > 2^{-j}\}$ est non vide.

Il est fini et possède donc un minimum $m \in [\phi(j) + 1, \phi(j + 1)]$.

Par définition du minimum, un élément $< m$ n'est pas dans E mais m l'est. On a donc $|S_n(\omega) - S_{\phi(j)}(\omega)| \leq 2^{-j}$ quand $\phi(j)+1 \leq n \leq m-1$ et $|S_m(\omega) - S_{\phi(j)}(\omega)| > 2^{-j}$ et donc $\omega \in B_{j,m}$.

- Réciproquement, si ω est dans l'un des $B_{j,m}$ avec $m \in [\phi(j) + 1, \phi(j + 1)]$ alors $|S_m(\omega) - S_{\phi(j)}(\omega)| > 2^{-j}$ et $\{|S_n(\omega) - S_{\phi(j)}(\omega)|, n \in [\phi(j) + 1, \phi(j + 1)]\}$ est de maximum $> 2^{-j}$ (puisque l'un de ses éléments est $> 2^{-j}$) i.e. $\omega \in B_j$.

Par double inclusion, on a montré que

$$B_j = \bigcup_{\phi(j) < m \leq \phi(j+1)} B_{j,m}$$

Soient $m, \ell \in [\phi(j) + 1, \phi(j + 1)]$ distincts. Par symétrie des rôles, on peut supposer que $m < \ell$. Si $\omega \in B_{j,\ell}$ alors, puisque $\phi(j) + 1 \leq m < \ell$, $|S_m(\omega) - S_{\phi(j)}(\omega)| \leq 2^{-j}$ et $\omega \notin B_{j,m}$. On a montré que

$$\text{Les } B_{j,m} \text{ pour } m \in [\phi(j) + 1, \phi(j + 1)] \text{ sont disjoints deux à deux}$$

Remarque : là encore, il ne faut pas essayer de raisonner par équivalence pour prouver l'égalité. Et ne pas essayer de tout prouver en même temps.

22. On a $A_j \subset B_j$ (si un élément d'un ensemble est strictement plus grand que 2^{-j} alors le maximum des éléments de l'ensemble l'est aussi). Or, les $B_{j,m}$ forment une partition de B_j . On a donc

$$A_j = A_j \cap B_j = A_j \cap \bigcup_{m=\phi(j)+1}^{\phi(j+1)} B_{j,m} = \bigcup_{m=\phi(j)+1}^{\phi(j+1)} A_j \cap B_{j,m}$$

La réunion est disjointe avec la question précédente et la σ -additivité de $\mathbb{P}$ donne

$$\mathbb{P}(A_j) = \sum_{m=\phi(j)+1}^{\phi(j+1)} \mathbb{P}(A_j \cap B_{j,m})$$

23. On fixe $j \in \mathbb{N}$, $m \in \llbracket \phi(j) + 1, \phi(j+1) \rrbracket$ et $\alpha \in \mathbb{R}$. L'évènement

$$E_{\alpha,m,j} = (|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}) \cap B_{j,m}$$

ne dépend que des variables $X_{\phi(j)+1}, \dots, X_{\phi(j+1)}$. On peut donc comme l'écrire comme réunion d'évènements du type

$$(X_{\phi(j)+1}, \dots, X_{\phi(j+1)}) = (\varepsilon_{\phi(j)+1}, \dots, \varepsilon_{\phi(j+1)})$$

où les ε_k valent ± 1 . Ces évènements sont disjoints et chacun est de probabilité $\frac{1}{2^{\phi(j+1)-\phi(j)}}$ (avec l'indépendance des X_k). On en déduit que $E_{\alpha,m,j}$ est de probabilité du type $\frac{K}{2^{\phi(j+1)-\phi(j)}}$ avec $K \in \mathbb{N}$, K étant le nombre de uplets $(\varepsilon_{\phi(j)+1}, \dots, \varepsilon_{\phi(j+1)})$ convenables.

$$2^{\phi(j+1)-\phi(j)} \mathbb{P}((|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}) \cap B_{j,m}) \text{ est un entier}$$

Par ailleurs, on remarque que

$$\begin{aligned} (e_{\phi(j)+1}, \dots, e_m, e_{m+1}, \dots, e_{\phi(j+1)}) &\in E_{\alpha,m,j} \\ \iff (e_{\phi(j)+1}, \dots, e_m, -e_{m+1}, \dots, -e_{\phi(j+1)}) &\in E_{-\alpha,m,j} \end{aligned}$$

ce qui montre que $E_{\alpha,m,j}$ et $E_{-\alpha,m,j}$ sont en bijection et ont donc même cardinal. La constante K évoquée ci-dessus est ainsi la même pour α et pour $-\alpha$ et les probabilités de $E_{\alpha,m,j}$ et $E_{-\alpha,m,j}$ sont ainsi égales. On a donc

$$\alpha \mapsto 2^{\phi(j+1)-\phi(j)} \mathbb{P}((|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}) \cap B_{j,m}) \text{ est paire}$$

Remarque : il faut lire l'énoncé attentivement. C'est la fonction qui est paire et pas les valeurs prises par celle-ci. En fait, on peut aussi montrer que l'image de α est un entier pair (car si un uplet convient alors son opposé convient aussi et deux uplets opposés sont différents). Mais ce n'est pas la question et cela ne sert à rien.

24. Supposons que $\omega \in B_j$. La question 21 donne l'existence d'un unique $m \in \llbracket \phi(j) + 1, \phi(j+1) \rrbracket$ tel que $\omega \in B_{j,m}$.

Notons $Z_\alpha = \alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}$. En exprimant Z_1 et Z_{-1} , on trouve aisément que

$$S_m - S_{\phi(j)} = \frac{1}{2}(Z_1 + Z_{-1})$$

Comme $\omega \in B_{j,m}$, $|S_m(\omega) - S_{\phi(j)}(\omega)| > 2^{-j}$. On en déduit par inégalité triangulaire que

$$2^{1-j} < |Z_1(\omega)| + |Z_{-1}(\omega)|$$

Si, par l'absurde, on avait $|Z_1(\omega)|$ et $|Z_{-1}(\omega)|$ tous deux $\leq 2^{-j}$, la somme serait $\leq 2^{1-j}$ et ce serait contradictoire. On a donc $|Z_1(\omega)|$ ou $|Z_{-1}(\omega)|$ qui est $> 2^{-j}$.

$$B_j \subset \bigcup_{\substack{\phi(j)+1 \leq m \leq \phi(j+1) \\ \alpha \in \{1, -1\}}} (|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j}) \cap B_{j,m}$$

25. Par croissance de $\mathbb{P}$ puis inégalité de Boole, on a donc

$$\mathbb{P}(B_j) \leq \sum_{\substack{\phi(j)+1 \leq m \leq \phi(j+1) \\ \alpha \in \{1, -1\}}} \mathbb{P}(|\alpha S_{\phi(j+1)} - \alpha S_m + S_m - S_{\phi(j)}| > 2^{-j} \cap B_{j,m})$$

Par parité de la fonction de la question 23, le terme du membre de droite est égal à deux fois le terme pour $\alpha = 1$. Ce terme pour $\alpha = 1$ vaut

$$\sum_{\phi(j)+1 \leq m \leq \phi(j+1)} \mathbb{P}(|S_{\phi(j+1)} - S_{\phi(j)}| > 2^{-j} \cap B_{j,m}) = \sum_{\phi(j)+1 \leq m \leq \phi(j+1)} \mathbb{P}(A_j \cap B_{j,m}) = \mathbb{P}(A_j)$$

avec la question 22. Ainsi

$$\mathbb{P}(B_j) \leq 2\mathbb{P}(A_j)$$

C. Convergence de la série aléatoire $\sum X_n a_n$

26. Posons $F_J = \bigcup_{j \geq J} B_j$. $(F_J)_{J \geq 0}$ est une suite décroissante d'événements et par théorème de continuité décroissante,

$$\mathbb{P}\left(\bigcup_{j \in \mathbb{N}} F_J\right) = \lim_{J \rightarrow +\infty} \mathbb{P}(F_J)$$

Fixons alors $J \in \mathbb{N}$. Par inégalité de Boole et avec la question 25,

$$0 \leq \mathbb{P}(F_J) \leq \sum_{j=J}^{\infty} \mathbb{P}(B_j) \leq 2 \sum_{j=J}^{\infty} \mathbb{P}(A_j)$$

On utilise alors la question 20 pour obtenir

$$0 \leq \mathbb{P}(F_J) \leq 2 \sum_{j=J}^{\infty} 2^{-j}$$

C'est le reste d'une série convergente et c'est donc un terme de limite nulle quand $J \rightarrow +\infty$. Par théorème d'encadrement,

$$\mathbb{P}\left(\bigcap_{J \in \mathbb{N}} \bigcup_{j \geq J} B_j\right) = 0$$

Cette technique est très classique et vous devez l'avoir en tête quand on vous propose un événement du type intersection de réunion ou réunion d'intersection (passage au complémentaire). On pourra consulter, en particulier, le classique lemme de Borel et Cantelli.

27. En passant au complémentaire dans la question précédente, on obtient

$$\mathbb{P} \left(\bigcup_{J \in \mathbb{N}} \bigcap_{j \geq J} \overline{B_j} \right) = 1$$

$\omega \in B_j$ signifie que le maximum des $|S_n(\omega) - S_{\phi(j)}(\omega)|$ pour $n \in [\phi(j) + 1, \phi(j + 1)]$ est $> 2^{-j}$ c'est à dire que l'une de ces quantités est $> 2^{-j}$.

Ainsi, $\omega \in \overline{B_j}$ signifie que tous les $|S_n(\omega) - S_{\phi(j)}(\omega)|$ pour $n \in [\phi(j) + 1, \phi(j + 1)]$ sont $\leq 2^{-j}$.

On vient de voir qu'il est presque sûr que tous ces événements se réalisent à partir d'un certain rang ce qui correspond au résultat demandé.

28. On a l'inclusion des événements, $\overline{B_j} \subset (|S_{\phi(j+1)} - S_{\phi(j)}| \leq 2^{-j})$.

La question précédente donne a fortiori le caractère presque certain de l'évènement

$$C = \bigcap_{J=0}^{\infty} \bigcap_{j \geq J} (|S_{\phi(j+1)} - S_{\phi(j)}| \leq 2^{-j})$$

Notons alors C' l'évènement de convergence de $(S_{\phi(j)})_{j \in \mathbb{N}}$.

Si $\omega \in C$ alors la série $\sum (S_{\phi(j+1)}(\omega) - S_{\phi(j)}(\omega))_{j \in \mathbb{N}}$ est absolument convergente (et donc convergente) puisqu'il existe J tel qu'à partir du rang J , $|S_{\phi(j+1)}(\omega) - S_{\phi(j)}(\omega)| \leq 2^{-j}$ et puisque $\sum (2^{-j})$ converge. Par dualité suite-séries (i.e. en revenant aux sommes partielles), la suite $(S_{\phi(j)}(\omega))$ converge. Ainsi $C \subset C'$.

Comme $\mathbb{P}(C) = 1$, on a aussi $\mathbb{P}(C') = 1$.

La suite $(S_{\phi(j)})_{j \in \mathbb{N}}$ est presque sûrement convergente

29. Soit ω élément de l'évènement de la question 27 (c'est à dire $\overline{B}$). Il lui est donc associé un J . Soit $n \geq \phi(J) + 1$. La suite $(\phi(j))_{j \in \mathbb{N}}$ étant strictement croissante et composée d'entiers, elle tend vers $+\infty$ et

$$\exists ! j_n \geq J, \phi(j_n) + 1 \leq n \leq \phi(j_n + 1)$$

On peut alors écrire que

$$S_n(\omega) = S_{\phi(j_n)}(\omega) + (S_n(\omega) - S_{\phi(j_n)}(\omega))$$

Par définition de l'évènement $\overline{B}$ qui contient ω ,

$$|S_n(\omega) - S_{\phi(j_n)}(\omega)| \leq 2^{-j_n}$$

Quand $n \rightarrow +\infty$, on a $j_n \rightarrow +\infty$. Ainsi, $(S_n(\omega) - S_{\phi(j_n)}(\omega)) \rightarrow 0$.

Mais on a de même $(S_{\phi(j_n)}(\omega))$ qui converge (avec la question précédente qui donne $C \subset C'$ avec les notations utilisées dans cette question, et toujours car $j_n \rightarrow +\infty$ quand $n \rightarrow +\infty$).

On a donc écrit $S_n(\omega)$ comme la somme de deux termes généraux de suites convergentes et $(S_n(\omega))$ converge.

La convergence d'une série étant par définition la convergence de la suite des sommes partielles et puisque l'évènement $\overline{B}$ de la question 27 est presque sûr,

La série $\sum X_n a_n$ est presque sûrement convergente

Remarque : on est passé rapidement sur le fait que $j_n \rightarrow +\infty$ quand $n \rightarrow +\infty$. Il ne nous semble pas que ce soit un point à développer. Si on veut le faire, on peut noter que (j_n) est croissante (cela est simple) et non majorée car sinon, $n \leq \phi(j_n + 1)$ montrerait que $\mathbb{N}$ est majoré. Ici, cet argument, certes important, n'est pas central à notre avis. La preuve donnée montre la compréhension du processus.

Dérivation C^K pour des séries aléatoires de fonctions

30. Supposons la propriété (H_2) vérifiée. Pour tout ℓ , $f_n(x_\ell)$ est le terme général d'une série absolument convergente et donc de limite nulle. A partir d'un certain rang N , ce terme est en module ≤ 1 et ainsi

$$\forall n \geq N, 0 \leq f_n(x_\ell)^2 \leq |f_n(x_\ell)|$$

Par comparaison des séries positives, $\sum (f_n(x_\ell)^2)$ est convergente. Ainsi, (H'_2) est aussi vérifiée.

La propriété (H_2) entraîne la propriété (H'_2)

31. Pour un ℓ fixé, l'évènement F_ℓ de convergence de $\sum (X_n f_n(x_\ell))$ est presque sûr d'après la question 29 (l'hypothèse (H'_2) nous permettant d'utiliser cette question). On s'intéresse à l'évènement

$$F = \bigcap_{\ell=1}^K F_\ell$$

C'est une intersection finie (dénombrable suffirait) d'évènements presque sûrs et c'est donc un évènement presque sûr.

La probabilité de convergence de toutes les séries $\sum X_n f_n(x_\ell)$ vaut 1

32. Soit $\omega \in \Omega$. Posons $g_n = X_n(\omega)(f_n - P_n)$.

- Les fonctions g_n sont toutes de classe C^K sur $[0, 1]$.
- Pour tout $\ell \in \llbracket 1, K \rrbracket$, $g_n(x_\ell) = 0$ est le terme général d'une série absolument convergente.
- $\|g_n^{(K)}\|_\infty = \|(f_n - P_n)^{(K)}\|_\infty = \|f_n^{(K)}\|_\infty$ ($P_n^{(K)} = 0$) est le terme général d'une série convergente.

La partie II s'applique et indique que $\sum_{n=0}^\infty g_n$ est de classe C^K avec des dérivées d'ordre 0 à K toutes normalement convergentes sur $[0, 1]$ et des dérivées qui s'obtiennent en dérivant terme à terme.

L'évènement proposé est l'univers Ω tout entier

Remarque : on a donc l'évènement certain ce qui est mieux que presque sûr !

33. Il nous suffit de montrer que l'intersection des évènements des questions 31 et 32 est inclus dans celui proposé dans cette question. On se donne donc $\omega \in \Omega$ tel que

$$\forall \ell \in \llbracket 1, K \rrbracket, \sum (X_n(\omega) f_n(x_\ell)) \text{ converge}$$

et tel que ω est dans l'évènement de la question 32 (ce qui, en réalité, est toujours vérifié).

On pose $g_n = X_n(\omega)f_n$ et, en adoptant les mêmes notations que dans la question précédente, on écrit que

$$g_n = X_n(\omega)P_n + X_n(\omega)(f - P_n)$$

- Par choix de ω , $X_n(\omega)(f - P_n)$ est le terme général d'une série qui converge uniformément sur $[0, 1]$ ainsi que toutes ses dérivées jusqu'à l'ordre K . De plus, la somme de la série est de classe C^K et on obtient ses dérivées en dérivant terme à terme.
- On veut maintenant obtenir la même propriété pour la série de fonctions de terme général $Q_n = X_n(\omega)P_n$. Montrons que l'on peut appliquer le théorème de régularité des sommes de séries de fonctions.
 - En utilisant les notations de la question 7, on a $Q_n = \sum_{j=1}^K X_n(\omega)f_n(x_j)L_j$ qui définit une fonction (polynomiale de degré $\leq K-1$) de classe C^K et

$$\forall k \in \llbracket 0, K \rrbracket, Q_n^{(k)} = \sum_{j=1}^K X_n(\omega)f_n(x_j)L_j^{(k)}$$

- Comme les séries $\sum (X_n(\omega)f_n(x_\ell))$ convergent, $\sum (Q_n^{(k)})$ converge simplement sur $[0, 1]$ pour tout $k \in \llbracket 0, K \rrbracket$.
- $Q_n^{(K)} = 0$ est le terme général d'une série qui converge uniformément sur $[0, 1]$.

Le théorème du cours s'applique et donne que $\sum_{n=0}^{\infty} Q_n$ est de classe C^K sur $[0, 1]$, que ses dérivées s'obtiennent en dérivant terme à terme ET que toutes les séries dérivées intermédiaires convergent uniformément sur $[0, 1]$.

On en déduit alors que la même propriété est vraie pour g_n et on a montré que

l'évènement proposé est presque sûr

Remarque : c'est une question qui est difficile en particulier car il utilise le théorème de régularité des séries de fonctions de manière atypique. En effet, dans celui-ci on vérifie souvent que toutes les séries dérivées convergent uniformément sur tout segment alors que c'est en réalité une des conclusions si on énonce le résultat avec des hypothèses faibles comme c'est le cas dans le programme.

34. La question ne me semble pas avoir de sens. On cherche K tel que les hypothèses soient vérifiées et non pas "tel que l'évènement se réalise" (on ne cherche pas un élément de l'univers). On interprète donc la question en ce sens.

Toutes les fonctions f_n sont de classe C^∞ sur $[0, 1]$ ($\sin(x/n)$ prend des valeurs dans $[0, 1]$ quand $n \geq 1$ et $x \in [0, 1]$ et le terme dans le logarithme est > 0).

Pour un $x \in [0, 1]$ donné, $f_n(x) \sim \frac{x}{n}$ quand $n \rightarrow +\infty$ et donc $f_n(x)^2 = O(1/n^2)$ est le terme général d'une série convergente. On pourra donc trouver des x_i vérifiant (H'_2) pour n'importe quel $K \in \mathbb{N}^*$.

Il s'agit donc de choisir K tel que $\sum (f_n^{(K)})$ converge normalement sur $[0, 1]$. Or,

$$f_n''(x) = -\frac{1 + \sin(x/n)}{n^2(1 + \sin(x/n))^2}$$

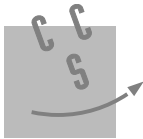
Comme $x \in [0, 1] \subset [0, \pi/2]$, le dénominateur est plus grand que n^2 et donc

$$\|f_n''\|_\infty \leq \frac{2}{n^2}$$

qui est le terme d'une série convergente.

$K = 2$ convient

Remarque : pour $x \in]0, 1]$, $f_n(x) \sim \frac{x}{n}$ quand $n \rightarrow +\infty$ est le terme général d'une série divergente. L'hypothèse (H_2) n'est, elle, pas vérifiée.



Option informatique

MP

2022

CONCOURS CENTRALE-SUPÉLEC

4 heures

Calculatrice autorisée

Ce sujet aborde différents problèmes autour des automates et des expressions rationnelles. On explore dans une première partie des propriétés sur le miroir d'un langage, sur les palindromes et sur les automates correspondants. On implémente ensuite l'algorithme de détermination d'un automate, afin de construire de manière effective l'automate de Brzozowski qui a la propriété d'être minimal. Dans une deuxième partie, on travaille sur la syntaxe des expressions rationnelles, puis sur une construction de l'expression rationnelle associée à un automate donné, par un algorithme diviser pour régner, dû à Conway, en exploitant une représentation matricielle d'un automate et la construction de l'étoile d'une matrice d'expressions rationnelles. Enfin, dans une troisième partie, on introduit les dérivées d'Antimirov, qui permettent d'obtenir un automate fini non déterministe avec peu d'états qui reconnaît le langage spécifié par une expression rationnelle. Les trois parties sont indépendantes, de difficulté progressive.

Langages et mots

On appelle *alphabet* tout ensemble fini de lettres. On note généralement l'alphabet Σ .

On note Σ^* l'ensemble de tous les mots formés sur l'alphabet Σ .

La *longueur* (ou la *taille*) d'un mot $w \in \Sigma^*$ est son nombre de lettres et se note $|w|$. Le *mot vide*, noté ε , est le seul mot de longueur nulle.

Si un mot $w \in \Sigma^*$ est de longueur $|w| = n$, on le note $w = a_0 a_1 \dots a_{n-1}$, où les a_i sont des lettres de Σ .

Un *langage* sur l'alphabet Σ est un ensemble $L \subset \Sigma^*$.

L'*étoile de Kleene* d'un langage L , notée L^* , est le plus petit langage qui inclut L , qui contient ε et qui est stable par concaténation.

La concaténation de deux langages L et L' est notée $L \cdot L'$, souvent abrégé en LL' lorsqu'il n'y a pas d'ambiguïté.

Automates finis

Un *automate fini non déterministe* sur un alphabet Σ est un quadruplet $A = (Q, I, F, T)$, où Q est un ensemble fini d'états, $I \subset Q$ est le sous-ensemble des états initiaux, $F \subset Q$ est le sous-ensemble des états finaux et l'ensemble $T \subset Q \times \Sigma \times Q$ est l'ensemble des *transitions*, étiquetées par les lettres de l'alphabet Σ .

Si $(q, a, q') \in T$, on note $q \xrightarrow{a} q'$ cette transition.

Pour représenter graphiquement un automate, on utilise une flèche entrante pour désigner un état initial et une flèche sortante pour désigner un état final, comme l'illustre l'exemple de la figure 1.

Un mot $w = a_0 \dots a_{n-1}$ est reconnu par l'automate A s'il existe une succession de transitions :

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} \dots \xrightarrow{a_{n-1}} q_n \quad \text{avec} \quad q_0 \in I \quad \text{et} \quad q_n \in F.$$

On dira que le mot w étiquette un chemin dans l'automate A allant de q_0 à q_n .

Le langage d'un automate A , noté L_A , est exactement l'ensemble des mots reconnus par l'automate A . On dit alors que A reconnaît L_A . Un langage est dit *reconnaisable* s'il est le langage d'un automate fini.

Un *automate fini déterministe* sur un alphabet Σ est un quadruplet $A = (Q, \{q_0\}, F, \delta)$, où l'ensemble des états initiaux est un singleton (un unique état initial) et où l'ensemble des transitions T est remplacé par une fonction de transition δ définie sur un sous-ensemble de $Q \times \Sigma$ et à valeurs dans Q . Pour chaque couple $(q, a) \in Q \times \Sigma$, il existe au plus une transition (q, a, q') qui, si elle existe, est telle que $q' = \delta(q, a)$.

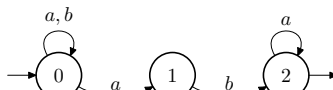
L'automate est déterministe *complet* si la fonction de transition δ est définie sur $Q \times \Sigma$. Dans ce cas, on définit la *fonction de transition étendue* δ^* sur $Q \times \Sigma^*$ par

$$\forall q \in Q, \quad \begin{cases} \delta^*(q, \varepsilon) = q \\ \delta^*(q, wa) = \delta(\delta^*(q, w), a) \end{cases} \quad \forall w \in \Sigma^*, \forall a \in \Sigma$$

Les automates seront représentés par le type Caml suivant

```
type automate = { nb : int; (* nombre d'états *)
                  init : int list; (* états initiaux *)
                  final : int list; (* états finaux *)
                  trans : (int * char * int) list };; (* transitions *)
```

l'ensemble d'états Q d'un automate implémenté étant toujours supposé être un intervalle d'entiers $\llbracket 0, n-1 \rrbracket$.

Figure 1 L'automate $\mathcal{A}_1$

Par exemple, l'automate $\mathcal{A}_1$ de la figure 1 est codé par

```
let a1 = { nb = 3 ;
          init = [0];
          final = [2];
          trans = [(0, 'a', 0); (0, 'a', 1); (0, 'b', 0); (1, 'b', 2); (2, 'a', 2)] } ;;
```

On accède au nombre d'états par `a1.nb`, à la liste des états initiaux par `a1.init`, à la liste des états finaux par `a1.final` et à la liste des transitions par `a1.trans`.

Expressions rationnelles

Soit Σ un alphabet. On définit la *syntaxe des expressions rationnelles* par :

- $\emptyset$, ε et a sont des expressions rationnelles, pour toute lettre $a \in \Sigma$;
- si E et F sont deux expressions rationnelles, alors $(E + F)$, $(E \cdot F)$ et E^* sont des expressions rationnelles.

La *sémantique des expressions rationnelles* est définie par l'application $\mathcal{L}$ qui associe à toute expression rationnelle un langage rationnel sur Σ par :

$$\begin{cases} \mathcal{L}(\emptyset) = \emptyset & (\text{langage vide}) \\ \mathcal{L}(\varepsilon) = \{\varepsilon\} & (\text{langage contenant le mot vide}) \\ \mathcal{L}(a) = \{a\} & \forall a \in \Sigma \end{cases}$$

et, si E et F sont deux expressions rationnelles,

$$\begin{cases} \mathcal{L}(E + F) = \mathcal{L}(E) \cup \mathcal{L}(F) \\ \mathcal{L}(E \cdot F) = \mathcal{L}(E) \cdot \mathcal{L}(F) \\ \mathcal{L}(E^*) = \mathcal{L}(E)^* \end{cases}$$

où $\star$ représente l'étoile de Kleene d'un langage et $\cdot$ représente la concaténation de deux langages.

Programmation

Le seul langage de programmation autorisé dans cette épreuve est Caml. Toutes les fonctions des modules `Array` et `List`, ainsi que les fonctions de la bibliothèque standard (celles qui s'écrivent sans nom de module, comme `max` ou `incr` ainsi que les opérateurs comme `/` ou `mod`) peuvent être librement utilisés.

Généralement, les objets mathématiques dans le texte seront notés A , m , i , n , ℓ , alors qu'ils seront représentés en Caml par `a`, `m`, `i`, `n`, `l`.

Les complexités demandées sont des complexités temporelles dans le pire des cas et seront exprimées sous la forme $O(f(n, m))$, où f est une fonction usuelle simple et où n et m sont des paramètres correspondant aux tailles des objets en entrée de l'algorithme.

I Mots et automates

I.A – Miroir d'un mot et automate transposé

Pour tout mot $w = a_0 a_1 \dots a_{n-1}$ de longueur $n \in \mathbb{N}^*$, on définit son *mot miroir* $\tilde{w}$ par $\tilde{w} = a_{n-1} \dots a_1 a_0$. Par convention, le mot vide ε est son propre miroir.

Pour tout langage $L \subset \Sigma^*$, on définit son *langage miroir* $\tilde{L}$ constitué de l'ensemble des mots miroirs du langage L :

$$\tilde{L} = \{\tilde{w} \mid w \in L\}.$$

Q 1. Décrire le langage L_1 de l'automate $\mathcal{A}_1$ de l'exemple de la figure 1 et décrire son langage miroir $\tilde{L}_1$.

Q 2. Dessiner un automate $\tilde{\mathcal{A}}_1$, reconnaissant le langage miroir $\tilde{L}_1$.

Soit $A = (Q, I, F, T)$ un automate non déterministe et $L = L_A$ le langage qu'il reconnaît.

Q 3. Donner, en justifiant, la construction de l'automate miroir $\tilde{A} = (Q, I', F', T')$ qui reconnaît le langage $\tilde{L}$.

Q 4. Écrire une fonction `transpose` de signature `automate -> automate` qui étant donné un automate A non déterministe en entrée, renvoie un automate non déterministe qui reconnaît le miroir de L_A .

Q 5. Quelle est la complexité de cette fonction ?

I.B – Palindromes et rationalité

Soit $w \in \Sigma^*$. On dit que le mot w est un *palindrome* si $\tilde{w} = w$.

Q 6. Écrire une fonction `palindrome` de signature `string -> bool` qui teste, en temps linéaire, si un mot est un palindrome.

On rappelle que pour tout $0 \leq i < (\text{String.length } s)$, le i -ième caractère de la chaîne de caractères s est obtenu par l'expression `s.[i]`.

Pour un alphabet Σ , on note $\text{Pal}(\Sigma)$ l'ensemble des palindromes de Σ^* .

Q 7. Montrer que si Σ est un alphabet à une lettre, alors $\text{Pal}(\Sigma)$ est rationnel.

Q 8. Montrer que si Σ contient au moins deux lettres, alors $\text{Pal}(\Sigma)$ n'est pas rationnel.

On pourra utiliser un automate et un mot de $\text{Pal}(\Sigma) \cap a^*ba^*$.

Soit $L \subset \Sigma^*$ un langage reconnu par l'automate $A = (Q, I, F, T)$.

Pour $(q, q') \in Q^2$, on note $L_{q,q'}$ le langage de tous les mots w qui étiquettent un chemin dans A partant de q et arrivant en q' .

Q 9. Montrer que $L_{q,q'}$ est reconnaissable et exprimer le langage L_A en fonction de langages $L_{q,q'}$.

Q 10. Montrer que $\text{Pal}(\Sigma) \cap (\Sigma^2)^* = \{u\tilde{u} \mid u \in \Sigma^*\}$.

Soit L un langage rationnel reconnu par un automate $A = (Q, I, F, T)$.

On définit les langages $D(L) = \{w\tilde{w} \mid w \in L\}$ et $R(L) = \{w \in \Sigma^* \mid w\tilde{w} \in L\}$.

Q 11. Décrire simplement les langages $D(a^*b)$ et $R(a^*b^*a^*)$.

Q 12. Les langages $D(L)$ et $R(L)$ sont-ils reconnaissables ?

On pourra faire intervenir les langages $L_{q,q'}$, définis ci-dessus.

I.C – Détermination

On rappelle que pour tout automate $A = (Q, I, F, T)$ non déterministe, on peut définir l'automate déterminisé accessible $A_{\text{det}} = (Y, \{I\}, F', \delta)$ où $Y \subset \mathcal{P}(Q)$ est l'ensemble des états accessibles depuis l'état initial $\{I\}$ dans l'automate des parties. Cet automate déterminisé accessible reconnaît le même langage que l'automate A .

Q 13. Écrire un automate $\mathcal{A}_2$ non déterministe à 4 états qui reconnaît le langage $L_2 = (b + ab)^*ba$. Cet automate devra avoir un unique état initial et un unique état final.

Q 14. Appliquer l'algorithme de détermination sur l'automate miroir $\widetilde{\mathcal{A}}_2$ afin d'obtenir l'automate $\mathcal{A}_3 = (\widetilde{\mathcal{A}}_2)_{\text{det}}$. Les états de $\mathcal{A}_3$ seront renommés $e_0, e_1, \dots$.

Q 15. Appliquer l'algorithme de détermination sur l'automate miroir $\widetilde{\mathcal{A}}_3$ afin d'obtenir l'automate $\mathcal{A}_4 = (\widetilde{\mathcal{A}}_3)_{\text{det}}$. Ses états seront renommés $q_0, q_1, \dots$.

Q 16. Quel doit être le langage reconnu par l'automate $\mathcal{A}_4$?

On cherche à généraliser cette construction de façon effective. Pour cela, on va implémenter l'algorithme de détermination.

Il faut d'abord choisir une représentation pour les parties de Q (c'est-à-dire des ensembles d'états). Une solution naïve consisterait à utiliser des listes d'états. Lors du déroulement de l'algorithme de détermination, on peut être amené à effectuer des réunions d'ensembles. Une concaténation simple des listes génère des doublons qu'il faut ensuite supprimer afin que les listes codent bien des ensembles d'états.

Q 17. Écrire une fonction `supprimer` de signature `'a list -> 'a list` qui prend une liste en entrée et supprime toutes les occurrences multiples de ses éléments.

Q 18. Donner la complexité de votre algorithme en fonction de la taille de la liste d'entrée.

On choisit plutôt de coder les ensembles d'états par des entiers.

Pour un automate $A = (Q, I, F, T)$ tel que $Q = \llbracket 0, n-1 \rrbracket$, toute partie de Q va être représentée par un entier entre 0 et $2^n - 1$. Dans la suite, on supposera $n \leq 20$. Soit X une partie de $\llbracket 0, n-1 \rrbracket$. On définit le numéro de X par la fonction suivante

$$\text{numero}(X) = \sum_{i \in X} 2^i.$$

On se donne `pow` un tableau des puissances de 2, qui contient toutes les puissances 2^k , pour $0 \leq k \leq 20$.

```
let pow = Array.make 21 1 ;;
for i = 1 to 20 do
  pow.(i) <- pow.(i-1) * 2
done ;;
```

Soit $q \in \llbracket 0, n-1 \rrbracket$ un état et $k \in \llbracket 0, 2^n - 1 \rrbracket$ le numéro d'un ensemble d'états X , c'est-à-dire $\text{numero}(X) = k$.

Q 19. Écrire une fonction `est_dans` de signature `int -> int -> bool` qui teste, à l'aide d'opérations arithmétiques, si l'état q est dans l'ensemble d'états représenté par le numéro k en $O(1)$ opérations.

Soit ℓ une liste d'états contenant éventuellement plusieurs fois le même état, représentant l'ensemble X .

Q 20. Écrire une fonction `numero` de signature `int list -> int` qui calcule le numéro de l'ensemble X .

Par exemple $\ell = [1; 5; 2; 5; 2; 5; 2; 1; 2; 1]$ représente l'ensemble $X = \{1, 2, 5\}$, de numéro $38 = 2^1 + 2^2 + 2^5$.

Soit ℓ une liste d'états et X un ensemble d'états représenté par son numéro k .

Q 21. Écrire une fonction `intersecte` de signature `int list -> int -> bool` qui vérifie si un élément de ℓ est contenu dans l'ensemble X représenté par k .

On prépare désormais la fonction de transition de l'automate déterminisé accessible.

Soit X un ensemble d'états de Q . On suppose désormais que l'automate est sur l'alphabet à deux lettres $\Sigma = \{a, b\}$.

On cherche à calculer la fonction de transition $\delta : \mathcal{P}(Q) \times \Sigma \rightarrow \mathcal{P}(Q)$ de l'automate déterminisé. On rappelle que, pour $c \in \{a, b\}$ et $X \in \mathcal{P}(Q)$,

$$\delta(X, c) = \bigcup_{q \in X} \{q' \in Q \mid (q, c, q') \in T\}.$$

La transition $(X, c, \delta(X, c))$ sera alors dans l'automate déterminisé.

En parcourant l'ensemble des transitions T de l'automate, on va simultanément calculer les états $(\delta(X, a), \delta(X, b))$, ce qui correspond à la table de transition depuis l'état X .

Q 22. Écrire une fonction `etat_suivant` de signature `int -> (int*char*int) list -> (int*int)` qui, étant donné en entrée un entier k tel que $k = \text{numero}(X)$ et la liste des transitions T , calcule le couple d'entiers (k_a, k_b) tels que $k_a = \text{numero}(\delta(X, a))$ et $k_b = \text{numero}(\delta(X, b))$.

Au moment de construire l'automate déterminisé accessible A_{det} , on va être amené à renommer (c'est-à-dire ici renuméroter) les états de A_{det} pour avoir au final un ensemble d'états Y de la forme $\llbracket 0, N - 1 \rrbracket$ où N sera le nombre de parties de Q accessibles dans l'automate des parties. Pour cela, on va simplement utiliser une liste contenant des couples (k, v) où k est le numéro d'un ensemble d'états X et v le numéro final par lequel k sera remplacé. Par exemple, si à un moment donné de l'algorithme, la liste contient $(6, 2)$, "l'ensemble d'états 6" (qui correspond dans $\mathcal{P}(Q)$ à $\{1, 2\}$) est renuméroté 2.

Q 23. Écrire une fonction `cherche` de signature `int -> (int*int) list -> int` qui renvoie le nouveau numéro d'un ensemble d'états représenté par son numéro k dans une liste comme ci-dessus (-1 si k n'est pas présent).

Q 24. Écrire une fonction `determine` de signature `automate -> automate` qui calcule le déterminisé accessible de l'automate d'entrée. On expliquera brièvement la démarche utilisée.

Q 25. Quelle est la complexité de votre fonction `determine` en fonction du nombre d'états n de A et du nombre d'états N de A_{det} ?

I.D – Algorithme de Brzozowski

L'algorithme de Brzozowski permet d'obtenir un automate déterministe ayant un nombre minimal d'états, reconnaissant le même langage que l'automate initial.

On se donne un automate $A = (Q, I, \{f\}, T)$ qui reconnaît le langage L et tel que l'automate miroir $\tilde{A}$ est déterministe et accessible.

On note $A_{\text{det}} = (Y, \{I\}, F, \delta)$ le déterminisé accessible de A .

Si u est un mot et L un langage, on note $u^{-1}L = \{w \in \Sigma^* \mid uw \in L\}$.

Q 26. Soit $q \in Q$ un état et $u \in \Sigma^*$ un mot. Montrer que si $q \in \delta^*(\{I\}, u)$, alors il existe un mot $w \in \Sigma^*$ tel que $uw \in L$.

Q 27. Montrer la propriété $(*)$: si l'on prend deux mots u et v dans Σ^* tels que $u^{-1}L = v^{-1}L$, alors dans l'automate A_{det} déterminisé, $\delta^*(\{I\}, u) = \delta^*(\{I\}, v)$.

Q 28. En déduire que si A est un automate quelconque reconnaissant L , alors en posant $B = (\tilde{A})_{\text{det}}$, montrer que $(\tilde{B})_{\text{det}}$ reconnaît L et vérifie la propriété $(*)$.

Q 29. Écrire une fonction `minimal` de signature `automate -> automate` appliquant la construction de Brzozowski sur l'automate d'entrée. On fera abstraction de la taille des automates générés, possiblement problématique.

II Expression rationnelle associée à un automate

Dans cette partie, on introduit un algorithme, dû à Conway, pour le calcul de l'expression rationnelle associée au langage d'un automate, via l'utilisation de matrices dont les coefficients sont des expressions rationnelles.

II.A – Simplification d'expressions rationnelles équivalentes

On se donne en Caml le type `exprat` des expressions rationnelles

```
type exprat = Vide
            | Epsilon
            | Lettre of char
            | Union of exprat * exprat ;;
            | Concat of exprat * exprat
            | Etoile of exprat ;;
```

II.A.1)

Q 30. Écrire une fonction `lettre` de signature `exprat -> int` qui renvoie le nombre de lettres présentes dans l'expression rationnelle en argument. Par exemple, si $E = (a^*b) + abba(a + \varepsilon)^* + \emptyset$, (`lettre e`) doit renvoyer 7.

Q 31. Écrire une fonction `est_vide` de signature `exprat -> bool` qui teste si le langage rationnel représenté par l'expression rationnelle en argument est vide.

II.A.2) Dans cette section, on travaille formellement sur la syntaxe des expressions rationnelles.

On utilise les équivalences évidentes suivantes

$$\emptyset + E \equiv E + \emptyset \equiv E \quad E \cdot \varepsilon \equiv \varepsilon \cdot E \equiv E \quad E \cdot \emptyset \equiv \emptyset \cdot E \equiv \emptyset \quad \emptyset^* \equiv \varepsilon \quad \varepsilon^* \equiv \varepsilon \quad (E^*)^* \equiv E^*$$

où la notation $E \equiv E'$ signifie que les langages représentés sont égaux : $\mathcal{L}(E) = \mathcal{L}(E')$.

La fonction suivante réalise une simplification à la racine sur une expression du type `Union` en suivant la règle donnée.

```
let su expr = match expr with
| Union( Vide , e ) -> e
| Union( e , Vide ) -> e
| _ -> expr;;
```

De même, on peut écrire une fonction `sc` : `exprat->exprat` qui simplifie à la racine une expression de type `Concat`. On suppose codées ces fonctions.

Q 32. Écrire une fonction `se` : `exprat -> exprat` qui simplifie à la racine une expression de type `Etoile` avec les règles données.

Prenons par exemple $E_n = (a + (b.(b.(b.(b...\emptyset)...)...$, où n lettres b concaténées se succèdent.

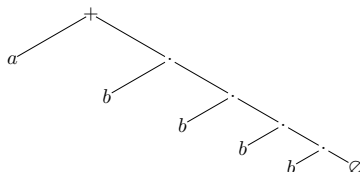


Figure 2 Exemple de l'arbre syntaxique de l'expression E_4

Q 33. Combien d'applications de règles décrites ci-dessus sont-elles nécessaires pour obtenir à partir de E_n l'expression équivalente a ?

Q 34. Écrire une fonction `simplifie` : `exprat -> exprat` qui simplifie une expression rationnelle selon les règles données.

II.B – Matrices d'expressions rationnelles

Dans la suite, on considère des matrices d'expressions rationnelles

```
type mat = exprat array array ;;
```

La *matrice nulle* de taille n est la matrice de taille n où chaque coefficient vaut $\emptyset$.

La *matrice identité* de taille n est la matrice de taille n où chaque coefficient vaut ε sur la diagonale et $\emptyset$ en dehors de la diagonale.

On définit la *somme de deux matrices* A et B de taille $(n \times m)$ par la matrice $A + B$ de taille $(n \times m)$ où $[A + B]_{i,j} = A_{i,j} + B_{i,j}$ où le $+$ représente l'opération rationnelle d'union et $0 \leq i \leq n - 1$, $0 \leq j \leq m - 1$.

On définit le *produit de deux matrices* A et B de taille $(n \times p)$ et $(p \times q)$ à la manière du produit matriciel usuel AB , de taille $(n \times q)$, où la somme de coefficients est remplacée par l'union et où le produit de coefficients est remplacé par la concaténation des expressions rationnelles.

II.B.1)

Q 35. Écrire une fonction `somme` de signature `mat -> mat -> mat` effectuant la somme de deux matrices d'expressions rationnelles de même taille $(n \times p)$. Quelle est sa complexité ?

Q 36. Écrire une fonction `produit` de signature `mat -> mat -> mat` effectuant le produit de deux matrices d'expressions rationnelles, en supposant que les tailles sont bien compatibles (la première de taille $(n \times p)$ et la seconde de taille $(p \times q)$.) On ne vérifiera pas la compatibilité des tailles. Quelle est sa complexité ?

On cherche désormais à définir l'étoile de Kleene d'une matrice carrée d'expressions rationnelles.

II.B.2) Étude de l'étoile d'une matrice de taille 2

Plaçons-nous dans le cas d'une matrice carrée de taille 2, $M = \begin{pmatrix} a & b \\ c & d \end{pmatrix}$, où a, b, c et d sont quatre lettres.

On associe à cette matrice M le graphe étiqueté à deux sommets de la figure 3.

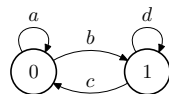


Figure 3

On note $L_{i,j}$ le langage de l'automate $\mathcal{A}_{i,j} = (\{0, 1\}, \{i\}, \{j\}, T)$, où $T = \{(i, M_{i,j}, j) \mid (i, j) \in \{0, 1\}^2\}$.

Q 37. Donner une expression rationnelle sur l'alphabet $\{a, b, c, d\}$ pour décrire chaque langage $L_{i,j}$.

II.B.3) Étoile d'une matrice carrée de taille quelconque

Pour la définition de l'étoile d'une matrice carrée d'expressions rationnelles, on procède récursivement, sur la taille n de la matrice carrée M :

- si la taille vaut 1, $M = (e)$, donc $M^* = (e^*)$;
- sinon, pour M de taille $n \geq 2$, on découpe M par blocs

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

où A et D sont carrées de taille ≥ 1 , et on définit

$$M^* = \begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix}$$

où

$$A' = (A + BD^*C)^* \quad B' = A^*B(D + CA^*B)^* \quad C' = D^*C(A + BD^*C)^* \quad D' = (D + CA^*B)^*$$

On suppose déjà codées les deux fonctions suivantes sur les matrices d'expressions rationnelles :

- (`decouper m n1 n2`) renvoie quatre matrices blocs A, B, C et D telles que A est carrée de taille n_1 , D est carrée de taille n_2 et $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$, où la matrice d'entrée M est carrée de taille $n = n_1 + n_2$.

`decouper : mat -> int -> int -> (mat*mat*mat*mat)`

- (`recoller a b c d`) renvoie la matrice $M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$ à partir des quatre blocs A, B, C et D codés par `a, b, c, d` dont les tailles sont compatibles.

`recoller : mat -> mat -> mat -> mat -> mat`

On propose la décomposition récursive suivante pour une matrice M carrée de taille $n \geq 2$,

$$M = \begin{pmatrix} a & B \\ C & D \end{pmatrix}$$

où a est une lettre et D est carrée de taille $n-1$. Ainsi,

$$M^* = \begin{pmatrix} A' & B' \\ C' & D' \end{pmatrix}$$

où

$$A' = (a + BD^*C)^* \quad B' = a^*B(D + Ca^*B)^* \quad C' = D^*C(a + BD^*C)^* \quad D' = (D + Ca^*B)^*$$

Q 38. Évaluer les différentes complexités des sommes et produits effectués, et en déduire que si $C(n)$ est la complexité du calcul de l'étoile pour une matrice de taille n , alors $C(n) = 2C(n-1) + O(n^2)$. En déduire la complexité de cet algorithme.

On se place dans le cas où la taille n de la matrice M est une puissance de 2, avec $n \geq 2$.

On propose désormais la décomposition récursive suivante

$$M = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

où les matrices A et D sont carrées de taille $n/2$ chacune.

Q 39. Évaluer les différentes complexités des sommes et produits effectués, et en déduire que si $C(n)$ est la complexité du calcul de l'étoile pour une matrice de taille n , alors $C(n) = 4C(n/2) + O(n^3)$. En déduire la complexité de cet algorithme.

Q 40. Comment gérer le cas des matrices M de taille n quelconque ? Quelle complexité peut-on obtenir pour le calcul de M^* ?

Q 41. Écrire la fonction `etoile` de signature `mat -> mat` qui renvoie l'étoile d'une matrice en utilisant l'algorithme récursif le plus adéquat.

II.C – Algorithme de Conway

Soit $A = (Q, I, F, T)$ un automate, où l'ensemble d'états est $Q = \llbracket 0, n-1 \rrbracket$.

On définit M_A la *matrice de transition* de l'automate A par la matrice d'expressions rationnelles de taille $(n \times n)$ telle que pour $0 \leq i, j \leq n-1$, $[M_A]_{i,j} = \sum_{c \in \Sigma(i,c,j) \in T} c$ s'il existe au moins une telle lettre c , $\emptyset$ sinon.

On admet la propriété suivante : pour tout état $(i, j) \in Q^2$, $\mathcal{L}([M_A^*]_{i,j}) = L_{i,j}$, où $L_{i,j}$ est le langage défini en question 9.

Q 42. Montrer que

$$L_A = \mathcal{L}([X M_A^* Y]_{0,0})$$

où X est une matrice ligne d'expressions rationnelles de la forme $X = (x_0 \quad \dots \quad x_{n-1})$ où chaque $x_i \in \{\emptyset, \varepsilon\}$, et Y est une matrice colonne d'expressions rationnelles de la forme $Y = \begin{pmatrix} y_0 \\ \vdots \\ y_{n-1} \end{pmatrix}$ où chaque $y_j \in \{\emptyset, \varepsilon\}$. On

précisera les valeurs de X et Y en fonction de l'automate A .

Q 43. Écrire la fonction `langage` de signature `automate -> exprat` prenant en entrée un automate et renvoyant une expression rationnelle représentant le langage de cet automate. Quelle est la complexité de cette fonction ?

III Automate des dérivées d'Antimirov

On propose pour conclure une méthode permettant de calculer un automate non déterministe ayant peu d'états à partir d'une expression rationnelle.

Si S et S' sont deux ensembles d'expressions rationnelles, on convient que

$$S \cdot S' = \{E \cdot E' \mid (E, E') \in S \times S'\}$$

En particulier, on a $\emptyset \cdot S = \emptyset$ et $\{\varepsilon\} \cdot S = S$.

Soit E une expression rationnelle sur un alphabet Σ et soit $a \in \Sigma$ une lettre. On définit la dérivée partielle de E par a , notée $\partial_a(E)$, comme un *ensemble d'expressions rationnelles* défini inductivement par

$$\begin{aligned} \partial_a(\emptyset) &= \emptyset & (\text{où } \emptyset \text{ est l'ensemble vide}) \\ \partial_a(\varepsilon) &= \emptyset \\ \partial_a(b) &= \begin{cases} \{\varepsilon\} & \text{si } a = b \\ \emptyset & \text{sinon} \end{cases} & \text{pour toute lettre } b \in \Sigma \\ \partial_a(E + F) &= \partial_a(E) \cup \partial_a(F) \\ \partial_a(E^*) &= \partial_a(E) \cdot \{E^*\} \\ \partial_a(EF) &= \begin{cases} \partial_a(E) \cdot \{F\} & \text{si } \varepsilon \notin \mathcal{L}(E) \\ \partial_a(E) \cdot \{F\} \cup \partial_a(F) & \text{sinon} \end{cases} \end{aligned}$$

Notons bien que dans une dérivée partielle, on a une expression rationnelle, et que son résultat est un ensemble d'expressions rationnelles.

Par exemple, pour $E = a^*(a + b) = (a^*) \cdot (a + b)$, on calcule $\partial_a(E)$ et $\partial_b(E)$ ainsi : comme $\varepsilon \in \mathcal{L}(a^*)$,

$$\begin{aligned} \partial_a(E) &= (\partial_a(a^*)) \cdot \{a + b\} \cup \partial_a(a + b) \\ &= (\partial_a a) \cdot \{a^*\} \cdot \{a + b\} \cup \partial_a(a) \cup \partial_a(b) \\ &= \{\varepsilon\} \cdot \{a^*(a + b)\} \cup \{\varepsilon\} \cup \emptyset \\ &= \{a^*(a + b); \varepsilon\} \\ \partial_b(E) &= (\partial_b(a^*)) \cdot \{a + b\} \cup \partial_b(a + b) \\ &= (\partial_b a) \cdot \{a^*(a + b)\} \cup \emptyset \cup \{\varepsilon\} \\ &= \emptyset \cup \{\varepsilon\} \\ &= \{\varepsilon\} \end{aligned}$$

Q 44. Pour $E = (ab + b)^*ba$, calculer $\partial_a(E)$ et $\partial_b(E)$.

Cette définition de dérivée partielle est étendue à tout mot $w \in \Sigma^*$ et à des ensembles d'expressions rationnelles par : pour $a \in \Sigma$, $w \in \Sigma^*$ et S un ensemble d'expressions rationnelles,

$$\partial_\varepsilon(E) = \{E\} \quad \partial_{wa}(E) = \partial_a(\partial_w(E)) \quad \partial_w(S) = \bigcup_{E \in S} \partial_w(E)$$

On construit alors l'automate d'Antimirov à partir des dérivées partielles d'une expression rationnelle.

Partons de E une expression rationnelle. L'automate d'Antimirov de l'expression E est $A = (Q, I, F, T)$ défini par

$$\begin{cases} Q = \{E_1 \mid \exists w \in \Sigma^*, E_1 \in \partial_w(E)\} \\ I = \{E\} \\ F = \{E_1 \in Q \mid \varepsilon \in \mathcal{L}(E_1)\} \\ T = \{(E_1, c, E_2) \in Q \times \Sigma \times Q \mid E_2 \in \partial_c(E_1)\} \end{cases}$$

On rappelle la notation $w^{-1}L$ de la partie I : pour tout mot $w \in \Sigma^*$ et tout langage $L \subset \Sigma^*$,

$$w^{-1}L = \{u \in \Sigma^* \mid wu \in L\}$$

Q 45. Dessiner l'automate obtenu à partir de l'expression rationnelle $E = (ab + b)^*ba$. On indiquera précisément l'ensemble d'états Q .

Q 46. Montrer que pour tous mots u, v et tout langage L , $v^{-1}u^{-1}L = (uv)^{-1}L$.

Pour S ensemble d'expressions rationnelles, on note $\mathcal{L}(S)$ la réunion des langages des expressions de S . On admet que, si E est une expression rationnelle et x une lettre,

$$\mathcal{L}(\partial_x(E)) = x^{-1}\mathcal{L}(E)$$

Q 47. Soit S un ensemble d'expressions rationnelles sur Σ et w un mot de Σ^* . Montrer que

$$\mathcal{L}(\partial_w(S)) = w^{-1}\mathcal{L}(S).$$

Q 48. Montrer que pour tout mot $w \in \Sigma^*$, l'ensemble $\partial_w(E)$ est l'ensemble des états accessibles depuis l'état E en lisant le mot w .

Q 49. En déduire que l'automate d'Antimirov reconnaît bien le langage de l'expression rationnelle E .

Pour tout mot $w \in \Sigma^*$ et $w \neq \varepsilon$, et pour toutes expressions rationnelles E et F sur Σ , on vérifie que

$$\partial_w(E + F) = \partial_w(E) \cup \partial_w(F) \quad (\text{III.1})$$

$$\partial_w(EF) \subset \partial_w(E) \cdot F \cup \bigcup_{v \in S^+(w)} \partial_v(F) \quad (\text{III.2})$$

$$\partial_w(E^*) \subset \bigcup_{v \in S^+(w)} \partial_v(E) \cdot E^* \quad (\text{III.3})$$

où $S^+(w)$ est l'ensemble des suffixes non vides d'un mot w .

Pour une expression rationnelle E , on note

$$Q(E) = \bigcup_{w \in \Sigma^*, w \neq \varepsilon} \partial_w(E).$$

Q 50. Montrer que pour toute expression rationnelle E , le cardinal de $Q(E)$ est majoré par le nombre de lettres présentes dans l'écriture syntaxique de E (qu'on notera $\|E\|$). Qu'en déduit-on sur l'automate d'Antimirov ?

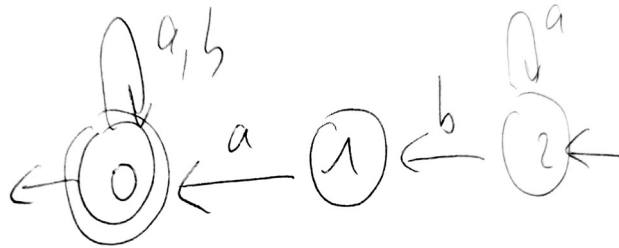
• • • FIN • • •

SOLUTION

Q1. $L_1 = (a+b)^* aba^*$ et $\widetilde{L}_1 = a^* ba (a+b)^*$

Commentaires : Question élémentaire qui ne doit pas vous poser de difficulté. Si c'est le cas, retravailler le chapitre langage.

Q2. Il suffit d'inverser toutes les flèches de l'automate et d'échanger les états finaux et initiaux.



Commentaires : Même remarque que précédemment.

Q3. Il suffit d'inverser toutes les flèches de l'automate et d'échanger les états finaux et initiaux. On pose $Q' = Q$, $I' = F$, $F' = I$ et

$$T' = \{(q', x, q) \text{ avec } (q, x, q') \in T\} \text{ alors } A' = (Q', I', F', T')$$

reconnait $\widetilde{L}$. Un mot $w = a_0 \cdots a_{n-1}$ est reconnu par A si et seulement s'il existe des états $q_0, \dots, q_n$ de Q avec $q_0 \in I$, $q_n \in F$ et

$$q_0 \xrightarrow{a_0} q_1 \xrightarrow{a_1} q_2 \rightarrow \cdots \rightarrow q_{n-1} \xrightarrow{a_{n-1}} q_n$$

est un chemin dans l'automate c'est-à-dire pour chaque i , $q_i \xrightarrow{a_i} q_{i+1}$ est une transition dans A i.e $(q_i, a_i, q_{i+1}) \in T$. Ceci est équivalent à dire que

$$q_0 \xleftarrow{a_0} q_1 \xleftarrow{a_1} q_2 \leftarrow \cdots \leftarrow q_{n-1} \xleftarrow{a_{n-1}} q_n$$

est un chemin dans l'automate A' (car $q \xrightarrow{x} q'$ est une transition dans A si et seulement si $q' \xrightarrow{x} q$ est une transition dans A') c'est-à-dire que

$$q_n \xrightarrow{a_n} q_{n-1} \xrightarrow{a_{n-1}} q_{n-2} \rightarrow \cdots \rightarrow q_1 \xrightarrow{a_0} q_0$$

est un chemin dans l'automate A' avec $q_0 \in F = I'$ et $q_n \in I = F'$ si et seulement si le mot $a_n a_{n-1} \cdots a_0 = \widetilde{w}$ est reconnu par A' .

Ainsi, le langage reconnu par A' est le miroir du langage reconnu par A et comme ce dernier est L , on en déduit que A' reconnaît $\widetilde{L}$.

Commentaires : Il s'agit d'une construction classique. Pour les preuves, entraînez-vous à en faire fondé sur les définitions et non des espoirs. Celles-ci et celles de votre cours sur l'automate produit ou de glushkov par exemple sont accessibles et typiques des preuves sur les automates. Les correcteurs ne valorisent aucune explication non fondées rigoureusement sur les définitions de l'automate ou du langage reconnu par celui-ci.

Q4.

```
let transpose a =
  let rec aux l = match l with
    | [] -> []
    | (x,q,x')::r -> (x',q,x)::(aux r)
  in let t = aux a.trans in
  {nb = a.nb ; init = a.final ; final = a.init ; trans = t};;
```

Commentaires : Question de niveau standard et proche du cours. Si vous avez eu des difficultés, retravailler la programmation des automates (calcul de δ , de δ^* , reconnaître un mot, etc).

Q5. Elle est de complexité $O(|T|)$ (le coût pour inverser toutes les transitions, les autres opérations étant en nombre borné et de coût constant).

Commentaires : Question élémentaire ... si vous avez répondu à la question précédente.

Q6.

```
let palindrome m =
  let n = String.length m in
  let continue = ref true and i = ref 0 in
  while !i <= n - 1 - (!i) && !continue do
    continue := m.[!i] == m.[n-1-(!i)] ;
    incr i ;
  done;
  !continue;;
```

Commentaires : Il s'agit d'une question de cours.

Q7. Si $\Sigma = \{a\}$ pour une certaine lettre a alors tout mot sur Σ est de la forme $a \cdots a$ qui est un palindrome donc $\text{Pal}(\Sigma) = \Sigma^*$.

Commentaires : Question élémentaire.

Q8. Soient a et b deux lettres distinctes appartenant à Σ . Supposons que $\text{Pal}(\Sigma)$ est rationnel. Comme a^*ba^* l'est (il est dénoté par une expression rationnelle), on peut affirmer que $L = \text{Pal}(\Sigma) \cap a^*ba^*$ est un langage rationnel. Or, il est immédiat que

$$L = \{a^nba^n, n \in \mathbb{N}\}.$$

D'après le lemme de l'étoile (ou de pompage), il existe un entier $N \geq 1$ tel que pour tout m de L peut s'écrire $m = uvw$ avec $|uv| \leq N$, $v \neq \varepsilon$ et $uv^k w$ appartient à L pour tout entier k . Considérons alors le mot $m = a^N b a^N$ alors uv s'écrit a^n avec $n \leq N$ et $w = a^{N-n} b a^N$ donc u s'écrit a^r , v s'écrit a^s avec $s \geq 1$ et

$$uv^k w = a^{r+ks+N-n} b a^N$$

doit appartenir à L . En particulier, c'est un palindrome, ce qui est impossible car le nombre de a avant b tend vers $+\infty$ lorsque k tend vers $+\infty$ alors que le nombre de a après b est fixe ! Conclusion $Pal(\Sigma)$ n'est pas rationnel.

Commentaires : Question classique aux concours CentraleSupElec et Mines-Ponts. Vous l'avez probablement fait en cours. Le lemme de pompage (ou de l'étoile) est un outil courant pour prouver qu'un langage n'est pas rationnel. N'hésitez pas à le travailler. Au final, la question est sélective.

Q9. L'automate $A_{q,q'} = (Q, \{q\}, \{q'\}, T)$ reconnaît manifestement $L_{q,q'}$ donc $L_{q,q'}$ est reconnaissable. Un mot w appartient à L si et seulement s'il est reconnu par A si et seulement s'il existe un état initial q de A et un état final q' de A et un chemin de q à q' lisant w si et seulement s'il existe un état initial q de A , un état final q' de A tel que w appartient à $L_{q,q'}$ d'où

$$L = \bigcup_{q \in I, q' \in F} L_{q,q'}.$$

Commentaires : Question relativement aisée ... si vous maîtriser correctement la notion d'automates et langages reconnus associés.

Q10. Si u est un mot sur Σ alors $m = u\tilde{u}$ est un palindrome sur Σ et son nombre de lettres est pair (donc c'est un mot de $(\Sigma^2)^*$) d'où l'inclusion

$$\{u\tilde{u}, u \in \Sigma^*\} \subset Pal(\Sigma) \cap (\Sigma^2)^*.$$

Soit m un palindrome qui a un nombre pair de lettres. Il s'écrit $m = m_1 m_2 \cdots m_{2n}$ (nombre pair de lettres) et, comme c'est un palindrome, on a :

$$m_1 = m_{2n}, m_2 = m_{2n-1}, m_3 = m_{2n-2}, \dots, m_{n-1} = m_{n+2}, m_n = m_{n+1}$$

ce qui permet d'écrire :

$$m = m_1 m_2 m_3 \cdots m_n m_n m_{n-1} \cdots m_3 m_2 m_1 = u\tilde{u}$$

avec $u = m_1 m_2 m_3 \cdots m_n$ ce qui montre l'inclusion réciproque et permet de conclure à cette question.

Commentaires : Question sélective. Le candidat qui travaille par équivalence montre qu'il ne comprend pas qu'il ne peut s'agir d'une équivalence donc il montre son incompréhension ! Soyez rigoureusement dans vos raisonnements et travailler par double implication, chacune devant être correctement argumenté (c'est une preuve qui est attendu, non pas réénoncer ce qu'il faut établir, les arguments doivent être fondés sur les définitions et les résultats pré-établis).

Q11. On a

$$D(a^*b) = \{a^n b b a^n, n \in \mathbb{N}\} = \{a^n b^2 a^n, n \in \mathbb{N}\}.$$

Soit w un mot de $R(a^*b^*a^*)$ alors $w\tilde{w}$ appartient à $a^*b^*a^*$ donc s'écrit $w\tilde{w} = a^p b^q a^r$. Si $p \geq 1$ et $q \geq 1$ alors w commence par un certain nombre de a consécutif puis des b alors $\tilde{w}$ commence par des b consécutifs. Il ne peut avoir de lettres a ensuite sinon il s'écrit $w = a^s b^t a^u$ avec $s \geq 1, t \geq 1, u \geq 1$ donc

$$w\tilde{w} = a^s b^t a^{2u} b^t a^s$$

n'appartient pas à $a^*b^*a^*$. Ainsi, nous avons l'inclusion $R(a^*b^*a^*) \subset a^*b^*$. L'inclusion réciproque est immédiate puisque si $w \in a^*b^*$ alors $\tilde{w} \in b^*a^*$ donc

$$w\tilde{w} \in a^*b^*b^*a^* = a^*b^*a^*.$$

Ainsi, nous avons démontré que

$$R(a^*b^*a^*) = a^*b^*.$$

Commentaires : Question de difficulté standard pour ce concours.

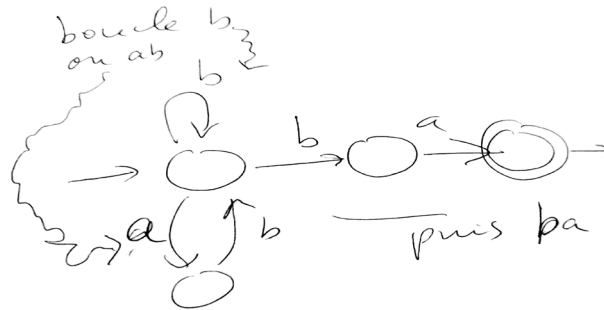
Q12. Les langages $D(L)$ ne sont pas nécessairement reconnaissables. Par exemple si $L = a^*b$, $D(L)$ ne l'est pas (même preuve que pour la question 8, remplacer b par b^2 dans celle-ci). Soit w un mot de $R(L)$ alors $w\tilde{w} \in L$ donc il existe un état initial i et un état final f tel que $w\tilde{w} \in L_{i,f}$. Si on désigne par q l'état où aboutit la lecture de w depuis l'état initial i alors $w \in L_{i,q}$ et $\tilde{w} \in L_{q,f}$ (on part de i pour aller à q pour lire w , ensuite on lit $\tilde{w}$ depuis q jusqu'à f). Comme $\tilde{w} \in L_{q,f}$, on peut affirmer que $w \in \widetilde{L_{q,f}}$. Réciproquement, si $w \in L_{i,q} \cap \widetilde{L_{q,f}}$ pour un certain état q , un certain état initial i et un certain état final f alors $w\tilde{w}$ est lu depuis l'état i pour aboutir à l'état f donc il appartient à $L_{i,f}$ ce qui assure que $w\tilde{w}$ appartient à L . Par conséquent,

$$R(L) = \bigcup_{i \in I, f \in F, q \in Q} (L_{i,q} \cap \widetilde{L_{q,f}}).$$

Comme les états $L_{i,q}$ et $L_{q,f}$ sont tous reconnaissables (question 9) donc le langage $\widetilde{L_{q,f}}$ aussi (question 3). L'intersection $L_{i,q} \cap \widetilde{L_{q,f}}$ est reconnaissable (les langages reconnaissables sont stables par intersection) donc $R(L)$ aussi (les langages reconnaissables sont stables par union).

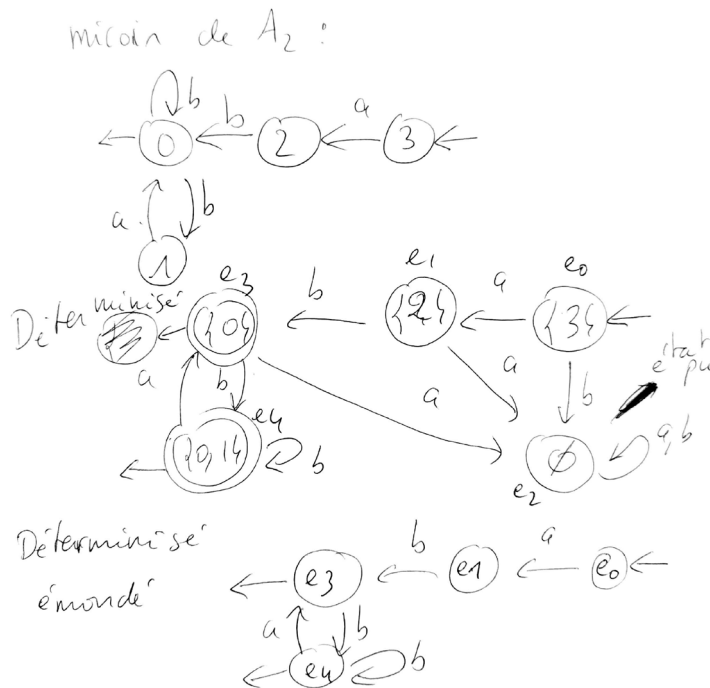
Commentaires : Question difficile car elle est ouverte et que le sujet donne une indication ... valable juste pour l'une des questions. Au final, beaucoup de pièges. La première se base sur un contre-exemple obtenu par les exemples du sujet, l'autre par l'indication. Si la première est accessible (surtout s'il n'y avait pas l'indication floue!), l'autre est destinée aux meilleures copies (les bons candidats sont aptes à produire une réponse convenable).

Q13.



Commentaires : Question élémentaire.

Q14. Dessin du miroir puis du déterminisé puis de son émondage. La table des transitions est après les dessins.

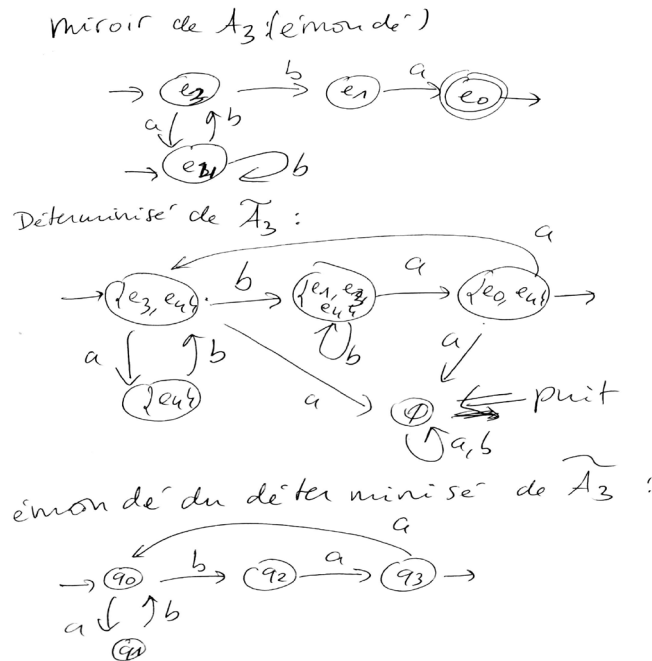


On dresse la table

etat/lettre	a	b
{3}	{2}	$\emptyset$
{2}	$\emptyset$	{0}
{0}	$\emptyset$	{0, 1}
{0, 1}	{0}	{0, 1}

Commentaires : Question élémentaire.

Q15. Dessin du miroir puis du déterminisé puis de son émondage. La table des transitions est après les dessins.



On dresse la table

etat/lettre	a	b
$\{e_3, e_4\}$	$\{e_4\}$	$\{e_1, e_3, e_4\}$
$\{e_4\}$	$\emptyset$	$\{e_3, e_4\}$
$\{e_1, e_3, e_4\}$	$\{e_0, e_4\}$	$\{e_1, e_3, e_4\}$
$\{e_0, e_4\}$	$\emptyset$	$\{e_3, e_4\}$

Commentaires : Question élémentaire.

Q16. A_2 reconnaît L_2 donc son miroir $\tilde{A}_2$ reconnaît $\tilde{L}_2$. Son déterminisé A_3 reconnaissant le même langage, on en déduit que A_3 reconnaît $\tilde{L}_2$. De même, $\tilde{A}_3$ reconnaît $\tilde{\tilde{L}_2} = L_2$ donc A_3 reconnaît L_2 .

Commentaires :

Q17.

```
let supprimer l =
  let rec appartient x l =
    match l with
```



```

|[] -> false
|y::q -> (y==x) || (appartient x q)
in
let rec aux l accu =
  match l with
  |[] -> accu
  |x::q when not (appartient x accu) -> aux q (x::accu)
  |x::q -> aux q accu
in aux l [];;

```

Commentaires : *Question de niveau standard et classique sans difficulté.*

Q18. La fonction *appartient l x* a une complexité en $O(|l|)$ (elle s'appelle au plus $|l|$ fois, les opérations en dehors de ces appels sont en nombre borné et de coût constant). La fonction *appartient* est exécutée $|l|$ fois par la fonction *aux* sur des listes de taille au plus $|l|$, les autres opérations étant en nombre borné et de coût constant, la complexité de *aux l []* est en $O(|l|) O(|l|) = O(|l|^2)$.

Commentaires : *Même remarque.*

Q19. On faut tester si 2^k est dans l'écriture binaire de q c'est-à-dire si la division de q par 2^k a pour chiffre des unités 1 (s'il est présent alors $q = q' + 2^k + q''$ avec $q' < 2^k$ et $q'' = 0$ ou $q'' \geq 2^{k+1}$ donc la quotient de q par 2^k vaut 1. S'il n'est pas présent alors $q = q' + q''$ et la quotient de q par 2^k est le quotient de q' par 2^k , ce dernier est au moins égal à $\frac{2^{k+1}}{2^k} = 2$)

```

let est_dans q k =
  (q/(pow.(k)) mod 2 == 1;;

```

Commentaires : *Question sélective car la difficulté est d'obtenir une complexité bornée. Le candidat doit analyser la situation pour trouver une bonne interprétation. Au final, cette question s'adresse aux bonnes copies.*

Q20. Si un état i n'est pas présent, on l'ajoute à l'ensemble donc on ajoute 2^i à l'entier qui lui correspond.

```

let numero l =
  let rec aux l accu =
    match l with
    |[] -> accu
    |i::q -> aux q (pow.(i)+accu)
  in aux (supprimer l) 0;;

```

Commentaires : *Question de difficulté standard.*

Q21.

```
let rec intersecte l k = match l with
| [] -> false
| i::q -> (est_dans i k) || (intersecte q k);;
```

Commentaires : *Question élémentaire.*

Q22. Lors de chaque transition (q, c, q') , si q' n'a pas été listé, on le rajoute à la liste des états accessibles.

```
let etat_suivant k l =
  let rec aux l acca accb =
    match l with
    | [] -> numero acca, numero accb
    | (q,c,q')::r when est_dans q k->
      if c == 'a' then aux r (q'::acca) accb
      else aux r acca (q'::accb)
  | (q,c,q')::r -> aux r acca accb
  in aux l [] [];;
```

Commentaires : *Question de difficulté standard.*

Q23.

```
let rec cherche k l =
  match l with
  | [] -> -1
  | (q,v)::r when q==k -> v
  | (q,v)::r -> cherche k r;;
```

Commentaires : *Question élémentaire.*

Q24. On détermine les états finaux de l'automate des parties (ceux qui contiennent un état final de l'automate initial) puis on convertit les transitions de l'automate initial à l'automate déterminisé, ce qui permet de faire ensuite la construction.

```
(*on détermine tous les états finaux de l'automate
déterminisé à l'aide de la liste des états finaux
de l'automate initial*)

let etats_finaux finaux n =
  let m = pow.(n) and fin = ref [] in
  for k=0 to m-1 do
    if intersecte finaux k then fin :=k::(!fin) done;
  !fin;;

(*convertir les transitions de l'automate vers les
```

```

transitions dans l'automate déterminisé*)

let transitions_determinise l n =
  let m = pow.(n) and t = ref [] in
  for k=0 to m-1 do
    let (ka,kb) = etat_suivant k l in
    if ka >=0 then t := (k,'a',ka)::(!t);
    if kb >=0 then t :=(k,'b',kb)::(!t)
  done;
  !t;;

let determinise a =
  {nb = pow.(a.nb) ;
   init = [numero a.init] ;
   final = etats_finaux a.final a.nb ;
   trans = transitions_determinise a.trans a.nb};;

```

Commentaires : Question sélective destinée aux candidats ayant une bonne maîtrise de la programmation et une vision claire de l'automate des parties (sa version utile i.e. la partie accessible).

Q25. La fonction *etat_suivant* $k\ l$ a une complexité en $O(|l|)$ (elle appelle au plus $|l|$ fois la fonction *aux*, en dehors de ces appels, les opérations sont en nombre fini et à coût constant (car *est_dans* l'est) donc elles ont un coût en $O(1)$ d'où un coût total en $|l| O(1) = O(|l|)$. Comme la liste l est de taille en $O(n^2)$ puis que les triplets (q, x, q') (avec q ayant n fois, q' aussi et x un nombre fini de choix $O(1)$ car l'alphabet est fini, ici 2 éléments) ont au plus $n \times O(1) \times n = O(n^2)$ choix au plus. Ainsi, *etat_suivant* $k\ l$ a une complexité au plus en $O(n^2)$.

La fonction *intersection* $l\ k$ a une complexité en $O(|l|)$ et comme l sera la liste des états finaux qui sont au plus N , sa complexité est en $O(N)$. Elle est exécutée N fois dans *etats_finaux_finaux* donc cette dernière fonction a une complexité totale en $NO(N) = O(N^2)$.

La fonction *transitions_determinise* $l\ n$ a une complexité en $NO(n^2) = O(Nn^2)$ (N itérations de la boucle, les opérations ont un coût constant en nombre fini sauf *etats_suivants* dont le coût est en $O(n^2)$).

Au total, la fonction *determinise* a a une complexité totale en

$$O(N^2) + O(Nn^2) = O(N^2 + Nn^2) = O(N(N + n^2)).$$

Commentaires : Question difficile car elle demande d'avoir traité la question précédente. Sa complexité est de difficulté standard pour ce concours..

Q26. Puisque $q \in \delta^* (\{I\}, u)$, il existe un chemin c dans A de $\{I\}$ à q qui lit le mot u . Comme tout état de $\tilde{A}$ est accessible et que f est un état initial de $\tilde{A}$, il existe un chemin de f à q dans l'automate $\tilde{A}$ c'est-à-dire des états $q_0, \dots, q_m$ et des lettres $v_0, \dots, v_{m-1}$ tels que

$$\{f\} = q_0 \xrightarrow{v_0} q_1 \rightarrow \dots q_{m-1} \xrightarrow{v_{m-1}} q_m = q$$

soit un chemin dans $\tilde{A}$ donc

$$q = q_m \xrightarrow{v_{m-1}} q_{m-1} \rightarrow \dots q_1 \xrightarrow{v_0} q_0 = \{f\}$$

est un chemin dans l'automate A c'est-à-dire que $\{f\} \in \delta^*(\{q\}, w)$ si on pose $w = v_{m-1} \cdots v_0$ i.e. qu'il y a un chemin c' dans l'automate A qui part de q à $\{f\}$ lisant le mot w . En concaténant les chemins c puis c' , on obtient un chemin dans l'automate de $\{I\}$ à f (état final de A qui reconnaît le langage L) lisant uw donc $uw \in L$.

Commentaires : Question de difficulté standard pour ce concours. N'hésitez pas à la retravailler si vous avez eu des difficultés à la traiter (ou pas réussi!).

Q27. Commençons par une remarque fondamentale. L'automate $\tilde{A}$ est accessible donc, pour tout état q de $\tilde{A}$ (donc de A), il existe un chemin $\tilde{c} : f \rightsquigarrow q$ dans $\tilde{A}$ partant de son unique état initial et aboutissant à q . Autrement dit, depuis tout état q de A , il existe un chemin $c : q \rightsquigarrow f$ de q vers l'unique état final de A . En outre, le caractère déterministe de $\tilde{A}$, si $\tilde{w}$ est le mot lu lors du chemin $\tilde{c}$ alors, pour tout état q' de A et tout chemin $\tilde{c}' : f \rightsquigarrow q'$ lisant $\tilde{w}$ alors $q = q'$. Autrement dit, dans l'automate A , si deux états q et q' de A aboutissent à f en lisant le même mot w alors $q = q'$!

Pour démontrer l'égalité souhaitée, on procède par double inclusion. Soit $q \in \delta^*(\{I\}, u)$ alors, il existe un mot w tel que

$$uw \in L \Leftrightarrow w \in u^{-1}L = v^{-1}L \Leftrightarrow vw \in L.$$

En outre, d'après la réponse à la question précédente,

- il existe un état initial $i \in I$ et un chemin $i \rightsquigarrow^u q$ lisant u ;
- il existe un mot w et un chemin $q \rightsquigarrow^w f$ lisant w .

Comme $vw \in L$, il existe un chemin $i' \rightsquigarrow^{vw} f$ lisant vw en partant d'un état initial i' et arrivant à l'état final f . Si on note q' l'état de ce chemin où l'on passe par la lecture de v alors :

- $q' \in \delta^*(\{I\}, v)$ dans l'automate A ;
- on dispose d'un chemin $q' \rightsquigarrow^w f$ lisant w .

Comme il y a aussi un chemin $q' \rightsquigarrow^w f$ lisant w , on peut affirmer que $q = q'$ donc $q \in \delta^*(\{I\}, v)$ d'où l'inclusion $\delta^*(\{I\}, u) \subset \delta^*(\{I\}, v)$. En échangeant les rôles de u et v , on obtient l'autre inclusion d'où l'égalité attendue.

Commentaires : Question difficile à destination des candidats ayant une bonne compréhension des automates. On n'oublie pas de travailler par implication et non par équivalence (ce que font trop de candidats).

Q28. Si A reconnaît L alors $\tilde{A}$ reconnaît $\tilde{L}$. Son déterminisé $\tilde{A}_{\text{det}} = B$ reconnaît $\tilde{L}$. Par le même raisonnement, on en déduit que $\tilde{B}_{\text{det}}$ reconnaît $\tilde{\tilde{L}} = L$ et vérifie (*) d'après la question précédente.

Commentaires : Question élémentaire.

Q29.

```
let minimal a =
  determinise (transpose (determinise (transpose a)));;
```

Commentaires : *Question élémentaire.*

Q30.

```
let rec lettre e = match e with
| Lettre(a) -> 1
| Union(e1,e2) -> (lettre e1) + (lettre e2)
| Concat(e1,e2) -> (lettre e1) + (lettre e2)
| Etoile(e1) -> lettre e1
| _ -> 0;;
```

Commentaires : *Question élémentaire.*

Q31.

```
let est_vide e = match e with
| Vide -> true
| Concat(f,g)-> (est_vide f) || (est_vide g)
| Union (f,g)-> (est_vide f) && (est_vide g)
| _ -> false;;
```

Commentaires : *Question élémentaire.*

Q32.

```
let su expr = match expr with
| Union( Vide , e ) -> e
| Union( e , Vide ) -> e
| _ -> expr;;

let sc expr = match expr with
| Concat( Vide , e ) -> Vide
| Concat( e , Vide ) -> Vide
| Concat(e,Epsilon) -> e
| Concat(Epsilon,e) -> e
| _ -> expr;;

let se expr = match expr with
| Etoile(Vide) -> Epsilon
| Etoile(Epsilon) -> Epsilon
| Etoile(Etoile e) -> Etoile e
| _ -> expr;;
```

Commentaires : *Question sélective car un nombre important de candidat a du mal à traiter tous les cas possibles.*

Q33. Il en faut $n + 1$ (1 pour passer de $b.\emptyset$ à $\emptyset$ puis 1 pour passer de $c + \emptyset$ à $\emptyset$ et il y a n additions!).

Commentaires : *Question élémentaire.*

Q34.

```
let rec simplifie e = match e with
| Concat(e1,e2) -> sc ( Concat(simplifie e1, simplifie e2))
| Union(e1,e2) -> su ( Union ( simplifie e1 , simplifie e2))
| Etoile(e) -> se (Etoile (simplifie e))
| e' -> e';;
```

Commentaires : *Question sélective car un nombre important de candidats ne pensent à simplifier les sous-expressions et/ou à appliquer les bonnes réductions su, etc.*

Q35.

```
let somme a b =
  let n = Array.length a and m = Array.length a.(0) in
  let c = Array.make_matrix n m Vide in
  for i = 0 to n-1 do
    for j=0 to m-1 do
      c.(i).(j) <- simplifie (Union(a.(i).(j) , b.(i).(j)))
    done;done;
  c;;
```

Commentaires : *Question élémentaire.*

Q36.

```
let produit a b =
  let n = Array.length a
  and p = Array.length a.(0)
  and q = Array.length b.(0) in
  let c = Array.make_matrix n q Vide in
  for i = 0 to n-1 do
    for j=0 to q-1 do
      for k=0 to p-1 do
        c.(i).(j) <- Union(c.(i).(j),
                           Concat(a.(i).(k) , b.(k).(j)))
      done; done;done;
  c;;
```

Commentaires : *La question est discriminante car ... tous les candidats ne se rappellent pas de la formule exacte du produit matriciel. Ensuite, il faut pour gérer la " somme " générale et penser à bien remplacer les additions par des unions et les produits par des concaténations. Au final, la question est réussie dans les bonnes copies.*

Q37. $L_{0,0} = (a^*bd^*c)^*$ (la boucle générale de 0 et qui revient pour la première fois à 0 boucle sur a puis lit b puis boucle avec d puis lit c , on itère autant de fois que nécessaire cette boucle).

$L_{0,1} = L_{0,0}bd^* = (a^*bd^*c)bd^*$ (l'état initial est 0, l'état final est 1, avant de passer pour la dernière fois à 1, on boucle de 0 à 0 c'est-à-dire on lit les mots de $L_{0,0}$ puis on passe de 0 à 1 en lisant b puis on boucle en lisant d puis on sort car on ne peut plus revenir à 0).

$L_{1,0} = d^*cL_{0,1} = d^*c(a^*bd^*c)bd^*$ (l'état initial est 1, l'état final est 0, de 1, on peut boucler autant de fois que l'on veut en lisant d puis on passe à 0 en lisant c , le mot qui reste à lire commence à l'état 0 et fini à l'état 1 donc il s'agit d'un mot de $L_{0,1}$).

$L_{1,1} = (d^*ca^*b)^*d^*$ (même raisonnement qu'avec $L_{0,0}$) ou choisir un raisonnement comme $L_{1,0}$ ou $L_{0,1}$. A vous de choisir.

Commentaires : Question de difficulté standard pour ce concours.

Q38. Complexité des sommes et produits. Soient M et M' sont deux matrices carrées à n lignes et n colonnes, C une matrice colonne à n lignes et L une matrice ligne à n colonnes, a une lettre le calcul de :

- $aM = (aM_{i,j})_{i,j}$ et $Ma = (M_{i,j}a)_{i,j}$ nécessite n^2 calculs de produit de deux facteurs donc un nombre total d'opération en $O(n^2)$;
- $M + M' = (M_{i,j} + M'_{i,j})_{i,j}$ nécessite n^2 calculs de sommes de deux termes donc un nombre total d'opération en $O(n^2)$;
- MC nécessite n calculs de sommes du type $\sum_{j=1}^n M_{i,j}C_j$, chaque somme nécessite n produits $M_{i,j}C_j$ et $n-1$ additions soit un nombre d'opérations en $O(n)$ donc le nombre total d'opérations en $nO(n) = O(n^2)$;
- LM nécessite aussi $O(n^2)$ opérations (sa transposition se ramène au cas précédent) ;
- $LC = \sum_{i=1}^n L_iC_i$ nécessite $O(n)$ opérations
- $CL = (C_iL_j)_{i,j}$ nécessite $O(n^2)$ opérations (n^2 multiplications)

Complexité des opérations, hors $*$, dans le passage de M à M^* . Par construction, M a n lignes, n colonnes, D a $n-1$ lignes et $n-1$ colonnes, C a $n-1$ lignes et 1 colonne c'est une matrice colonne, B a 1 ligne et $n-1$ colonnes donc c'est une matrice ligne.

D^* est une matrice à $n-1$ lignes et $n-1$ colonnes donc le produit BD^*C a un coût en $O(n^2)$ opérations donc $a + BD^*C$ a un coût en $O(n^2)$. Le coût des opérations, hors $*$ matricielles, du calcul de A' est en $O(n^2)$.

Ca^*B a un coût en $O(n^2)$, $D + Ca^*B$ aussi et comme c'est une matrice de $n-1$ lignes et $n-1$ colonnes alors que B est une matrice ligne à $n-1$ colonnes, le produit $B(D + Ca^*B)^*$ a un coût en $O(n^2)$. Le produit $aB(D + Ca^*B)^*$ a un coût en $O(n^2)$. Ainsi, le coût des opérations (hors $*$ matricielles) dans le calcul de B' est en

$$O(n^2) + O(n^2) + O(n^2) = O(n^2).$$

Par des calculs semblables, on observe que les calculs de C' et D' , hors $*$ matricielles, sont en $O(n^2)$.

Si on stocke le calcul de D^* et $(D + Ca^*B)^*$, chacun de ces calculs a un coût $C(n-1)$ donc un coût total $2C(n-1)$. Ces résultats étant stockés, les calculs restant pour produire M^*

sont en $O(n^2)$ (découpage de M en A, B, C, D , reconstitution de M^* à partir des A', B', C', D' , puis les calculs, hors $*$ matricielles de A', B', C', D') ont un coût en $O(n^2)$ (obtenues par un nombre fixe d'additions ou produit matriciel de matrices de taille au plus n). Par conséquent, le coût total du calcul de M^* est $2C(n-1) + O(n^2)$ d'où

$$C(n) = 2C(n-1) + O(n^2).$$

En effet, dans A' , comme BD^*C est un mot, son étoile a un coût constant donc il n'y a que deux $*$ matricielles à calculer.

En itérant la relation de récurrence, on a

$$\begin{aligned} C(n) &= 2C(n-1) + O(n^2) \\ &= 2\left(2C(n-2) + O((n-1)^2)\right) + O(n^2) \\ &= 2^2C(n-2) + O(2(n-1)^2 + n^2) \\ &= 2^2\left(2C(n-3) + O((n-2)^2)\right) + O(2(n-1)^2 + n^2) \\ &= 2^3C(n-3) + O(2^2(n-2)^2 + 2(n-1)^2 + n^2) \\ &\vdots \\ &= 2^n C(0) + O(2^n 0^2 + 2^{n-1} 1^2 + 2^{n-2} 2^2 + \dots + 2^2 (n-2)^2 + 2(n-1) + n^2). \end{aligned}$$

On en déduit que $C(n) = O(n2^n)$ (il y a n termes dont le plus grand est en 2^n).

Commentaires : Question très difficile à ce niveau. En mathématique, on peut montrer que l'on a en fait $C(n) = O(2^n)$ mais cela n'a aucune importance car la complexité reste toujours au moins exponentielle (donc inexploitable pour des valeurs un tant soit peu grande) et qu'il me paraît extrêmement peu probable qu'un étudiant au concours donne ce résultat optimal. Tout candidat fournissant un résultat en $O(n^k 2^n)$ sera fortement récompensé.

Q39. Toutes les matrices intervenant dans les calculs ont une taille au plus n donc les opérations matricielles (hors $*$ matricielles) auront une complexité totale en $O(n^3)$ (je laisse le lecteur s'en convaincre) Les $*$ matricielles à calculer (et à stocker définitivement !) sont celles de $D, A + BD^*C, A$ et $D + CA^*B$ sur des matrices carrées de taille $\frac{n}{2}$ soit un coût total $4C\left(\frac{n}{2}\right)$. Ainsi, on en déduit que

$$C(n) = 4C\left(\frac{n}{2}\right) + O(n^3).$$

Si n est une puissance de 2 (pour absorber toutes les divisions par 2), il existe un entier N tel que $n = 2^N$. Si on pose $c_N = C(2^N)$, on a

$$\begin{aligned} c_N &= C(2^N) = 4C\left(\frac{2^N}{2}\right) + O((2^N)^3) = 4C(2^{N-1}) + O(8^N) \\ &= 4c_{N-1} + O(8^N). \end{aligned}$$

En itérant cette relation, on obtient

$$\begin{aligned}
 c_N &= 4c_{N-1} + O(8^N) = 4(4c_{N-2} + O(8^{N-1})) + O(8^N) \\
 &= 4^2 c_{N-2} + O\left(\frac{1}{2}8^N\right) + O(8^N) = 4^2 c_{N-2} + O(2 \times 8^N) \\
 &= 4^2 (4c_{N-3} + O(8^{N-2})) + O(2 \times 8^N) \\
 &= 4^3 c_{N-3} + O(8^N) + O(2 \times 8^N) = 4^3 c_{N-3} + O(3 \times 8^N) \\
 &\vdots \\
 &= 4^N c_0 + O(N \times 8^N) = O(N \times 8^N)
 \end{aligned}$$

Comme $n = 2^N$, $n^3 = 8^N$ et $N = \log_2(N)$ on peut écrire

$$c(n) = C(2^N) = c_N = O(N \times 8^N) = O(n^3 \log_2(n))$$

Commentaires : Même remarque que précédemment. On peut montrer en mathématiques que $c(n) = O(n^3)$ mais la déviation $\log_2(n)$ n'est pas trop dramatique et il serait étonnant qu'un candidat le démontre seul (sauf s'il connaît un théorème général).

Q40. On effectue les mêmes découpages (à l'unité près selon la parité des entiers n rencontrés), les matrices A, B, C, D changent de taille fréquemment mais les complexités sont conservés (car la moitié supérieurs ou inférieure de n est asymptotiquement équivalente à $\frac{n}{2}$) et les calculs précédents perdurent donc la complexité restera en $O(n^3 \log_2(n))$ (ou $O(n^3)$ si on connaît le résultat mathématique).

Commentaires : Même remarque que précédemment.

Q41.

```

let rec etoile m =
  let n = Array.length m in
  match n with
  | n when n <= 1 -> [[|Etoile(m.(0).(0))|]]
  | n -> let a,b,c,d = decouper m (n/2) (n-n/2) in
    let ae = etoile a and de = etoile d in let
      a' = etoile(somme a (produit (produit b de) c))
    and
      b' = produit(produit ae b)
      (etoile(somme d (produit(produit c ae) b)))
    and
      c' = produit (produit de c)
      (etoile(somme a (produit (produit b de) c)))
    and
      d' = etoile(somme d (produit (produit c ae) b))
    in let m = recoller a' b' c' d'
    in for i=0 to n-1 do
      for j=0 to n-1 do
        m.(i).(j) <- simplifie m.(i).(j); done; done;
      m;;

```

Commentaires : Question destinée aux meilleures copies.

Q42. Par définition du produit matriciel, $M_A^* Y$ est une matrice colonne dont la i^e ligne vaut

$$(M_A^* Y)_i = \sum_{j=0}^{n-1} (M_A^*)_{i,j} Y_j.$$

X étant une matrice ligne, le produit $X (M_A^* Y)$ est une matrice 1×1 dont le seul coefficient est :

$$(X M_A^* Y)_{0,0} = \sum_{i=0}^{n-1} X_i (M_A^* Y)_i = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} X_i (M_A^*)_{i,j} Y_j.$$

Le sujet indique que, pour chaque (i, j) , on a : $\mathcal{L}([M_A]_{i,j}) = L_{i,j}$. Comme $\mathcal{L}$

$$(e + e') = \mathcal{L}(e) + \mathcal{L}(e') \text{ et } \mathcal{L}(ee') = \mathcal{L}(e) \mathcal{L}(e'),$$

on peut écrire :

$$\begin{aligned} \mathcal{L}[(X M_A^* Y)_{0,0}] &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \mathcal{L}(X_i) \mathcal{L}[(M_A^*)_{i,j}] \mathcal{L}(Y_j) \\ &= \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} \mathcal{L}(X_i) L_{i,j} \mathcal{L}(Y_j). \end{aligned}$$

Or, d'après la question 9, on a

$$L_A = \bigcup_{i \in I, f \in F} L_{i,j} = \sum_{i \in I, j \in F} L_{i,j}.$$

On choisit alors X de sorte que $X_i = \emptyset$ (donc $\mathcal{L}(X_i) = \emptyset$) si i n'est pas un état initial et $X_i = \varepsilon$ (donc $\mathcal{L}(X_i) = \varepsilon$) si i est un état initial. De même, on choisit Y de sorte que $Y_i = \emptyset$ si j n'est pas un état final et $Y_i = \varepsilon$ si j est un état initial. Par conséquent, on peut affirmer que

$$\begin{aligned} \mathcal{L}(X_i) L_{i,j} \mathcal{L}(Y_j) &= L_{i,j} \text{ si } i \in I \text{ et } j \in F \\ \mathcal{L}(X_i) L_{i,j} \mathcal{L}(Y_j) &= \emptyset \text{ sinon} \end{aligned}$$

donc, pour ce choix de X et Y , on a :

$$\mathcal{L}[(X M_A^* Y)_{0,0}] = L_A.$$

Commentaires : Même remarque que précédemment.

Q43.

```
let langage a =
  let n = a.nb in
  let mA = Array.make_matrix n n Vide
  in let xA = Array.make_matrix 1 n Vide
```

```

in let yA = Array.make_matrix n 1 Vide
in
let rec matA l = match l with
| [] -> ()
| (i,c,j)::r -> mA.(i).(j)<- Union (mA.(i).(j),
                                   Lettre c) ; matA r

in
let rec matX ini = match ini with
| [] -> ()
| i::r -> xA.(0).(i)<-Epsilon ; matX r
in
let rec matY fin = match fin with
| [] -> ()
| i::r -> yA.(i).(0)<-Epsilon ; matY r
in matA a.trans ; matX a.init ; matY a.final;
simplifie ((produit xA (produit
                    (etoile mA) yA)).(0).(0));;

```

Commentaires : *Même remarque que précédemment.*

Q44. $E = (ab + b)^* ba$.

Calcul de $\partial_a(E)$. Comme $\varepsilon \in \mathcal{L}((ab + b)^*)$, on a

$$\begin{aligned}\partial_a(E) &= \partial_a((ab + b)^*) \cdot \{ba\} \cup \partial_a(ba) \\ &= \partial_a((ab + b)) \cdot \{(ab + b)^*\} \cdot \{ba\} \cup [\partial_a(b) \cdot \{a\}]\end{aligned}$$

puisque $\varepsilon \notin \mathcal{L}(ba)$. Par définition, comme $a \neq b$, on a $\partial_a(b) = \emptyset$ donc

$$\partial_a(E) = \partial_a((ab + b)) \cdot \{(ab + b)^*\} \cdot \{ba\}$$

et

$$\begin{aligned}\partial_a((ab + b)) &= \partial_a(ab) \cup \underbrace{\partial_a(b)}_{=\emptyset} = \partial_a(ab) \\ &= \partial_a(a) \cdot \{b\} = \varepsilon \cdot \{b\} = \{b\}\end{aligned}$$

d'où

$$\partial_a(E) = \{b\} \cdot \{(ab + b)^*\} \cdot \{ba\} = \{b(ab + b)^* ba\}.$$

De même, comme $\varepsilon \in \mathcal{L}((ab + b)^*)$, on a

$$\begin{aligned}\partial_b(E) &= \partial_b((ab + b)^*) \cdot \{ba\} \cup \partial_b(ba) \\ &= \partial_b(ab + b) \cdot \{(ab + b)^*\} \cdot \{ba\} \cup \partial_b(b) \cdot \{a\} \\ &= \partial_b(ab + b) \cdot \{(ab + b)^*\} \cdot \{ba\} \cup \{a\}\end{aligned}$$

car $\partial_b(b) = \varepsilon$. En outre, on a

$$\begin{aligned}\partial_b((ab + b)) &= \partial_b(ab) \cup \underbrace{\partial_b(b)}_{=\varepsilon} = \partial_b(ab) \cup \{\varepsilon\} \\ &= \underbrace{\partial_b(a)}_{=\emptyset} \cdot \{b\} \cup \{\varepsilon\} = \varepsilon\end{aligned}$$

donc

$$\partial_b(E) = \{(ab+b)^*\} \cdot \{ba\} \cup \{a\} = \{(ab+b)^*ba, a\}.$$

Commentaires : Question de difficulté standard. Au final, elle est très sélective car il faut que le candidat vienne jusqu'à cette question et qu'il ait l'esprit frais ! Bravo à elle et à lui.

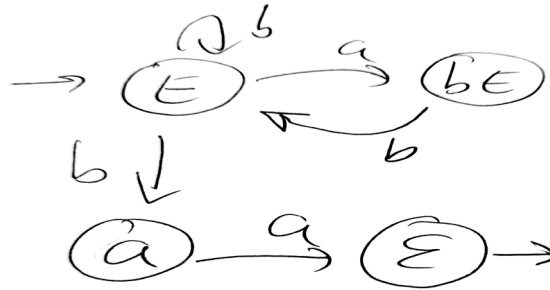
Q45. Comme

$$\partial_a(E) = \{b(ab+b)^*ba\} \text{ et } \partial_b(E) = \{(ab+b)^*ba, a\},$$

si on pose $E' = (ab+b)^*ba$ alors

$$\partial_a(E') = \emptyset, \partial_b(E') = E, \partial_a(a) = \{\varepsilon\}, \partial_b(b) = \emptyset, \partial_a(\varepsilon) = \partial_b(\varepsilon) = \emptyset,$$

on obtient l'automate suivant (en ayant émondé l'état $\emptyset$ qui est un état puit).



Commentaires : Même remarque que précédemment.

Q46. Soit w un mot alors

$$w \in v^{-1}u^{-1}L \Leftrightarrow vw \in u^{-1}L \Leftrightarrow uvw \in L \Leftrightarrow w \in (uv)^{-1}L$$

d'où l'égalité ensembliste $v^{-1}u^{-1}L = (uv)^{-1}L$.

Commentaires : Question élémentaire.

Q47. Vérifions la formule pour une lettre a et E une expression régulière en procédant par récurrence sur le nombre d'opérations $(+, \cdot, \text{etc})$ formant E .

Si $E = \emptyset$ alors $\partial_a(E) = \emptyset$ donc

$$\mathcal{L}(\partial_a(E)) = \mathcal{L}(\emptyset) = \emptyset = a^{-1}\emptyset = a^{-1}\mathcal{L}(E)$$

Si $E = \varepsilon$ alors $\partial_a(E) = \emptyset$ et $a^{-1}\mathcal{L}(\varepsilon) = a^{-1}\{\varepsilon\} = \emptyset$ (aucun mot avec au moins une lettre n'apparaît dans le langage $\{\varepsilon\}$) donc $\mathcal{L}(\partial_a(E)) = a^{-1}\mathcal{L}(E)$.

Si $E = \{a\}$ (resp. $E = \{b\}$ avec $b \neq a$) alors $\partial_a(E) = \{\varepsilon\}$ (resp. $\emptyset$) donc $\mathcal{L}(\partial_a(E)) = a^{-1}\mathcal{L}(E)$.

Supposons la propriété $\mathcal{L}(\partial_a(F)) = a^{-1}\mathcal{L}(F)$ vraie pour $F = U$ et $F = V$.

Si $E = U + V$ alors

$$\begin{aligned} \partial_a(E) &= \partial_a(U) \cup \partial_a(V) \Rightarrow \mathcal{L}(\partial_a(E)) = \mathcal{L}(\partial_a(U)) \cup \mathcal{L}(\partial_a(V)) \\ &= a^{-1}\mathcal{L}(U) \cup a^{-1}\mathcal{L}(V) = a^{-1}\mathcal{L}(U + V) = a^{-1}\mathcal{L}(E) \end{aligned}$$

Si $E = U^*$ alors

$$\begin{aligned}\partial_a(E) &= \partial_a(U) \cdot \{U^*\} \Rightarrow \mathcal{L}(\partial_a(E)) = \mathcal{L}(\partial_a(U)) \cdot \{U^*\} \\ &= a^{-1}\mathcal{L}(U)\mathcal{L}(U^*) = a^{-1}\mathcal{L}(U^*) = a^{-1}\mathcal{L}(E).\end{aligned}$$

En effet, pour tout langage L , on a :

$$\begin{aligned}w &\in a^{-1}L.L^* \Leftrightarrow aw \in L.L^* \Rightarrow aw \in L^* \Rightarrow w \in a^{-1}L^* \\ w &\in a^{-1}L^* \Rightarrow aw \in L^* \text{ et } aw \text{ n'est pas vide donc } aw \in L.L^* \\ &\Leftrightarrow w \in a^{-1}L.L^*\end{aligned}$$

d'où $a^{-1}L.L^*$

Si $E = U.V$ avec $\varepsilon \notin \mathcal{L}(U)$ alors

$$\begin{aligned}\partial_a(E) &= \partial_a(U) \cdot \{V\} \Rightarrow \mathcal{L}(\partial_a(E)) = \mathcal{L}(\partial_a(U)) \cdot \mathcal{L}(V) \\ &= a^{-1}\mathcal{L}(U) \cdot \mathcal{L}(V) = a^{-1}\mathcal{L}(U.V) = a^{-1}\mathcal{L}(E).\end{aligned}$$

Si $\varepsilon \in \mathcal{L}(E)$ alors

$$\begin{aligned}\partial_a(E) &= \partial_a(U) \cdot \{V\} \cup \partial_a(V) \Rightarrow \mathcal{L}(\partial_a(E)) = \mathcal{L}(\partial_a(U)) \cdot \mathcal{L}(V) \cup \mathcal{L}(\partial_a(V)) \\ &= a^{-1}\mathcal{L}(U) \cdot \mathcal{L}(V) \cup a^{-1}\mathcal{L}(V) = a^{-1}\mathcal{L}(U) \cdot \mathcal{L}(V) = a^{-1}\mathcal{L}(U.V) = a^{-1}\mathcal{L}(E).\end{aligned}$$

(si $\varepsilon \in L$ alors $a^{-1}LL' = a^{-1}(L + \varepsilon)L' = a^{-1}LL' + a^{-1}L'$).

Ainsi, la formule est vraie pour une expression donc pour une réunion d'expression ;

On itère alors sur les lettres du mot w . C'est immédiat si w est le mot vide. Sinon w admet une dernière lettre a donc il s'écrit $w = w'a$ avec w' un certain mot et supposons la propriété vérifiée pour w' c'est-à-dire

$$\mathcal{L}(\partial_{w'}(S)) = (w')^{-1}\mathcal{L}(S)$$

(on présume l'hypothèse de récurrence vraie). Par définition, si E est une expression régulière, on a :

$$\begin{aligned}\mathcal{L}(\partial_w(E)) &= \mathcal{L}(\partial_a(\partial_{w'}(E))) = \mathcal{L}(\partial_a(S')) \quad (S' = \partial_{w'}(E)) \\ &= \mathcal{L}\left(\bigcup_{E \in S'} \partial_a(E)\right) = \bigcup_{E \in S'} \mathcal{L}(\partial_a(E)) \\ &= \bigcup_{E \in S'} a^{-1}\mathcal{L}((E)) = a^{-1} \bigcup_{E \in S'} \mathcal{L}((E)) \\ &= a^{-1}\mathcal{L}(S') = a^{-1}\mathcal{L}(\partial_{w'}(E)) = a^{-1}(w')^{-1}\mathcal{L}(E) \\ &= (w'a)^{-1}\mathcal{L}(E) = w^{-1}\mathcal{L}(E).\end{aligned}$$

Ainsi, la formule est vraie pour tout mot et toute formule donc elle est vraie pour tout mot et toute famille S de formules.

Commentaires : Question difficile car quel candidat peut avoir traité déjà 47 questions en moins de 4 h. Félicitations à cet impétrant (je n'ose pas mettre le pluriel ...)

Q48. On procède par récurrence sur la longueur du mot w . Si $w = \varepsilon$ alors $\partial_w(E) = \{E\}$ qui est bien le seul état accessible depuis E en lisant aucune lettre. Supposons la propriété vraie pour tout mot de longueur $< n$ avec n un entier $n \geq 1$. Soit w un mot de longueur n , il

s'écrit $w = w'a$ pour un certain mot w' et une certaine lettre a alors $\partial_w(E) = \partial_a(\partial_{w'}(E))$. Par hypothèse de récurrence, $\partial_{w'}(E)$ est l'ensemble des états accessible depuis E en lisant w' . Soit E_1 l'un de ces états alors les états E_2 pour lequel on peut lire a sont les états E_2 égaux à $\partial_a(E_1)$ avec E_1 l'un des états de $\partial_{w'}(E)$ (par définition de l'automate) donc les états accessibles depuis E en lisant w forment l'ensemble

$$\partial_a(\partial_{w'}(E)) = \partial_w(E).$$

Commentaires : *Même remarque que précédemment.*

Q49. Soit w un mot, on a :

$$\begin{aligned} w &= \varepsilon w \in \mathcal{L}(E) \underset{\text{q46}}{\Leftrightarrow} \varepsilon \in w^{-1}\mathcal{L}(E) \underset{\text{q.47}}{\Leftrightarrow} \varepsilon \in \mathcal{L}(\partial_w(E)) \\ &\Leftrightarrow \exists E_f \in \partial_w(E) \text{ tel que } \varepsilon \in \mathcal{L}(E_f) \text{ (q48)} \Leftrightarrow E_f \in F \end{aligned}$$

ce qui est équivalent à ce que w soit reconnu par l'automate d'Antimirov.

Commentaires : *Même remarque que précédemment.*

Q50. La réponse à la première question résulte d'une récurrence forte sur le nombre d'opérations intervenant dans l'expression E . Je le laisse au lecteur (même jeu qu'à la question 47). L'automate d'Antimirov contient au plus $\|E\| + 1$ états.

Commentaires : *Même remarque que précédemment.*



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2022

ÉPREUVE D'INFORMATIQUE COMMUNE

Durée de l'épreuve : 2 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

Cette épreuve est commune aux candidats des filières MP, PC et PSI.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE COMMUNE

L'énoncé de cette épreuve comporte 12 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Modélisation numérique d'un matériau magnétique

Certains matériaux particuliers peuvent acquérir des états magnétiques qualifiés de paramagnétique et ferromagnétique. Le matériau est dit **paramagnétique** lorsqu'il ne possède pas d'aimantation spontanée, mais acquiert une aimantation sous l'effet d'un champ magnétique extérieur. Il est dit **ferromagnétique** lorsqu'il possède une aimantation même en l'absence de champ magnétique extérieur. Dans ces matériaux, la température T joue un rôle crucial : si T est supérieure à une température particulière T_C , nommée température de Curie, le matériau adopte un état paramagnétique. Dans le cas contraire ($T < T_C$), il adopte un état ferromagnétique. C'est par exemple le cas du fer, pour lequel la transition entre les deux états se produit à $T_C = 1043$ kelvin.

Dans un matériau magnétique, les divers éléments magnétiques (électrons, atomes) possédant un moment magnétique créent une aimantation moyenne à l'intérieur du matériau. Nous admettrons les principaux résultats de la théorie du paramagnétisme.

Ce sujet est constitué de 4 parties. Dans la première, on cherche à obtenir l'aimantation moyenne du matériau en fonction de la température à partir d'une formule théorique connue. Dans la seconde, on recherche dans une base de données les propriétés de matériaux magnétiques. Dans la troisième, on cherche à développer une modélisation microscopique d'un matériau magnétique à deux dimensions pour retrouver ce comportement (modèle d'Ising). Dans la quatrième, on s'intéresse aux domaines magnétiques du matériau (nommés domaines de Weiss).

Une courte documentation de quelques fonctions utiles est disponible à la fin du sujet.

Les candidats sont fortement incités à expliciter brièvement leurs programmes à l'aide de quelques commentaires bien placés.

L'utilisation du module numpy n'est pas autorisée.



Partie I : Transition paramagnétique/ferromagnétique sans champ magnétique extérieur

La théorie des matériaux indique que, dans un matériau ferromagnétique, l'aimantation volumique moyenne du matériau est donnée par :

$$M = N\mu \tanh\left(\frac{\mu B}{k_B T}\right) \quad (1)$$

où N est le nombre d'atomes par unité de volume, B est la valeur du champ magnétique à l'intérieur du matériau, μ le moment magnétique des atomes ou des ions, k_B la constante de Boltzmann, T la température et $\tanh : x \mapsto \frac{\exp(x) - \exp(-x)}{\exp(x) + \exp(-x)}$ la fonction tangente hyperbolique.

On considère une situation sans champ magnétique extérieur. Le champ magnétique local à l'intérieur du matériau ferromagnétique est donc celui créé par le matériau lui-même. On admet que ce champ magnétique est proportionnel à l'aimantation moyenne dans le matériau ($B = \lambda M$), et on obtient alors : $M = N\mu \tanh\left(\frac{\mu \lambda M}{k_B T}\right)$.

En introduisant l'aimantation réduite $m = \frac{M}{N\mu}$ et la température réduite $t = \frac{k_B T}{N\mu^2 \lambda} = \frac{T}{T_C}$, l'équation 1 devient :

$$m = \tanh(m/t) \quad (2)$$

Cette équation d'inconnue m ne possède pas de solution analytique : si on veut connaître une approximation de l'aimantation moyenne dans le matériau, il est donc nécessaire de la résoudre numériquement par une méthode de recherche de zéro.

1. Écrire les instructions nécessaires pour importer exclusivement les fonctions exponentielle (`exp`) et tangente hyperbolique (`tanh`) du module `math`, ainsi que les fonctions `randrange` et `random` du module `random`. Ces fonctions seront ainsi utilisables dans tous les programmes que vous écrirez ultérieurement.
2. À partir de l'équation 2, indiquer une équation `f(x, t) = 0`, d'inconnue `x` que l'on doit résoudre et écrire en Python la définition de la fonction `f` correspondante (paramètres `x` et `t`, valeur renvoyée `f(x, t)`).
3. Écrire une fonction `dicho(f, t, a, b, eps)` qui calcule une valeur approchée à `eps` près du zéro d'une fonction `f(m, t)` de variable `x` et de paramètre `t` fixé sur un intervalle $[a, b]$. On supposera pour simplifier que la fonction dont on recherche le zéro est continue et s'annule une fois et une seule sur l'intervalle $[a, b]$.
4. Établir l'expression de la complexité temporelle asymptotique de la fonction `dicho` en fonction de `a`, `b` et `eps`.

Une étude mathématique simple permet de prouver que l'équation $m = \tanh(m/t)$ n'a de solution $m > 0$ que pour $0 < t < 1$, ce qui revient à dire que le matériau ne possède une aimantation non nulle que pour une température inférieure à la température de Curie. On admet ainsi que si $t \geq 1$ alors $m = 0$. De plus, pour $t < 1$, on admet que la solution $m = 0$ ne doit pas être prise en compte car elle correspond à une solution instable.

5. En utilisant la fonction `dicho`, écrire une fonction `construction_liste_m(t1, t2)` qui construit et retourne une liste de 500 solutions de l'équation (1), pour t variant linéairement de t_1 à t_2 (bornes incluses). On cherchera les valeurs de m à 10^{-6} près avec un intervalle de recherche initial $m \in [0.001, 1]$.

En traçant l'aimantation m en fonction de la température t , on obtient le graphe de la figure 1 permettant de visualiser les domaines ferromagnétique ($t < 1$) et paramagnétique ($t \geq 1$).

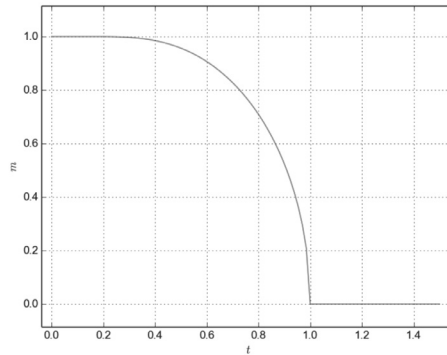


FIGURE 1 – aimantation réduite m en fonction de la température réduite t

Partie II : Recherche dans une base de données de matériaux magnétiques

Il existe des bases de données contenant les propriétés de nombreux matériaux, dont des propriétés magnétiques. Dans cette partie, on donne un modèle simplifié d'une telle base, et on souhaite effectuer quelques requêtes sur celle-ci.

La base de données possède la structure suivante :

- La table **matériaux** contient un champ **id_materiau**, clé primaire de la table de valeur entière, un champ **nom** de type chaîne de caractères pour le nom du matériau et un champ **t_curie** de valeur entière pour la température de Curie du matériau en kelvin.

id_materiau	nom	t_curie
4534	cobalt	1 388
1254	dioxyde de chrome	386
8713	nickel	627
8284	YIG	560
...	...	...

- La table **fournisseurs**, contenant un champ **id_fournisseur**, clé primaire de type entier qui précise le code de chaque fournisseur, et un champ **nom_fournisseur** de type chaîne de caractères pour le nom du fournisseur.

id_fournisseur	nom_fournisseur
145	Worldwide Materials
13	Materials Company
...	...

- La table **prix** qui contient un champ **id_prix**, clé primaire de type entier, un champ **id_mat** dont les valeurs sont incluses dans l'ensemble des valeurs de la clé **id_materiau** de la table **matériaux**, un champ **id_four** dont les valeurs sont incluses dans l'ensemble des valeurs de la clé **id_fournisseur** de la table **fournisseurs**, et un champ **prix_kg** de type flottant qui précise le prix au kg que ce fournisseur propose pour ce matériau, en euros. Un fournisseur qui ne propose pas un matériau donné n'a pas d'entrée correspondante dans cette table.

id_prix	id_mat	id_four	prix_kg
1	4567	145	50.40
2	8671	13	1357.30
3	1763	145	52.75
...	...	...	...

Les requêtes demandées dans cette partie sont à écrire en langage SQL.

- Écrire une requête permettant d'obtenir le nom de tous les matériaux qui ont une température de Curie strictement inférieure à 500 kelvins.

Un client potentiel souhaite acheter 4,5 kilogrammes de nickel (d'identifiant 8713, que l'on pourra utiliser directement dans les requêtes) et sélectionner le fournisseur le moins cher.

7. Écrire une requête permettant d'obtenir les noms de tous les fournisseurs proposant du nickel et le prix proposé par chacun pour 4,5 kilogrammes de nickel.
8. Modifier ou compléter la requête précédente afin d'obtenir le nom du fournisseur de nickel le moins cher ainsi que le prix à payer chez ce fournisseur pour ces 4,5 kilogrammes de nickel. En cas d'égalité du prix optimal entre plusieurs fournisseurs, on obtiendra les noms de tous les fournisseurs possibles.
9. Écrire une requête permettant d'obtenir le nom de tous les matériaux et le prix moyen pour un kilogramme de chacun de ces matériaux (la moyenne étant calculée pour tous les fournisseurs proposant ce matériau), en se limitant aux prix moyens strictement inférieurs à 50 euros par kilogramme.

Partie III : Modèle microscopique d'un matériau magnétique

Pour étudier l'effet du champ magnétique sur un matériau magnétique, on adopte une modélisation microscopique. On modélise les atomes par des sites portant chacun une grandeur physique, nommée *spin*, dont il n'est pas nécessaire de connaître les propriétés.

L'échantillon modélisé est une zone carrée à deux dimensions possédant h *spins* régulièrement répartis dans chaque direction, donc formant une grille carrée de $n = h^2$ *spins*. Chaque *spin* ne possède que deux états *down* ou *up*, ce que l'on modélise par une variable $s_i \in \{-1, +1\}$.

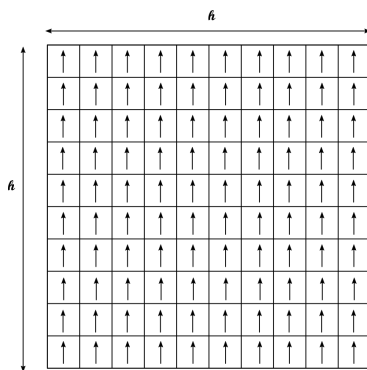


FIGURE 2 – Modèle des *spins* dans un matériau ferromagnétique

Pour implémenter cette configuration de *spins* décrivant l'état microscopique du matériau, on choisit de travailler sur une liste s , contenant n entiers, chacun valant -1 ou 1 . On notera le choix d'implémentation adoptée, qui impose de travailler sur **une simple liste de n éléments** pour modéliser une grille de taille $n = h \times h$, dans l'ordre suivant : première ligne puis deuxième ligne, etc.

Un domaine d'aimantation uniforme (cf. figure 2) sera donc représenté par une liste contenant n fois 1 ($[1, 1, 1, \dots, 1]$).

Le début du programme, outre les imports de module Python déjà réalisés à la question 1, est défini par :

```
1 | h = 100
2 | n = h**2
```

ce qui définit deux variables globales utilisables dans tout le programme.

10. Écrire une fonction `initialisation()` renvoyant une liste d'initialisation des domaines contenant n spins de valeur 1 comme sur la figure 2.

L'antiferromagnétisme est une propriété de certains milieux magnétiques. Contrairement aux matériaux ferromagnétiques, dans les matériaux antiferromagnétiques, l'interaction d'échange entre les atomes voisins conduit à un alignement antiparallèle des moments magnétiques atomiques (cf. figure 3). L'aimantation totale du matériau est alors nulle (on se limite au cas où h est pair).

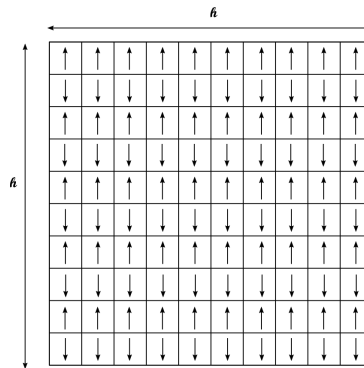


FIGURE 3 – Modèle des spins d'un matériau antiferromagnétique

11. Écrire une fonction `initialisation_anti()` renvoyant une liste `s` d'initialisation des domaines contenant h spins en largeur et h en hauteur en alternant les 1 et -1 comme sur la figure 3.
12. Pour afficher l'état global du matériau, il est nécessaire de convertir la liste `s` utilisée en un tableau de taille $h \times h$ représenté par une liste de listes. Écrire une fonction `repliement(s)` qui prend en argument la liste de spins `s` et qui renvoie une liste de h listes de taille h représentant le domaine.

Attention : dans la suite, l'utilisation de la fonction `repliement` n'est pas autorisée : on travaille exclusivement sur une liste unidimensionnelle `s`.

Dans la modélisation adoptée (sans champ magnétique extérieur), l'énergie d'une configuration, définie par l'ensemble des valeurs de tous les spins, est donnée par :

$$E = -\frac{J}{2} \sum_i \sum_{j \in V_i} s_i s_j \quad (3)$$

avec V_i l'ensemble des voisins du *spin* i .

On suppose que seuls les quatre *spins* situés juste au dessus, en dessous, à gauche et à droite de s_i sont capables d'interagir avec lui.

J est nommée **intégrale d'échange** et modélise l'interaction entre deux *spins* voisins. Pour simplifier, on considérera dans les programmes que $J = 1$.

Malgré le caractère fini de l'échantillon, on peut utiliser une modélisation très utile pour faire comme s'il était infini en utilisant les *conditions aux limites périodiques*. Lorsque l'on considère un *spin* dans la colonne située la plus à droite (*resp.* gauche), il ne possède pas de plus proche voisin à droite (*resp.* gauche) : on convient de lui en affecter un, qui sera situé sur la même ligne complètement à gauche (*resp.* droite) de l'échantillon. De même, le plus proche voisin manquant d'un *spin* situé sur la première (*resp.* dernière) ligne sera situé sur la dernière (*resp.* première) ligne de l'échantillon (voir la figure 4).

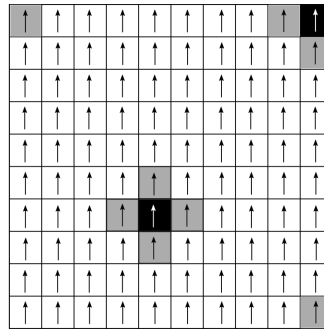


FIGURE 4 – Voisinage d'un *spin* (les voisins des *spins* sur les cases noires sont indiqués en gris)

13. Définir une fonction `liste_voisins(i)` qui renvoie la liste des indices des plus proches voisins du *spin* s_i d'indice i dans la liste $\mathbf{s}$ (dans l'ordre gauche, droite, dessous, dessus). On pourra utilement utiliser les opérations `%` et `//` de Python, qui renvoient le reste et le quotient de la division euclidienne.
14. Définir la fonction `energie(s)` qui calcule l'énergie d'une configuration $\mathbf{s}$ donnée (cf. équation 3).

Pour trouver une configuration stable pour les spins, il faut faire évoluer la liste vers une situation d'équilibre conformément aux principes de la physique statistique. On adopte une méthode probabiliste connue sous le nom de méthode de Monte-Carlo, dont le principe de fonctionnement est le suivant. À chaque étape :

- on choisit un *spin* au hasard dans l'échantillon,
- on calcule la variation d'énergie ΔE qui résulterait d'un changement d'orientation de ce *spin*,
- si $\Delta E \leq 0$, ce *spin* change de signe,
- si $\Delta E > 0$, ce *spin* change de signe avec la probabilité donnée par la loi de Boltzmann :

$$p = \exp\left(\frac{-\Delta E}{k_B T}\right).$$

Dans la suite, ΔE et $k_B T$ seront désignées par les variables `delta_e` et `T` correspondantes dans le programme.

15. Définir une fonction `test_boltzmann(delta_e, T)` qui renvoie `True` si le *spin* change de signe, et `False` sinon.
16. Juste après avoir sélectionné au hasard l'indice `i` d'un *spin* de la liste `s` à basculer éventuellement, pour évaluer l'écart d'énergie `delta_e` entre les deux configurations avant/après, on propose deux solutions sous forme des fonctions `calcul_delta_e1` et `calcul_delta_e2` :

```

1 def calcul_delta_e1(s, i):
2     s2 = s[:]
3     s2[i] = -s[i]
4     delta_e = energie(s2)-energie(s)
5     return delta_e

1 def calcul_delta_e2(s, i):
2     delta_e = 0
3     for j in liste_voisins(i):
4         delta_e = delta_e + 2*s[i]*s[j]
5     return delta_e

```

où `s[i]` est le *spin* choisi pour être éventuellement retourné. Indiquer la solution qui vous paraît la plus efficace pour minimiser le temps de calcul en justifiant votre réponse.

17. En utilisant la fonction `test_boltzmann`, définir une fonction `monte_carlo(s, T, n_tests)` qui applique la méthode de Monte-Carlo et qui modifie la liste `s` où l'on a choisi successivement `n_test spins` au hasard, que l'on modifie éventuellement suivant les règles indiquées dans les explications.
18. Écrire la fonction `aimantation_moyenne(n_tests, T)` qui :
 - initialise une liste des *spins* (avec la fonction `initialisation` par exemple),
 - la fait évoluer en effectuant `n_tests` tests de Boltzmann et les inversions éventuelles qui en découlent,
 - calcule et renvoie l'aimantation moyenne de la configuration à la température `T` (définie ici comme la somme des valeurs des *spins* divisée par le nombre total de *spins*).
19. Évaluer la complexité asymptotique de la fonction `aimantation_moyenne(n_tests, T)` en fonction de `n`, nombre de *spins* dans le système, et de `n_tests`.

20. Cette complexité asymptotique serait-elle modifiée si on avait voulu prendre en compte toutes les interactions entre deux *spins* quelconques dans le système, et plus seulement entre les plus proches voisins ? Justifier.

On réalise plusieurs simulations en faisant varier la température T autour de la température de Curie (ici $T_C=2,269$ compte-tenu du choix des valeurs de J et k_B). On représente alors les *spins* orientés vers le haut par une case foncée et les *spins* orientés vers le bas par une case claire. On obtient les résultats de la figure 5.

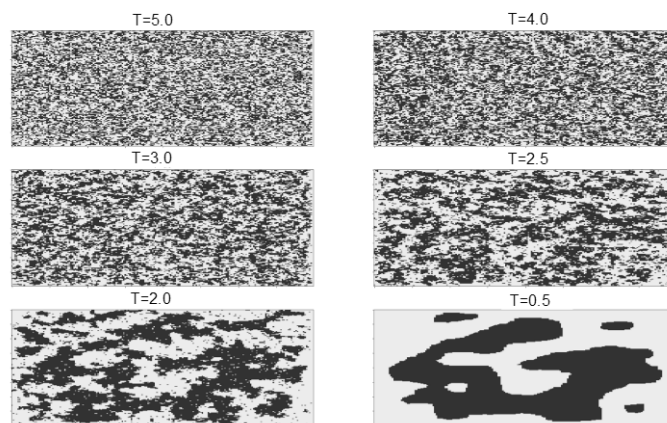


FIGURE 5 – Évolution du domaine en fonction de la température T

21. Indiquer l'influence de l'augmentation de la température sur le comportement du matériau ferromagnétique.

Partie IV : Exploration des domaines de Weiss

Une observation microscopique des matériaux magnétiques nous apprend que les zones magnétiques du matériau sont organisées en domaines, nommés domaines de Weiss. Dans un domaine de Weiss donné, tous les *spins* ont la même valeur.

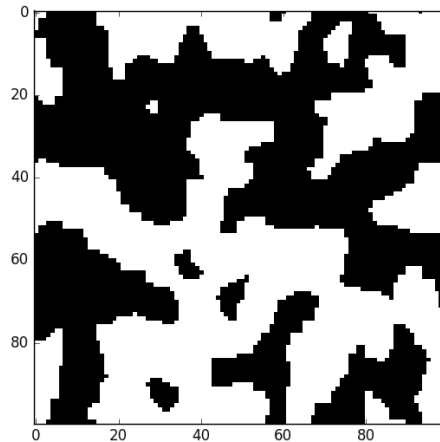


FIGURE 6 – Représentation des domaines de Weiss

On souhaite dans la suite décrire les différents domaines de Weiss afin, par exemple, de les colorier d'une manière différente. Pour cela, on va construire une liste, nommée `weiss`, possédant exactement la même taille que `s` (soit `n`) contenant initialement des `-1` (cette valeur signifiant que le *spin* correspondant n'a encore été affecté à aucun domaine de Weiss).

On souhaite alors écrire une fonction récursive `explorer_voisinage(s, i, weiss, num)` qui, à partir d'une configuration donnée `s`, d'un indice de départ `i` (repérant le *spin* dans `s`), ainsi qu'un entier `num` qui précise le numéro du domaine auquel appartient s_i , construit récursivement la liste `weiss` par effet de bord. Cette fonction doit réaliser les opérations suivantes :

- Pour chaque *spin* voisin du *spin* s_i , elle doit vérifier si les *spins* sont identiques, et s'il n'a pas déjà été affecté à un domaine de Weiss précédemment.
- Dès qu'un tel *spin* est ajouté, on inscrit son numéro de domaine dans la liste `weiss`, et on explore récursivement son voisinage.

22. Écrire le code de la fonction récursive `explorer_voisinage(s, i, weiss, num)` conforme à la description ci-dessus.

Avec la fonction précédente, la pile de récursion peut devenir de taille très importante dans le cas d'un domaine de grande taille. Afin de mieux contrôler ce parcours, on choisit de l'effectuer avec une structure de pile explicite. Cette pile sera représentée par une liste nommée `pile` sur laquelle on peut ajouter un élément (avec `append`) ou récupérer l'élément du dessus (via `pile.pop()` qui renvoie l'élément sur le dessus de la pile et le retire de la pile). Il s'agit donc, tant qu'il reste des *spins* à explorer, de :

- récupérer l'indice d'un *spin* à explorer dans la pile et le marquer dans la liste `weiss`,
- regarder dans son voisinage si des *spins* possèdent la même valeur et n'ont pas encore été affectés à un domaine, puis ajouter leurs indices dans la pile si c'est le cas.

23. Écrire le code de la fonction itérative `explorer_voisinage_pile(s, i, weiss, num, pile)` conforme à la description ci-dessus.

Enfin, la fonction précédente va permettre de construire la liste `weiss` contenant les numéros des domaines auxquels appartiennent tous les *spins* (le numéro du domaine auquel appartient le *spin* d'indice 0 sera pris à 0, le domaine suivant à 1, etc.).

24. Écrire le code de la fonction `weiss=construire_domaines_weiss(s)` qui construit et renvoie la liste `weiss` contenant le numéro des domaines de Weiss de chaque *spin* du domaine.

L'intérêt de la construction précédente est de disposer d'un marqueur différent pour chaque domaine de Weiss, permettant par exemple de les visualiser dans deux dimensions avec des couleurs différentes, comme dans l'image de la figure 7 :

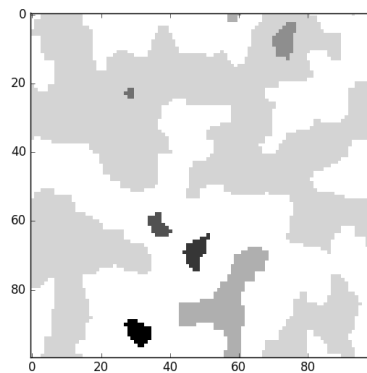


FIGURE 7 – Domaines de Weiss après marquage en nuances de gris

Fin de l'épreuve

Annexe : Documentation sommaire

Les fonctions présentées ci-dessous pourront être utilisées sous réserve de l'import du module Python auquel elles appartiennent.

- Module `math` : les fonctions `exp` et `tanh` permettent de calculer l'exponentielle et la tangente hyperbolique d'un entier ou d'un flottant.
- Module `random` :
 - `randrange(n)` permet de renvoyer un entier aléatoirement choisi parmi $0, 1, 2, \dots, n - 1$
 - `random()` permet de renvoyer un flottant aléatoire entre 0 et 1 suivant une densité de probabilité uniforme.

On admet que ces deux fonctions sont de complexité constante.

SOLUTION

Q1.

```
from math import exp, tanh
from random import randrange, random
```

Commentaires : Question élémentaire sans difficulté.

Q2. L'équation recherchée est $x - \tanh\left(\frac{x}{t}\right) = 0$.

```
def f(x,t):
    return x-tanh(x/t)
```

Commentaires : Question élémentaire sans difficulté.

Q3.

```
def dichotomie(f,t,a,b,eps):
    g,d , m = a,b,(a+b)/2
    while d-g > eps:
        if f(g,t)*f(m,t)<0:
            d=m
            m=(g+d)/2
        else:
            g=m
            m=(g+d)/2
    return m
```

Commentaires : Si cette question est formellement une question de cours, elle s'avère discriminante car un nombre significatif de candidats ne maîtrise pas suffisamment l'algorithme de dichotomie. Par exemple, ils confondent les quantités $\frac{a+b}{2}$ et $\frac{b-a}{2}$ qui n'ont pas du tout la même signification !

Q4. La boucle *while* s'arrête quand

$$\frac{b-a}{2^n} \leq \varepsilon \Leftrightarrow 2^n \geq \frac{b-a}{\varepsilon} \Leftrightarrow n \geq \log_2 \left(\frac{b-a}{\varepsilon} \right)$$

donc la complexité recherchée est en $O\left(\log_2 \left(\frac{b-a}{\varepsilon} \right)\right)$ (les opérations au sein et en dehors de la boucle *while* ayant un coût constant et en nombre fini, fixé)

Commentaires : Question sélective. Un nombre restreint de candidats connaissent la réponse. Pour les autres, une part importante d'entre eux ne parviennent pas à résoudre correctement l'inéquation $\frac{b-a}{2^n} \leq \varepsilon$ (mauvaise gestion des logarithmes) ou ne voit pas le processus de diviser pour régner. Dommage.

Q5.

```
def construction_liste_m(t1,t2):
    l=[]
    pas = (t2-t1)/499
    t=t1
    for i in range(500):
        if t >= 1:
            l.append(0)
        else:
            m=dicho(f,t,0.001,1,10**(-6))
            l.append(m)
        t=t+pas
    return l
```

Commentaires : Question sans difficulté informatique particulière. Un nombre important de candidats ne choisissent pas le bon pas pour obtenir 500 points. Ils divisent la longueur de l'intervalle par 500 de lieu de le diviser par 499. Le diable est dans les détails :-).

Q6.

```
SELECT nom FROM materiaux where t_curie<=500
```

Commentaires : Question élémentaire sans difficulté.

Q7. Version pragmatique en lisant le tableau

```
SELECT id_fournisseur, nom_fournisseur, prix_kg*4.5
FROM fournisseurs JOIN prix
ON prix.id_four = fournisseurs.id_fournisseur
WHERE prix.id_mat=8713
```

Version plus générale en ne connaissant pas le numéro identifiant le nickel

```
SELECT nom_fournisseur, prix_kg*4.5
FROM fournisseurs JOIN prix
ON prix.id_four = fournisseurs.id_fournisseur
WHERE prix.id_mat = (SELECT id_materiau FROM materiaux WHERE
    nom = "nickel")
```

Commentaires : Question discriminante mais qui ne doit pas poser trop de problème aux candidats ayant une compréhension correcte des jointures (pour la version pragmatique). Ce type de question est récurrente et de difficulté très raisonnable donc à travailler si elle vous pose problème. Pensez à multiplier les prix par 4.5.

Q8.

```
SELECT nom_fournisseur , prix_kg*4.5
FROM fournisseurs JOIN prix
ON prix.id_four = fournisseurs.id_fournisseur
WHERE prix.id_mat = (SELECT id_materiau FROM materiaux WHERE
    nom = "nickel") ORDER BY prix_kg*4.5 limit 1
```

Commentaires : Question demandant une bonne connaissance par le candidat du SQL du programme. Les candidats ayant tenté d'utiliser la fonction MIN ont souvent commis des erreurs de syntaxe. Question très discriminante à l'intention des bonnes copies.

Q9.

```
SELECT materiaux.nom, AVG(prix.prix_kg) as moy
FROM prix JOIN materiaux
ON materiaux.id_materiau = prix.id_mat
GROUP BY prix.id_mat
HAVING moy <= 50
```

Commentaires : Question demandant une bonne connaissance par le candidat du SQL du programme. Ne pas confondre la commande WHERE (condition vérifiée au préalable) avec la commande HAVING condition vérifiée à postériori, ce qui était indispensable ici. La gestion de la commande GROUP BY fut problématique pour un nombre significatif de candidats. Au final, la question fut à l'intention des bonnes copies.

Q10.

```
def initialisation():
    return [1 for i in range(n)]
```

Commentaires : Question élémentaire sans aucune difficulté.

Q11.

```
def initialisation_anti():
    temp=[]
    e=1
    c=0
    for i in range(n):
        if c==h:
            e=-e
            c=0
        c=c+1
        temp.append(e)
    return temp
```

Commentaires : Question discriminante car un nombre important de candidats eurent beaucoup de mal à faire l'alternance des 1 et des -1. Certains ont fait une moitié de 1 puis une moitié de -1 (1, ..., 1, -1, ..., -1) d'autres ont alterné consécutivement les 1 et les -1 (1, -1, 1, -1, ..., 1, -1).

Q12.

```
def repliement(s):
    resultat=[]
    temp=[]
    for e in s:
        temp.append(e)
        if len(temp) == h:
            resultat.append(temp)
            temp=[]
    return resultat
```

Commentaires : Question très discriminante. Un nombre important ont du mal à gérer des listes de listes ou bien tente de lier la position dans la liste initiale avec les coordonnées (i, j) dans la liste de liste (interprétée comme une matrice mathématique) sans être en mesure de donner les bonnes coordonnées (la bonne réponse étant, si on fixe un compteur c codant la position dans la liste initiale, $i = c // h$ et $j = c \% h$). Le plus simple étant de changer de liste tous les h éléments comme dans le corrigé ci-dessus.

Q13.

```
def liste_voisins(i):
    return [(i+h)%n, (i-h)%n, (i+1)%n, (i-1)%n]
```

Commentaires : Question discriminante car il fallait penser à distinguer le cas général (point en dehors des bords du carré) où les voisins sont faciles à définir (on avance ou recule d'une unité ou de h unités) des cas particuliers (les bords et les coins du carré). Dans ce cas, il est indispensable d'être rigoureux pour lister toutes les configurations. Peu de candidats furent en mesure d'utiliser simplement la notion de modulo pour avoir une formule juste, simple et compact.

Q14.

```
def energie(s):
    E=0
    for i in range(len(s)):
        for j in liste_voisin(i):
            E=E+s[i]*s[j]
    return -E/2
```

Commentaires : Question discriminante car il fallait faire la distinction entre position dans la liste (indice) et valeur du spin en cette position (valeur). Un nombre conséquent de candidats n'y parvinrent pas.

Q15.

```
def test_boltzmann(delta_e, T):
    if delta_e <=0:
        return True
    else:
        if random <= exp(-delta_e/T):
            return True
        else:
            return False
```

ou sous forme plus compact en utilisant les booléens

```
def test_boltzmann(delta_e, T):
    return (delta_e <=0) or (random <= exp(-delta_e/T))
```

Commentaires : *Un nombre important de candidats ne réussissent pas à gérer la gestion probabiliste de changement de signe. Si vous n'y êtes pas parvenu, il faut s'entraîner car ce type de situation est courant dans de nombreuses situations informatiques (choix gérer par une sélection au hasard).*

Q16. La première fonction refait l'intégralité des calculs d'énergie sur tous les spins (complexité en $O(n^2)$ car n itérations de la boucle et, pour chaque itération, il faut itérer sur tous les voisins dont le nombre maximal est n) alors que la seconde ne recalcule que la compensation nécessaire pour analyser les spins voisins du spin s_i (car tous les spins s_j voisins de s_i changent de signe donc au lieu d'enlever $s_i s_j$, il faut l'ajouter, le facteur 2 étant là pour effacer l'enlèvement et faire l'ajout) donc la complexité est en grand O du nombre de voisins de s_i donc en $O(n)$ et non en $O(n^2)$.

Commentaires : *Question discriminante qui demande au préalable un questionnaire du candidat sur la différence majeure entre les deux codes (l'un traite tous les spins, l'autre juste les spins voisins). Ensuite, il est attendu une justification, sous entendu en terme de complexité asymptotique et non juste en terme de " plus petit que ". Ce type de question demande un recul certain du candidat tant du point de vue informatique que du point de vue scientifique. Ces questions sont rémunératrices.*

Q17.

```
def monte_carlo(s, T, n_tests):
    for i in range(n_tests):
        i0 = randrange(n) # choix aléatoire d'une position
        delta_e=calcul_delta_e2(s,i0)
        if test_boltzmann(delta_e,T):
            s[i0]=-s[i0] # changement du spin en cette
                        position
```

Commentaires : *Question destinée aux bonnes copies ayant réussi plusieurs des questions précédentes.*

Q18.

```
def aimantation_moyenne(n_tests, T):
    s=initialisation()
    monte_carlo(s,T,n_tests)
    a=0
    for e in s:
        a=a+e
    return a/n
```

Commentaires : Question élémentaire sans difficulté particulière.

Q19. *initialisation()* a une complexité en $O(n)$, *monte_carlo*(T, n_test) a une complexité en $O(n\_test)$ (car le nombre de voisins est borné, ici par 4), le calcul de l'aimantation moyenne a une complexité totale en $O(n)$ donc la complexité totale de la fonction *aimantation_moyenne* est en

$$O(n) + O(n\_test) + O(n) = O(n + n\_test).$$

Commentaires : Question sélective car, si beaucoup de candidats ont vu deux boucles, l'une s'itérant n fois, l'autre n_test fois, beaucoup conclure à une complexité en $O(n \times n\_test)$! Ces candidats n'ont guère réfléchi et ont sorti une réponse automatique alors que les deux boucles sont séparées (et non imbriquées) !

Q20. En prenant toutes les interactions, le nombre de voisins devient alors n et non 4 (au maximum) donc *monte_carlo*(T, n_test) a une complexité en $O(n * n\_test)$, les autres fonctions conservant la même complexité. Au total, la fonction *aimantation_moyenne* est en

$$O(n) + O(n * n\_test) + O(n) = O(n * n\_test)$$

ce qui accroît significativement la complexité (si $h = 10^3$ alors $n = 10^6$ donc on multiplie par un million le temps de calcul !)

Commentaires : Question bien réussie ... uniquement par ceux ayant répondu correctement à la question précédente :-).

Q21. Lorsque la température diminue, le nombre de domaines de Weiss diminue et la taille de chacun d'eux augmentent significativement. Autrement dit, l'abaissement de la température entraîne une augmentation importante de la synchronisation des spins à grande échelle.

Commentaires : Comme souvent pour les questions d'interprétations, la question s'est avérée très sélective. Elle demande du recul scientifique de la part du candidat et une analyse contextualisée de la situation. Au final, peu de réponses sensées.

Q22.

```
def explorer_voisinage(s, i, weiss, num):
    weiss[i]=num
    voisins=liste_voisins(i)
    for j in voisins:
        if s[i]==s[j] and weiss[j] <0:
```

```

weiss[j]=num
explorer_voisinage(s,j,weiss,num)

```

Commentaires : *Question sélective à destination des meilleures copies ayant une maîtrise certaine de l'informatique (pile, parcours de graphe).*

Q23.

```

def explorer_voisinage_pile(s, i, weiss, num, pile):
    pile.append(i)
    while pile != []:
        k = pile.pop()
        weiss[k]=num
        voisins=liste_voisins(k)
        for j in voisins:
            if s[k] == s[j] and weiss[j] <0:
                weiss[j]=num
                pile.append(j)

```

Commentaires : *Même remarque que précédemment.*

Q24.

```

# fonction testant s'il reste des spins non marqués.
# Si c'est le cas, il renvoie l'indice d'un tel spin, sinon il
# renvoie -1
# signifiant que tous les spins sont marqués

def indice_restant(weiss):
    for i in range(n):
        if weiss[i]<0:
            return i
    return -1

def construire_domaines_weiss(s):
    weiss=[-1 for i in range(n)]
    indice = indice_restant(weiss)
    num = 0
    while indice >=0:
        explorer_voisinage_pile(s,indice,weiss,num,[])
        indice = indice_restant(weiss)
        num = num +1
    return weiss

```

Si on veut tester les codes sur un exemple

```

import numpy as np
h=100
n=h**2

```

```
s=initialisation_anti()  
monte_carlo(s, 10,1000)  
W=construire_domaines_weiss(s)  
M=repliement(W)  
plt.imshow(M)  
plt.show()
```

Commentaires : *Même remarque que précédemment.*

A2022 – MATH I MP



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2022

PREMIÈRE ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 3 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES I - MP

L'énoncé de cette épreuve comporte 7 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Formule asymptotique de Hardy et Ramanujan

L'objectif de ce problème est l'étude asymptotique du nombre de partitions d'un entier naturel n , c'est-à-dire du nombre de décompositions de n en somme d'entiers naturels non nuls (sans tenir compte de l'ordre des termes). Une définition rigoureuse de ce nombre, noté p_n , est donnée en début de partie **B**. Dans la partie **A**, on introduit une fonction P de variable complexe ; dans la fin de la partie **B** on démontre qu'il s'agit de la somme, sur le disque unité ouvert de $\mathbf{C}$, de la série entière $\sum_{n \geq 0} p_n z^n$. L'étude de P au voisinage de 1 permet alors, dans les parties suivantes, de progresser vers l'obtention d'un équivalent simple de la suite $(p_n)_{n \in \mathbf{N}}$ (formule asymptotique de Hardy et Ramanujan).

Tout au long du problème, le disque unité ouvert de $\mathbf{C}$ sera noté

$$D = \{z \in \mathbf{C} : |z| < 1\}.$$

Dans tout l'énoncé, on utilisera la dénomination « variable aléatoire réelle » pour signifier « variable aléatoire discrète réelle ».

On admettra aussi les deux identités classiques suivantes :

$$\sum_{n=1}^{+\infty} \frac{1}{n^2} = \frac{\pi^2}{6} \quad \text{et} \quad \int_{\mathbf{R}} e^{-\frac{u^2}{2}} du = \sqrt{2\pi}.$$

A. Fonctions L et P

- 1 ▷ Soit $z \in D$. Montrer la convergence de la série $\sum_{n \geq 1} \frac{z^n}{n}$. Préciser la valeur de sa somme lorsque $z \in]-1, 1[$. On notera

$$L(z) := \sum_{n=1}^{+\infty} \frac{z^n}{n}.$$

- 2 ▷ Soit $z \in D$. Montrer que la fonction $t \in [0, 1] \mapsto L(tz)$ est dérivable et donner une expression simple de sa dérivée. En déduire que $t \mapsto (1 - tz)e^{L(tz)}$ est constante sur $[0, 1]$ et conclure que

$$\exp(L(z)) = \frac{1}{1 - z}.$$

- 3 ▷ Montrer que $|L(z)| \leq -\ln(1 - |z|)$ pour tout z dans D . En déduire la convergence de la série $\sum_{n \geq 1} L(z^n)$ pour tout z dans D . Dans la suite, on notera, pour z dans D ,

$$P(z) := \exp \left[\sum_{n=1}^{+\infty} L(z^n) \right].$$

On remarque, en vertu de la question précédente et des propriétés de l'exponentielle, que

$$\forall z \in D, P(z) \neq 0 \quad \text{et} \quad P(z) = \lim_{N \rightarrow +\infty} \prod_{n=1}^N \frac{1}{1 - z^n}.$$

B. Développement de P en série entière

Pour $(n, N) \in \mathbf{N} \times \mathbf{N}^*$, on note $P_{n,N}$ l'ensemble des listes $(a_1, \dots, a_N) \in \mathbf{N}^N$ telles que $\sum_{k=1}^N k a_k = n$. Si cet ensemble est fini, on note $p_{n,N}$ son cardinal.

4 ▷ Soit $n \in \mathbf{N}$. Montrer que $P_{n,N}$ est fini pour tout $N \in \mathbf{N}^*$, que la suite $(p_{n,N})_{N \geq 1}$ est croissante et qu'elle est constante à partir du rang $\max(n, 1)$.

Dans toute la suite, on notera p_n la valeur finale de $(p_{n,N})_{N \geq 1}$.

5 ▷ Montrer par récurrence que

$$\forall N \in \mathbf{N}^*, \forall z \in D, \prod_{k=1}^N \frac{1}{1-z^k} = \sum_{n=0}^{+\infty} p_{n,N} z^n.$$

6 ▷ Soit $z \in D$. On convient que $p_{n,0} = 0$ pour tout $n \in \mathbf{N}$. En examinant la sommabilité de la famille $((p_{n,N+1} - p_{n,N})z^n)_{(n,N) \in \mathbf{N}^2}$, démontrer que

$$P(z) = \sum_{n=0}^{+\infty} p_n z^n.$$

En déduire le rayon de convergence de la série entière $\sum_n p_n x^n$.

7 ▷ Soit $n \in \mathbf{N}$. Montrer que pour tout réel $t > 0$,

$$p_n = \frac{e^{nt}}{2\pi} \int_{-\pi}^{\pi} e^{-in\theta} P(e^{-t+i\theta}) d\theta,$$

si bien que

$$p_n = \frac{e^{nt} P(e^{-t})}{2\pi} \int_{-\pi}^{\pi} e^{-in\theta} \frac{P(e^{-t+i\theta})}{P(e^{-t})} d\theta. \quad (1)$$

Dans le reste du problème, l'objectif est d'obtenir un équivalent du nombre p_n lorsque n tend vers $+\infty$. Cet équivalent sera obtenu via un choix approprié de t en fonction de n dans la formule (1).

C. Contrôle de P

8 ▷ Soit $x \in [0, 1[$ et $\theta \in \mathbf{R}$. En utilisant la fonction L , montrer que

$$\left| \frac{1-x}{1-xe^{i\theta}} \right| \leq \exp(-(1-\cos\theta)x).$$

En déduire que pour tout $x \in [0, 1[$ et tout réel θ ,

$$\left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp\left(-\frac{1}{1-x} + \operatorname{Re}\left(\frac{1}{1-xe^{i\theta}}\right)\right).$$

9 ▷ Soit $x \in [\frac{1}{2}, 1[$ et $\theta \in \mathbf{R}$. Montrer que

$$\frac{1}{1-x} - \operatorname{Re}\left(\frac{1}{1-xe^{i\theta}}\right) \geq \frac{x(1-\cos\theta)}{(1-x)((1-x)^2 + 2x(1-\cos\theta))}.$$

En déduire que

$$\left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp\left(-\frac{1-\cos\theta}{6(1-x)^3}\right) \quad \text{ou que} \quad \left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp\left(-\frac{1}{3(1-x)}\right).$$

D. Intermède : quelques estimations de sommes

On fixe dans cette partie un réel $\alpha > 0$ et un entier $n \geq 1$. Sous réserve d'existence, on pose

$$S_{n,\alpha}(t) := \sum_{k=1}^{+\infty} \frac{k^n e^{-kt\alpha}}{(1-e^{-kt})^n}.$$

On introduit aussi la fonction

$$\varphi_{n,\alpha} : x \in \mathbf{R}_+^* \mapsto \frac{x^n e^{-\alpha x}}{(1-e^{-x})^n},$$

qui est évidemment de classe $\mathcal{C}^\infty$.

10 ▷ Montrer que $\varphi_{n,\alpha}$ et $\varphi'_{n,\alpha}$ sont intégrables sur $]0, +\infty[$.

11 ▷ Montrer, pour tout réel $t > 0$, l'existence de $S_{n,\alpha}(t)$, sa positivité stricte, et l'identité

$$\int_0^{+\infty} \varphi_{n,\alpha}(x) dx = t^{n+1} S_{n,\alpha}(t) - \sum_{k=0}^{+\infty} \int_{kt}^{(k+1)t} (x-kt) \varphi'_{n,\alpha}(x) dx.$$

En déduire que

$$S_{n,\alpha}(t) = \frac{1}{t^{n+1}} \int_0^{+\infty} \frac{x^n e^{-\alpha x}}{(1-e^{-x})^n} dx + O\left(\frac{1}{t^n}\right) \quad \text{quand } t \rightarrow 0^+.$$

12 ▷ Démontrer, sans utiliser ce qui précède, que

$$\int_0^{+\infty} \frac{xe^{-x}}{1-e^{-x}} dx = \frac{\pi^2}{6}.$$

Dans le reste du problème, nous admettrons le résultat suivant (il peut être démontré par une méthode similaire) :

$$\int_0^{+\infty} \frac{x^2 e^{-x}}{(1-e^{-x})^2} dx = \frac{\pi^2}{3}.$$

E. Contrôle des fonctions caractéristiques

Étant donné une variable aléatoire réelle X sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbf{P})$, ainsi qu'un réel θ , les variables aléatoires réelles $\cos(\theta X)$ et $\sin(\theta X)$ sont d'espérance finie puisque bornées : on introduit alors le nombre complexe

$$\Phi_X(\theta) := \mathbf{E}(\cos(\theta X)) + i \mathbf{E}(\sin(\theta X)).$$

13 ▷ Soit X une variable aléatoire réelle. Montrer que $|\Phi_X(\theta)| \leq 1$ pour tout réel θ .

Dans les questions 14 ▷ à 18 ▷, on se donne une variable aléatoire réelle X suivant une loi géométrique, de paramètre $p \in]0, 1[$ arbitraire. On pose $q = 1 - p$.

14 ▷ Montrer que pour tout $(a, b) \in \mathbf{R}^2$ et tout réel θ ,

$$\Phi_{aX+b}(\theta) = \frac{p e^{i(a+b)\theta}}{1 - q e^{ia\theta}}.$$

15 ▷ Montrer que pour tout $k \in \mathbf{N}$, la variable aléatoire X^k est d'espérance finie. Montrer que Φ_X est de classe $\mathcal{C}^\infty$ sur $\mathbf{R}$ et que $\Phi_X^{(k)}(0) = i^k \mathbf{E}(X^k)$ pour tout $k \in \mathbf{N}$.

16 ▷ Montrer qu'il existe une suite $(P_k)_{k \in \mathbf{N}}$ de polynômes à coefficients dans $\mathbf{C}$, indépendante de p , telle que

$$\forall \theta \in \mathbf{R}, \forall k \in \mathbf{N}, \Phi_X^{(k)}(\theta) = p i^k e^{i\theta} \frac{P_k(q e^{i\theta})}{(1 - q e^{i\theta})^{k+1}} \quad \text{et} \quad P_k(0) = 1.$$

- 17 ▷ En déduire qu'il existe une suite $(C_k)_{k \in \mathbf{N}}$ de réels strictement positifs, indépendante de p , telle que

$$\forall k \in \mathbf{N}, \left| \mathbf{E}(X^k) - \frac{1}{p^k} \right| \leq \frac{C_k q}{p^k}.$$

- 18 ▷ En déduire qu'il existe un réel $K > 0$ indépendant de p tel que

$$\mathbf{E}((X - \mathbf{E}(X))^4) \leq \frac{K q}{p^4}.$$

Dans les questions 19 ▷ à 21 ▷, on se donne une variable aléatoire réelle centrée Y telle que Y^4 soit d'espérance finie.

- 19 ▷ Montrer successivement que Y^2 et $|Y|^3$ sont d'espérance finie, et que

$$\mathbf{E}(Y^2) \leq (\mathbf{E}(Y^4))^{1/2} \quad \text{puis} \quad \mathbf{E}(|Y|^3) \leq (\mathbf{E}(Y^4))^{3/4}.$$

- 20 ▷ Montrer, pour tout réel u , l'inégalité

$$\left| e^{iu} - 1 - iu + \frac{u^2}{2} \right| \leq \frac{|u|^3}{6}.$$

En déduire que pour tout réel θ ,

$$\left| \Phi_Y(\theta) - 1 + \frac{\mathbf{E}(Y^2) \theta^2}{2} \right| \leq \frac{|\theta|^3}{3} (\mathbf{E}(Y^4))^{3/4}.$$

- 21 ▷ Conclure que pour tout réel θ ,

$$\left| \Phi_Y(\theta) - \exp\left(-\frac{\mathbf{E}(Y^2) \theta^2}{2}\right) \right| \leq \frac{|\theta|^3}{3} (\mathbf{E}(Y^4))^{3/4} + \frac{\theta^4}{8} \mathbf{E}(Y^4).$$

F. Convergence vers une gaussienne

Étant donné un réel $t > 0$, on pose, suivant les notations de la partie C,

$$m_t := S_{1,1}(t) \quad \text{et} \quad \sigma_t := \sqrt{S_{2,1}(t)}.$$

Étant donné des réels $t > 0$ et θ , on pose

$$h(t, \theta) = e^{-im_t \theta} \frac{P(e^{-t} e^{i\theta})}{P(e^{-t})}.$$

Étant donné des réels $t > 0$ et u , on pose

$$\zeta(t, u) = \exp\left(i \frac{u}{\sigma_t} \left(m_t - \frac{\pi^2}{6t^2}\right)\right) \quad \text{et} \quad j(t, u) = \zeta(t, u) h\left(t, \frac{u}{\sigma_t}\right).$$

22 ▷ Soit $n \in \mathbf{N}^*$ ainsi que des complexes $z_1, \dots, z_n, u_1, \dots, u_n$ tous de module inférieur ou égal à 1. Montrer que

$$\left| \prod_{k=1}^n z_k - \prod_{k=1}^n u_k \right| \leq \sum_{k=1}^n |z_k - u_k|.$$

23 ▷ Soit $\theta \in \mathbf{R}$ et $t \in \mathbf{R}_+^*$. On considère, pour tout $k \in \mathbf{N}^*$, une variable aléatoire Z_k suivant la loi $\mathcal{G}(1 - e^{-kt})$, et on pose $Y_k = k(Z_k - \mathbf{E}(Z_k))$. Démontrer que

$$h(t, \theta) = \lim_{n \rightarrow +\infty} \prod_{k=1}^n \Phi_{Y_k}(\theta).$$

En déduire, à l'aide en particulier de la question 21 ▷, l'inégalité

$$\left| h(t, \theta) - e^{-\frac{\sigma_t^2 \theta^2}{2}} \right| \leq K^{3/4} |\theta|^3 S_{3,3/4}(t) + K \theta^4 S_{4,1}(t). \quad (2)$$

On rappelle que la constante K a été introduite à la question 18 ▷, les quantités $S_{n,\alpha}(t)$ dans la partie D.

24 ▷ Montrer que $\sigma_t \sim \frac{\pi}{\sqrt{3}t^{3/2}}$ quand t tend vers 0^+ . En déduire, pour tout réel u , que

$$j(t, u) \xrightarrow[t \rightarrow 0^+]{} e^{-u^2/2}.$$

25 ▷ Montrer qu'il existe un réel $\alpha > 0$ tel que

$$\forall \theta \in [-\pi, \pi], \quad 1 - \cos \theta \geq \alpha \theta^2.$$

À l'aide de la question 9 ▷, en déduire qu'il existe trois réels $t_0 > 0$, $\beta > 0$ et $\gamma > 0$ tels que, pour tout $t \in]0, t_0]$ et tout $\theta \in [-\pi, \pi]$,

$$|h(t, \theta)| \leq e^{-\beta(\sigma_t \theta)^2} \quad \text{ou} \quad |h(t, \theta)| \leq e^{-\gamma(\sigma_t |\theta|)^{2/3}}.$$

26 ▷ Conclure que

$$\int_{-\pi\sigma_t}^{\pi\sigma_t} j(t, u) \, du \xrightarrow[t \rightarrow 0^+]{} \sqrt{2\pi}.$$

G. La conclusion

Dans cette dernière partie, on admet que $P(e^{-t}) \sim \sqrt{\frac{t}{2\pi}} \exp\left(\frac{\pi^2}{6t}\right)$ quand t tend vers 0^+ .

27 ▷ En appliquant la formule (1) à $t = \frac{\pi}{\sqrt{6n}}$, démontrer que

$$p_n \sim \frac{\exp\left(\pi\sqrt{\frac{2n}{3}}\right)}{4\sqrt{3}n} \quad \text{quand } n \rightarrow +\infty,$$

formule découverte par Hardy et Ramanujan en 1918.

FIN DU PROBLÈME

SOLUTION

A. Fonctions L et P

1. Si $|z| < 1$ alors la suite (z^n/n) est bornée et si $|z| > 1$, elle est non bornée. Par lemme d'Abel, la série entière associée est de rayon de convergence égal à 1. En particulier, sa somme L est bien définie sur D et la série dérivée est de même rayon de convergence. De plus (dans le cas réel, la dérivée de la somme est la somme de la série dérivée sur l'intervalle ouvert de convergence)

$$\forall x \in]-1, 1[, L'(x) = \sum_{n=1}^{\infty} x^{n-1} = \frac{1}{1-x}$$

En primitivant, et comme $L(0) = 0$, on retrouve l'expression classique $L(x) = -\ln(1-x)$.

$$L \text{ est bien définie sur } D \text{ et } \forall x \in]-1, 1[, L(x) = -\ln(1-x)$$

Remarque : dès cette première question il faut être attentif au fait que si l'on peut considérer la série entière dérivée avec une variable complexe (la série entière est caractérisée par la suite des coefficients, peu importe où vit la variable), on ne peut identifier somme de la série dérivée et dérivée de la somme de la série que quand on se restreint à une variable réelle. En effet, le programme ne parle pas de dérivation d'une variable complexe. Par ailleurs, la somme demandée est censée être connue et on pouvait donner directement la valeur de $L(x)$.

2. Soit $z \in D$. On a

$$\forall t \in [0, 1], L(tz) = \sum_{n=1}^{\infty} \frac{z^n}{n} t^n$$

On a cette fois la somme d'une série entière (la variable est t) de rayon de convergence égal à $1/|z|$ (infini si $z = 0$) puisque le terme général est borné si $|tz| < 1$ et non borné si $|tz| > 1$. Par les mêmes arguments que ci-dessus, la dérivée de $\varphi_z : t \mapsto L(tz)$ vérifie

$$\forall |t| < \frac{1}{|z|}, \varphi'_z(t) = \sum_{n=1}^{\infty} z^n t^{n-1} = \frac{z}{1-zt}$$

Comme $|z| < 1$, on a $1/|z| > 1$ et cette expression est valable sur $[0, 1]$ a fortiori.

Remarque : on utilise ici un résultat sur les séries entières en fixant z et en considérant que la variable est t , et donc réelle. On peut donc utiliser le théorème de dérivation. Il est indispensable de bien préciser cela.

Posons $\psi(t) = (1-tz)e^{L(tz)} = (1-tz)e^{\varphi_z(t)}$. Par théorèmes d'opération, ψ est dérivable sur $[0, 1]$ et

$$\forall t \in [0, 1], \psi'^{L(tz)} + (1-tz)\varphi'_z{}^{L(tz)} = 0$$

Comme fonction de dérivée nulle sur un intervalle,

$$t \mapsto (1-tz)e^{L(tz)} \text{ est constante sur } [0, 1]$$

En 0, la fonction vaut 1. A l'aide de sa valeur en 1 (qui est donc aussi égale à 1), on obtient

$$\forall z \in D, \exp(L(z)) = \frac{1}{1-z}$$

3. Soit $z \in D$. Par inégalité triangulaire,

$$\forall n \in \mathbb{N}^*, \left| \sum_{k=1}^n \frac{z^k}{k} \right| \leq \sum_{k=1}^n \frac{|z|^k}{k}$$

Ce sont les termes généraux de suites convergentes (car $|z| < 1$) et en passant à la limite $n \rightarrow +\infty$, on obtient $|L(z)| \leq L(|z|)$. L'expression de la question 1 donne

$$\boxed{\forall z \in D, |L(z)| \leq -\ln(1 - |z|)}$$

Remarque : comme on est au début du problème, on a choisi de détailler en revenant aux sommes partielles puis en passant à la limite. On peut penser qu'un travail avec des sommes infinies est envisageable en notant l'existence des sommes infinies. En tous les cas, il faut absolument se garder de travailler avec des logarithmes de nombres complexes ou encore d'écrire des inégalités dans $\mathbb{C}$.

En particulier, pour $z \in D$, $|L(z^n)| \leq -\ln(1 - |z|^n) \underset{n \rightarrow +\infty}{\sim} |z|^n$ qui est le terme général d'une série positive convergente (car $|z| < 1$). Par théorème de comparaison, $\sum (L(z^n))$ converge absolument et donc (dans $\mathbb{C}$ qui est de dimension finie) :

$$\boxed{\forall z \in D, \sum (L(z^n)) \text{ converge}}$$

B. Développement de P en série entière

4. Si $(a_1, \dots, a_N) \in P_{n,N}$, alors $\forall k, ka_k \leq n$ (la somme de ces quantités positives valant n) et donc $a_k \in \llbracket 0, n \rrbracket$. On en déduit que

$$\boxed{P_{n,N} \text{ est fini de cardinal plus petit que } (n+1)^N}$$

Si $(a_1, \dots, a_N) \in P_{n,N}$, alors $(a_1, \dots, a_N, 0) \in P_{n,N+1}$ et $(a_1, \dots, a_N) \mapsto (a_1, \dots, a_N, 0)$ étant injective, le cardinal de $P_{n,N}$ est inférieur à celui de $P_{n,N+1}$ et donc

$$\boxed{(p_{n,N})_{N \geq 1} \text{ est croissante}}$$

Si $k \geq n+1$ et $a_k \geq 1$, $ka_k \geq n+1 > n$ et un uplet contenant a_k n'est pas dans $P_{n,N}$ (la somme des ka_k est $> n$).

Ainsi, un élément $P_{n,N}$ avec $N \geq n+1$ est uniquement composé de uplets du type $(a_1, \dots, a_n, 0, \dots, 0)$ avec $(a_1, \dots, a_n) \in P_{n,n}$. L'application $(a_1, \dots, a_n) \mapsto (a_1, \dots, a_n, 0, \dots, 0)$ est ainsi une bijection de $P_{n,n}$ dans $P_{n,N}$ (injectivité évidente et surjection justifiée juste avant) et $\forall N \geq n, p_{n,N} = p_{n,n}$.

La suite est donc constante à partir du rang n . Mais comme elle n'est définie qu'à partir du rang 1, on note plutôt que

$$\boxed{(p_{n,N})_{N \geq 1} \text{ est constante à partir du rang } \max(n, 1)}$$

Remarque : on a du ici prouver des inégalités et des égalités entre cardinaux. Il est naturel dans cette situation d'introduire de bonnes fonctions et d'utiliser le fait que deux ensembles finis en bijection ont même cardinal. Ou encore que s'il existe une injection de E dans F (finis) alors le cardinal de E est plus petit que celui de F . On pourra énoncer un résultat similaire avec une surjection.

5. On prouve par récurrence que la propriété

$$\forall z \in D, \prod_{k=1}^N \frac{1}{1-z^k} = \sum_{n=0}^{\infty} p_{n,N} z^n$$

est vraie pour tout entier $N \geq 1$.

- **Initialisation** : on a $p_{n,1} = 1$ (il n'existe qu'un seul $a_1 \geq 0$ tel que $a_1 = n$) et ainsi (l'existence des sommes est acquise)

$$\forall z \in D, \sum_{n=0}^{\infty} p_{n,1} z^n = \sum_{n=0}^{\infty} z^n = \frac{1}{1-z}$$

- **Hérédité** : supposons le résultat vrai à un rang $N \geq 1$ (récurrence forte qui est en fait inutile ici). On a alors (les sommes écrites existent, l'une par hypothèse de récurrence et l'autre comme somme géométrique de raison de module < 1)

$$\forall z \in D, \prod_{k=1}^{N+1} \frac{1}{1-z^k} = \left(\sum_{n=0}^{\infty} p_{n,N} z^n \right) \left(\sum_{n=0}^{\infty} z^{n(N+1)} \right)$$

Par théorème sur le produit de Cauchy de séries entières convergentes, on en déduit que

$$\forall z \in D, \prod_{k=1}^{N+1} \frac{1}{1-z^k} = \sum_{n=0}^{\infty} c_n z^n \quad \text{avec} \quad c_n = \sum_{k=0}^n p_{n-k,N} u_k$$

où $u_j = 1$ si $j = 0[N+1]$ et $u_j = 0$ sinon. Il s'agit donc de montrer que $c_n = p_{n,N+1}$.

Dans l'expression de c_n , on ne garde que les indices k multiples de $N+1$ (et donc du type $j(N+1)$) et ainsi

$$c_n = \sum_{0 \leq j \leq \frac{n}{N+1}} p_{n-(N+1)j,N}$$

Par ailleurs, on partitionne l'ensemble $P_{n,N+1}$ selon la valeur de la dernière coordonnée. Celle-ci vérifie $(N+1)a_{N+1} \leq n$ et on a donc

$$P_{n,N+1} = \bigcup_{0 \leq j \leq \frac{n}{N+1}} A_j \quad \text{avec} \quad A_j = \{(a_1, \dots, a_{N+1}) \in P_{n,N+1}, a_{N+1} = j\}$$

La réunion est disjointe et donc

$$p_{n,N+1} = \sum_{0 \leq j \leq \frac{n}{N+1}} |A_j|$$

A_j est en bijection avec l'ensemble des uplets $(a_1, \dots, a_N)$ tels que $a_1 + \dots + Na_N = n - j(N+1)$, c'est à dire $P_{n-j(N+1),N}$. On a donc $c_n = p_{n,N+1}$ et ceci donne le résultat au rang $N+1$ (comme annoncé plus haut).

Remarque : le résultat sur le produit de Cauchy est un cas particulier de sommation par paquet pour les familles sommables. Dans le cas simple des séries, on peut l'utiliser avec des séries absolument convergentes (c'est l'hypothèse à vérifier qui correspond à la sommabilité dans un cadre plus général). Le cours fournit ensuite un résultat qui en découle

sur le produit de Cauchy des séries entières. C'est celui-ci que nous avons utilisé. Il n'était donc pas utile de parler de convergence absolue (l'argument étant en fait inclu dans le théorème sur les séries entières). Il faut cependant bien préciser que c'est CE théorème que l'on utilise (plus loin, on utilisera celui avec les séries).

6. Posons $u_{n,N} = (p_{n,N+1} - p_{n,N})z^n$ et $v_{n,N} = |u_{n,N}| = (p_{n,N+1} - p_{n,N})|z|^n$. La sommabilité de $(u_{n,N})$ est par définition celle de $(v_{n,N})$ et comme cette dernière famille est positive, on peut considérer sa somme dans $\mathbb{R} \cup \{+\infty\}$ et la sommabilité équivaut au caractère fini de la somme. De plus, comme les termes sont positifs, on peut sommer dans l'ordre que l'on veut. On écrit que

$$\sum_{(n,N) \in \mathbb{N}^2} v_{n,N} = \sum_{N=0}^{\infty} \left(\sum_{n=0}^{\infty} (p_{n,N+1}|z|^n - p_{n,N}|z|^n) \right)$$

Pour $N \geq 1$, la question 5 permet d'écrire que

$$\sum_{n=0}^{\infty} (p_{n,N+1}|z|^n - p_{n,N}|z|^n) = \prod_{k=1}^{N+1} \frac{1}{1-|z|^k} - \prod_{k=1}^N \frac{1}{1-|z|^k}$$

Comme $\prod_{k=1}^N \frac{1}{1-|z|^k}$ est le terme général d'une suite convergente (de limite $P(|z|)$), la dualité suite-série (ou, si on préfère, le recours aux sommes partielles qui fait apparaître un télescopage) montre que l'on a ci-dessus le terme général d'une série convergente.

On a donc $\sum_{(n,N) \in \mathbb{N}^2} v_{n,N} < +\infty$ et la sommabilité de la famille.

Remarque : le programme permet, voire encourage, le travail dans $\mathbb{R} \cup \{+\infty\}$ pour prouver une sommabilité mais bien sûr cela n'est possible que dans le cas positif. Il faut insister sur l'hypothèse qui permet en fait un calcul formel.

On peut ainsi utiliser le théorème de Fubini avec la famille $(u_{n,N})$. Le même calcul que ci-dessus donne (on particularise le terme d'indice 0)

$$\sum_{(n,N) \in \mathbb{N}^2} u_{n,N} = \frac{1}{1-z} + \sum_{N=1}^{\infty} \left(\prod_{k=1}^{N+1} \frac{1}{1-z^k} - \prod_{k=1}^N \frac{1}{1-z^k} \right)$$

Par télescopage, dans la somme il reste la limite du terme moins le terme d'indice 1 et ainsi

$$\sum_{(n,N) \in \mathbb{N}^2} u_{n,N} = P(z)$$

Par ailleurs, en sommant dans l'autre sens, un télescopage apparaît aussi et on obtient (la suite $(p_{n,N})_{N \geq 0}$ est convergente de limite p_n et de premier terme nul)

$$\sum_{(n,N) \in \mathbb{N}^2} u_{n,N} = \sum_{n=0}^{\infty} \left(z^n \sum_{N=0}^{\infty} (p_{n,N+1} - p_{n,N}) \right) = \sum_{n=0}^{\infty} p_n z^n$$

On a finalement montré que

$$\boxed{\forall z \in D, \sum_{n=0}^{\infty} p_n z^n = P(z)}$$

Ceci justifie que le rayon de convergence de $\sum (p_n z^n)$ est plus grand que 1 (puisque l'on peut écrire la somme de la série entière quand $|z| < 1$). Comme $p_n \geq p_{n,1} = 1$, ce rayon est aussi plus petit que 1 (théorème de comparaison et $\sum z^n$ de rayon 1) et

$$\boxed{\sum (p_n z^n) \text{ est de rayon de convergence égal à } 1}$$

7. Pour $t > 0$, $e^{-t+i\theta}$ est de module < 1 et ainsi

$$\int_{-\pi}^{\pi} e^{-in\theta} P(e^{-t+i\theta}) d\theta = \int_{-\pi}^{\pi} \sum_{k=0}^{\infty} p_k e^{-in\theta} e^{-kt+ik\theta} d\theta$$

On veut intervertir somme et intégrale. Comme on intègre sur un segment, il nous suffit de montrer que la série de fonctions (de la variable θ) converge normalement sur le segment. Ceci a lieu puisque

$$\|\theta \mapsto p_k e^{-in\theta} e^{-kt+ik\theta}\|_{\infty} \leq p_k (e^{-t})^k$$

qui est le terme général d'une série convergente (car $0 \leq e^{-t} < 1$ et $\sum p_k z^k$ est de rayon 1). Ainsi

$$\int_{-\pi}^{\pi} e^{-in\theta} P(e^{-t+i\theta}) d\theta = \sum_{k=0}^{\infty} \int_{-\pi}^{\pi} p_k e^{-in\theta} e^{-kt+ik\theta} d\theta = \sum_{k=0}^{\infty} p_k e^{-kt} \int_{-\pi}^{\pi} e^{i(k-n)\theta} d\theta$$

L'intégrale est nulle si $k \neq n$ et vaut 2π si $k = n$. Ainsi

$$\int_{-\pi}^{\pi} e^{-in\theta} P(e^{-t+i\theta}) d\theta = 2\pi p_n e^{-nt}$$

On a donc prouvé que

$$\boxed{p_n = \frac{e^{nt}}{2\pi} \int_{-\pi}^{\pi} e^{-in\theta} P(e^{-t+i\theta}) d\theta}$$

Remarque : cette formule permet d'exprimer les coefficients d'une série entière en fonction de la somme de celle-ci. Il s'agit d'une formule de Cauchy qui permet de prouver l'unicité sous réserve d'existence d'un développement en série entière.

C. Contrôle de P

8. Comme $\exp(L(z)) = \frac{1}{1-z}$, quand $0 \leq x < 1$ et $\theta \in \mathbb{R}$, on a (on utilise l'égalité avec $xe^{i\theta}$ et avec x)

$$\frac{1-x}{1-xe^{i\theta}} = \exp(L(xe^{i\theta})) \exp(-L(x)) = \exp(L(xe^{i\theta}) - L(x))$$

Or, par définition de L ,

$$L(xe^{i\theta}) - L(x) = \sum_{n=1}^{\infty} \frac{x^n (e^{in\theta} - 1)}{n}$$

En passant à la partie réelle, on a donc (la partie réelle d'une somme infinie est la somme infinie des parties réelles, par exemple car le passage à la partie réelle est une opération continue et linéaire)

$$\operatorname{Re}(L(xe^{i\theta}) - L(x)) = \sum_{n=1}^{\infty} \frac{x^n (\cos(n\theta) - 1)}{n}$$

De plus $|e^z| = e^{\operatorname{Re}(z)}$ et ainsi

$$\left| \frac{1-x}{1-xe^{i\theta}} \right| = \exp \left(\sum_{n=1}^{\infty} \frac{x^n (\cos(n\theta) - 1)}{n} \right)$$

Dans la somme, les termes sont tous négatifs et la somme est donc plus petite que le premier terme. Par croissance de exp, on en déduit que

$$\boxed{\forall x \in [0, 1[, \forall \theta \in \mathbb{R}, \left| \frac{1-x}{1-xe^{i\theta}} \right| \leq \exp(-(1-\cos(\theta))x)}$$

Avec l'expression de la question 3, on a

$$\frac{P(xe^{i\theta})}{P(x)} = \lim_{N \rightarrow +\infty} \prod_{n=1}^N \frac{1-x^n}{1-x^n e^{in\theta}}$$

La première partie de la question donne

$$\left| \frac{1-x^n}{1-x^n e^{in\theta}} \right| \leq \exp(-(1-\cos(n\theta))x^n)$$

et par propriété de morphisme de l'exponentielle,

$$\left| \prod_{n=1}^N \frac{1-x^n}{1-x^n e^{in\theta}} \right| \leq \exp \left(- \sum_{n=1}^N (1-\cos(n\theta))x^n \right)$$

On s'intéresse à la somme qu'on écrit

$$\sum_{n=1}^N (1-\cos(n\theta))x^n = \sum_{n=1}^N x^n - \operatorname{Re} \left(\sum_{n=1}^N (xe^{i\theta})^n \right)$$

On ajoute à chaque somme le terme d'indice 0 (les termes se compensent) :

$$\sum_{n=1}^N (1-\cos(n\theta))x^n = \sum_{n=0}^N x^n - \operatorname{Re} \left(\sum_{n=0}^N (xe^{i\theta})^n \right)$$

Il reste à passer à la limite $N \rightarrow +\infty$ (continuité du passage au module, de l'exponentielle, toutes les séries convergent) pour obtenir

$$\boxed{\forall x \in [0, 1[, \forall \theta \in \mathbb{R}, \left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp \left(-\frac{1}{1-x} + \operatorname{Re} \left(\frac{1}{1-xe^{i\theta}} \right) \right)}$$

9. En multipliant par la quantité conjuguée puis en passant à la partie réelle, on a

$$\operatorname{Re} \left(\frac{1}{1-xe^{i\theta}} \right) = \frac{1-x\cos(\theta)}{(1-x\cos(\theta))^2 + (x\sin(\theta))^2} = \frac{1-x\cos(\theta)}{(1-x)^2 + 2x(1-\cos(\theta))}$$

Une réduction au même dénominateur donne

$$\frac{1}{1-x} - \operatorname{Re} \left(\frac{1}{1-xe^{i\theta}} \right) = \frac{(1-x)^2 + 2x(1-\cos(\theta)) - (1-x)(1-x\cos(\theta))}{(1-x)((1-x)^2 + 2x(1-\cos(\theta)))}$$

On s'intéresse au numérateur qui vaut $(x^2 + x)(1 - \cos(\theta))$ et qui est donc plus grand que $x(1 - \cos(\theta))$. En divisant par une quantité positive on ne change pas le sens de l'inégalité et on a prouvé que

$$\forall x \in [1/2, 1[, \forall \theta \in \mathbb{R}, \frac{1}{1-x} - \operatorname{Re} \left(\frac{1}{1-xe^{i\theta}} \right) \geq \frac{x(1 - \cos(\theta))}{(1-x)((1-x)^2 + 2x(1 - \cos(\theta)))}$$

Remarque : ceci est valable pour $x \in [0, 1[$.

La question précédente et la croissance de l'exponentielle donnent ainsi

$$\left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp \left(- \frac{x(1 - \cos(\theta))}{(1-x)((1-x)^2 + 2x(1 - \cos(\theta)))} \right)$$

On distingue deux cas.

- Si $(1-x)^2 \geq x(1 - \cos(\theta))$ alors $(1-x)^2 + 2x(1 - \cos(\theta)) \leq 3(1-x)^2$ et donc

$$\frac{x(1 - \cos(\theta))}{(1-x)((1-x)^2 + 2x(1 - \cos(\theta)))} \geq \frac{x(1 - \cos(\theta))}{3(1-x)^3} \geq \frac{1 - \cos(\theta)}{6(1-x)^3}$$

En passant à l'opposé, on change le sens de l'inégalité puis on peut composer par l'exponentielle (croissante) pour obtenir

$$\left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp \left(- \frac{1 - \cos(\theta)}{6(1-x)^3} \right)$$

- Sinon $(1-x)^2 + 2x(1 - \cos(\theta)) \leq 3x(1 - \cos(\theta))$ et de même

$$\left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp \left(- \frac{1}{3(1-x)} \right)$$

$$\forall x \in [1/2, 1[, \forall \theta \in \mathbb{R}, \left| \frac{P(xe^{i\theta})}{P(x)} \right| \leq \exp \left(- \frac{1 - \cos(\theta)}{6(1-x)^3} \right) \quad \text{ou} \quad \exp \left(- \frac{1}{3(1-x)} \right)$$

D. Intermède : quelques estimations de sommes

10. Je remarque tout d'abord que

$$\forall x > 0, \varphi_{n,\alpha}(x) = \frac{e^{-\alpha x}}{\psi(x)^n} \quad \text{où} \quad \psi(x) = \frac{1 - e^{-x}}{x}$$

et on a

$$\forall x \neq 0, \psi(x) = \sum_{n=1}^{\infty} \frac{(-1)^{n-1}}{n!} x^{n-1}$$

L'expression reste valable en posant $\psi(0) = 1$ et ψ est DSE de rayon de convergence infini et est donc de classe C^∞ sur $\mathbb{R}$. Elle est > 0 sur $\mathbb{R}^+$ et par théorèmes d'opérations, $\varphi_{n,\alpha}$ prolongée en 0 par la valeur 1 est de classe C^∞ sur $\mathbb{R}^+$.

L'intégrabilité de $\varphi_{n,\alpha}$ et de sa dérivée sur $\mathbb{R}^+$ ne posent donc de problème qu'au voisinage de $+\infty$.

$(1 - e^{-x})^n$ est de limite 1 en $+\infty$. On a immédiatement $\varphi_{n,\alpha}(x) = o(1/x^2)$ au voisinage de

$+\infty$ par croissances comparées et puisque $\alpha > 0$. La même domination est valable pour $\varphi'_{n,\alpha}$ car

$$\varphi'^{-\alpha x}_{n,\alpha} \left(-\frac{\alpha x^n}{(1-e^{-x})^n} - \frac{nx^{n-1}}{(1-e^{-x})^n} - \frac{nx^n e^{-x}}{(1-e^{-x})^{n+1}} \right)$$

Par comparaison aux fonctions de Riemann, on a donc l'intégrabilité voulue au voisinage de $+\infty$ et

$\varphi_{n,\alpha}$ et $\varphi'_{n,\alpha}$ sont intégrables sur $\mathbb{R}^{+*}$

Remarque : l'étude en 0 en introduisant la fonction ψ qui est DSE est élégante mais ne viendra pas forcément à l'esprit du candidat. Il est bien sûr possible de remarquer que par théorèmes d'opération la fonction $\varphi_{n,\alpha}$ est de classe C^1 sur $\mathbb{R}^{+*}$ et de faire une étude locale de $\varphi_{n,\alpha}$ et de sa dérivée en 0. On montre facilement que ces fonction sont $O(1)$ au voisinage de 0 et donc intégrables sur un tel voisinage.

11. Soit $t > 0$. Quand $k \rightarrow +\infty$, $\frac{k^n e^{-kt\alpha}}{(1-e^{-kt})^n} \sim k^n e^{-kt\alpha} = o(1/k^2)$ car $t\alpha > 0$ (croissances comparées). Ceci est le terme général d'une série positive convergente. Comme le terme général est strictement positif, la somme l'est aussi.

$S_{n,\alpha}$ est définie et > 0 sur $\mathbb{R}^{+*}$

Par relation de Chasles, on a

$$\int_0^{(p+1)t} \varphi_{n,\alpha}(t) dt = \sum_{k=0}^p \int_{kt}^{(k+1)t} \varphi_{n,\alpha}(x) dx$$

Dans l'intégrale, on effectue une intégration par partie en primitivant $u'(x) = 1$ en $u(x) = x - kt$ et en dérivant $v = \varphi_{n,\alpha}$. Ceci est licite puisque u et v sont de classe C^1 sur le segment d'intégration (vu pour v en question précédente). On obtient

$$\int_{kt}^{(k+1)t} \varphi_{n,\alpha}(x) dx = t\varphi_{n,\alpha}((k+1)t) - \int_{kt}^{(k+1)t} \varphi'_{n,\alpha}(x)(x-kt) dx$$

On somme ces relations pour $k = 0, \dots, p$ et on fait tendre p vers $+\infty$ (on a montré que la somme du membre de gauche existe et il en est donc de même de celle à droite) :

$$\int_0^{+\infty} \varphi_{n,\alpha}(t) dt = \sum_{k=0}^{\infty} \left(t\varphi_{n,\alpha}((k+1)t) - \int_{kt}^{(k+1)t} \varphi'_{n,\alpha}(x)(x-kt) dx \right)$$

Comme $S_{n,\alpha}(t)$ existe on peut écrire

$$S_{n,\alpha}(t) = \frac{1}{t^n} \sum_{k=1}^{\infty} \varphi_{n,\alpha}(kt) = \frac{1}{t^n} \sum_{k=0}^{\infty} \varphi_{n,\alpha}((k+1)t)$$

On peut scinder la somme obtenue plus haut et on obtient

$$\int_0^{+\infty} \varphi_{n,\alpha}(t) dt = t^{n+1} S_{n,\alpha}(t) - \sum_{k=0}^{\infty} \int_{kt}^{(k+1)t} \varphi'_{n,\alpha}(x)(x-kt) dx$$

On en déduit que

$$S_{n,\alpha}(t) - \frac{1}{t^{n+1}} \int_0^{\infty} \varphi_{n,\alpha}(x) dx = \frac{\alpha_n(t)}{t^n} \text{ avec } \alpha_n(t) = \sum_{k=0}^{\infty} \int_{kt}^{(k+1)t} \varphi'_{n,\alpha}(x) \frac{x-kt}{t} dx$$

Pour $x \in [kt, (k+1)t]$, on a $0 \leq \frac{x-kt}{t} \leq 1$ et ainsi (l'intégrale qui majore existe avec l'intégrabilité de $\varphi'_{n,\alpha}$ sur $\mathbb{R}^+$)

$$|\alpha_n(t)| \leq \sum_{k=0}^{\infty} \int_{kt}^{(k+1)t} |\varphi'_{n,\alpha}| = \int_0^{\infty} |\varphi'_{n,\alpha}|$$

α_n est ainsi bornée sur $\mathbb{R}^{+*}$ et on a montré (en remplaçant $\varphi_{n,\alpha}$ par son expression) que

$$S_{n,\alpha}(t) = \frac{1}{t^{n+1}} \int_0^{\infty} \frac{x^n e^{-\alpha x}}{(1-e^{-x})^n} dx + O\left(\frac{1}{t^n}\right) \text{ quand } t \rightarrow 0^+$$

Remarque : il faut prendre garde ici à justifier que les quantités écrites existent, par exemple quand on écrit $\sum_{n=0}^{\infty} (a_n + b_n)$ comme l'addition de deux sommes de séries. Il est aussi conseillé quand on effectue une première intégration par parties dans un problème, de bien préciser les hypothèses à vérifier. Notez qu'ici ces intégrations par parties ne se font pas dans des intégrales généralisées (auquel cas il faudrait être encore plus attentif).

12. Si $x > 0$, $\frac{1}{1-e^{-x}}$ est la somme de la série géométrique (e^{-kx}) et donc

$$\int_0^{\infty} \frac{xe^{-x}}{1-e^{-x}} dx = \int_0^{\infty} xe^{-x} \sum_{k=0}^{\infty} e^{-kx} dx = \int_0^{\infty} \sum_{k=1}^{\infty} xe^{-kx} dx$$

On veut opérer une interversion entre intégrale et somme infinie.

Posons $f_k : x \mapsto xe^{-kx}$. (f_k) est une suite de fonctions continues sur $\mathbb{R}^+$ et intégrables ($f_k(x) = o(1/x^2)$ au voisinage de $+\infty$ qui est le seul problème).

$\sum(f_k)$ converge simplement sur $\mathbb{R}^{+*}$ vers $x \mapsto \frac{xe^{-x}}{1-e^{-x}}$ qui est continue sur $\mathbb{R}^{+*}$. Enfin, une intégration par parties donne (on primitive e^{-kx} , on dérive x et on vérifie que le crochet admet une limite en $+\infty$, seul endroit où un problème se pose)

$$\int_0^{\infty} |f_k| = \left[\frac{xe^{-kx}}{k} \right]_0^{+\infty} + \frac{1}{k} \int_0^{\infty} e^{-kx} dx$$

Le crochet est nul et la seconde intégrale vaut $1/k$. Ainsi

$$\int_0^{\infty} |f_k| = \frac{1}{k^2}$$

est le terme général d'une série convergente. On a donc par théorème d'intégration terme à terme

$$\int_0^{\infty} \frac{xe^{-x}}{1-e^{-x}} dx = \sum_{k=1}^{\infty} \int_0^{\infty} xe^{-kx} dx$$

et le même calcul d'intégrale permet de conclure que

$$\int_0^{\infty} \frac{xe^{-x}}{1-e^{-x}} dx = \sum_{k=1}^{\infty} \frac{1}{k^2} = \frac{\pi^2}{6}$$

Remarque : on est passé un peu vite sur l'intégration par parties mais ce n'est pas le coeur de la question. Il faut quand même noter quelquechose sur cette IPP. Par ailleurs, il est important de bien détailler les hypothèses à vérifier pour le théorème utilisé. On aurait pu choisir le théorème de convergence dominée, mais comme on doit dans tous les cas calculer l'intégrale, celui utilisé semble plus naturel.

E. Contrôle des fonctions caractéristiques

13. Le cours nous indique que si Y admet un moment d'ordre 2 alors elle admet un moment d'ordre 1 et

$$0 \leq \mathbb{V}(Y) = \mathbb{E}((Y - \mathbb{E}(Y))^2) = \mathbb{E}(Y^2) - \mathbb{E}(Y)^2$$

Ainsi, dans ce cas, $\mathbb{E}(Y)^2 \leq \mathbb{E}(Y^2)$.

Ici, on a (les variables sont bornées et admettent donc des moments à tout ordre)

$$|\Phi_X(\theta)|^2 = \mathbb{E}(\cos(\theta X))^2 + \mathbb{E}(\sin(\theta X))^2 \leq \mathbb{E}(\cos(\theta X)^2) + \mathbb{E}(\sin(\theta X)^2)$$

et par linéarité du passage à l'espérance, ceci est majoré par l'espérance de 1, c'est à dire
1. Reste à passer à la racine carrée (croissante) pour obtenir

$$|\Phi_X(\theta)| \leq 1$$

14. On utilise la formule de transfert :

$$\begin{aligned} \Phi_{aX+b}(\theta) &= \sum_{k=1}^{\infty} (\cos((ak+b)\theta) + i \sin((ak+b)\theta)) p(1-p)^{k-1} \\ &= \frac{p}{1-p} \sum_{k=1}^{\infty} e^{i(ak+b)\theta} (1-p)^k \\ &= \frac{pe^{ib\theta}}{1-p} \sum_{k=1}^{\infty} (qe^{ia\theta})^k \\ &= \frac{pe^{ib\theta}}{1-p} \frac{qe^{ia\theta}}{1-qe^{ia\theta}} \end{aligned}$$

en ayant reconnu une somme géométrique de raison de module < 1 . On a ainsi

$$\Phi_{aX+b}(\theta) = \frac{pe^{i(a+b)\theta}}{1-qe^{ia\theta}}$$

15. X^k étant à valeurs positives, on peut écrire son espérance dans $\mathbb{R} \cup \{+\infty\}$. La formule de transfert donne

$$\mathbb{E}(X^k) = \sum_{n=1}^{\infty} n^k p(1-p)^{n-1}$$

Comme $1-p \in]0, 1[$, par croissances comparées on a $n^k p(1-p)^{n-1} = o(1/n^2)$ qui est le terme général d'une série positive convergente. La somme ci-dessus est donc finie et

$$\mathbb{E}(X^k) < +\infty.$$

Toujours par formule de transfert, on a

$$\Phi_X(\theta) = \sum_{n=1}^{\infty} e^{in\theta} p(1-p)^{n-1}$$

On veut appliquer le théorème de régularité des sommes de séries.

- $f_n : \theta \mapsto e^{in\theta} p(1-p)^{n-1}$ est de classe C^∞ sur $\mathbb{R}$ de dérivée k -ième $\theta \mapsto (in)^k e^{in\theta} p(1-p)^{n-1}$.

- $\|f_n^{(k)}\|_\infty \leq n^k p(1-p)^{n-1}$ est le terme général d'une série convergente (vu ci-dessus) et la série ainsi que toutes les séries dérivées convergent normalement sur $\mathbb{R}$.

Le théorème s'applique et indique que $\Phi_X \in C^\infty(\mathbb{R})$ et que

$$\forall \theta \in \mathbb{R}, \forall k \in \mathbb{N}, \Phi_X^{(k)}(\theta) = \sum_{n=1}^{\infty} (in)^k e^{in\theta} p(1-p)^{n-1}$$

En particulier, on a

$$\boxed{\forall k \in \mathbb{N}, \Phi_X^{(k)}(0) = i^k \mathbb{E}(X^k)}$$

16. Avec la question 14, on a

$$\Phi_X(\theta) = \frac{pe^{i\theta}}{1 - qe^{i\theta}} = pe^{i\theta} \frac{P_0(qe^{i\theta})}{1 - qe^{i\theta}}$$

Supposons avoir trouvé un polynôme P_k tel que

$$\forall \theta, \Phi_X^{(k)}(\theta) = pi^k e^{i\theta} \frac{P_k(qe^{i\theta})}{(1 - qe^{i\theta})^{k+1}}$$

On peut alors dériver cette expression pour obtenir

$$\Phi_X^{(k+1)}(\theta) = pi^k e^{i\theta} \left(\frac{iP_k(qe^{i\theta})}{(1 - qe^{i\theta})^{k+1}} + \frac{qie^{i\theta} P_k'(qe^{i\theta})}{(1 - qe^{i\theta})^{k+1}} + \frac{(k+1)iqe^{i\theta} P_k(qe^{i\theta})}{(1 - qe^{i\theta})^{k+2}} \right)$$

En posant $P_{k+1} = (1-X)P_k + (1-X)XP_k' + (k+1)XP_k$, on obtient alors

$$\forall \theta, \Phi_X^{(k+1)}(\theta) = pi^{k+1} e^{i\theta} \frac{P_{k+1}(qe^{i\theta})}{(1 - qe^{i\theta})^{k+2}}$$

De plus $P_0(0) = 1$ et si $P_k(0) = 0$ alors $P_{k+1}(0) = 0$ et les P_k sont indépendants de p . On a donc montré par récurrence que

$$\boxed{\forall k \in \mathbb{N}, \exists P_k \in \mathbb{C}[X], \Phi_X^{(k)} : \theta \mapsto pi^k e^{i\theta} \frac{P_k(qe^{i\theta})}{(1 - qe^{i\theta})^{k+1}} \text{ et } P_k(0) = 1}$$

Remarque : il est sans doute préférable d'annoncer la récurrence dès le début du raisonnement. Comme on est bien avancé dans le problème, on peut aussi se permettre ce type de rédaction. Il faut quand même que le mot "récurrence" apparaisse.

17. Avec les deux questions précédentes, on a

$$\mathbb{E}(X^k) = \frac{1}{i^k} \Phi_X^{(k)}(0) = \frac{P_k(q)}{p^k}$$

et ainsi

$$\mathbb{E}(X^k) - \frac{1}{p^k} = \frac{P_k(q) - 1}{p^k} = \frac{P_k(q) - P_k(0)}{p^k} = \frac{1}{p^k} \int_0^q P_k'(t) dt$$

$t \mapsto P_k'(t)$ est continue sur $[0, 1]$ et est donc bornée sur ce segment. En notant C_k sa norme infinie sur $[0, 1]$, on a alors

$$\left| \mathbb{E}(X^k) - \frac{1}{p^k} \right| \leq \frac{1}{p^k} \int_0^q |P_k'(t)| dt \leq \frac{qC_k}{p^k}$$

Il existe $(C_k)_{k \in \mathbb{N}}$ indépendante de p telle que $\forall k, \left| \mathbb{E}(X^k) - \frac{1}{p^k} \right| \leq \frac{qC_k}{p^k}$

Si C_k était nulle, il suffirait de prendre $1 + C_k$ pour avoir une constante > 0 .

18. X suivant la loi géométrique de paramètre p , on a $\mathbb{E}(X) = \frac{1}{p}$. On écrit que

$$\begin{aligned} (X - \mathbb{E}(X))^4 &= X^4 - \frac{4X^3}{p} + \frac{6X^2}{p^2} - \frac{4X}{p^3} + \frac{1}{p^4} \\ &= \left(X^4 - \frac{1}{p^4} \right) - \frac{4}{p} \left(X^3 - \frac{1}{p^3} \right) + \frac{6}{p^2} \left(X^2 - \frac{1}{p^2} \right) - \frac{4}{p^3} \left(X - \frac{1}{p} \right) \end{aligned}$$

les "constantes" s'annulant dans l'expression ci-dessus. Par linéarité de l'espérance et avec la question précédente,

$$\begin{aligned} \mathbb{E}((X - \mathbb{E}(X))^4) &\leq \left(\mathbb{E}(X^4) - \frac{1}{p^4} \right) - \frac{4}{p} \left(\mathbb{E}(X^3) - \frac{1}{p^3} \right) + \frac{6}{p^2} \left(\mathbb{E}(X^2) - \frac{1}{p^2} \right) - \frac{4}{p^3} \left(\mathbb{E}(X) - \frac{1}{p} \right) \\ &\leq \frac{q}{p^4} (C_4 + 4C_3 + 6C_2 + 4C_1) \end{aligned}$$

Il existe $K > 0$ indépendante de p telle que $\mathbb{E}((X - \mathbb{E}(X))^4) \leq \frac{Kq}{p^4}$

19. On a $0 \leq Y^2 \leq 1 + Y^4$ (en distinguant selon que $|Y(\omega)| \leq 1$ et $|Y(\omega)| \geq 1$). Comme Y^4 et 1 admettent une espérance, il en est donc de même pour Y^2 . De plus, par formule de transfert,

$$\mathbb{E}(Y^2) = \sum_{y \in Y(\Omega)} y^2 \mathbb{P}(Y = y) = \sum_{y \in Y(\Omega)} y^2 \sqrt{\mathbb{P}(Y = y)} \sqrt{\mathbb{P}(Y = y)}$$

En utilisant l'inégalité de Cauchy-Schwarz dans l'ensemble des familles de carré sommable, on obtient

$$\mathbb{E}(Y^2) \leq \left(\sum_{y \in Y(\Omega)} y^4 \mathbb{P}(Y = y) \right)^{1/2} \left(\sum_{y \in Y(\Omega)} \mathbb{P}(Y = y) \right)^{1/2} = \sqrt{\mathbb{E}(Y^4)}$$

Y^2 est d'espérance finie et $\mathbb{E}(Y^2) \leq \sqrt{\mathbb{E}(Y^4)}$

On rappelle que l'inégalité de Cauchy-Schwarz indique que pour Y_1, Y_2 admettant des moments d'ordre 2, on a $Y_1 Y_2$ qui admet une espérance et

$$\mathbb{E}(Y_1 Y_2)^2 \leq \mathbb{E}(Y_1^2) \mathbb{E}(Y_2^2)$$

En appliquant ceci avec $Y_1 = |Y|$ et $Y_2 = Y^2$, on trouve que $|Y|^3$ est d'espérance finie et que $\mathbb{E}(|Y|^3)^2 \leq \mathbb{E}(Y^4) \mathbb{E}(Y^2)$ et ainsi

Y^3 est d'espérance finie et $\mathbb{E}(|Y|^3) \leq (\mathbb{E}(Y^4))^{3/4}$

20. Si f est une fonction de classe C^3 de $\mathbb{R}$ dans $\mathbb{C}$, la formule de Taylor avec reste intégrale donne

$$\forall u \in \mathbb{R}, f(u) = f(0) + uf'(0) + \frac{u^2}{2} f''(0) + \int_0^u \frac{(u-t)^2}{2} f^{(3)}(t) dt$$

On applique cela à $u \mapsto e^{iu}$ et on utilise l'inégalité triangulaire avec l'intégrale. Comme $|f^{(3)}(u)| = 1$, on obtient

$$\forall u \in \mathbb{R}, \left| e^{iu} - 1 - iu + \frac{u^2}{2} \right| \leq \int_{[0,u]} \frac{|u-t|^2}{2} dt$$

où il faut faire attention au signe de u pour "mettre les bornes dans le bon sens" (d'où l'écriture de l'intégrale). En distinguant selon le signe de u , on voit aisément que le majorant vaut toujours $\frac{|u|^3}{6}$. Ainsi

$$\boxed{\forall u \in \mathbb{R}, \left| e^{iu} - 1 - iu + \frac{u^2}{2} \right| \leq \frac{|u|^3}{6}}$$

Remarque : on aurait, bien sûr, pu utiliser une inégalité de Taylor-Lagrange.

Par formule de transfert, on a

$$\Phi_Y(\theta) = \mathbb{E}(e^{i\theta Y}) = \sum_{y \in Y(\Omega)} e^{i\theta y} \mathbb{P}(Y = y)$$

Notons par ailleurs que

$$1 = \sum_{y \in Y(\Omega)} \mathbb{P}(Y = y)$$

$$0 = i\theta \mathbb{E}(Y) = \sum_{y \in Y(\Omega)} i\theta y \mathbb{P}(Y = y)$$

$$\frac{\theta^2}{2} \mathbb{E}(Y^2) = \sum_{y \in Y(\Omega)} \frac{(\theta y)^2}{2} \mathbb{P}(Y = y)$$

En combinant ces égalités et par inégalité triangulaire, on a donc

$$\left| \Phi_Y(\theta) - 1 + \frac{\theta^2}{2} \mathbb{E}(Y^2) \right| \leq \sum_{y \in Y(\Omega)} \left| e^{i\theta y} - 1 - i\theta y + \frac{(\theta y)^2}{2} \right| \mathbb{P}(Y = y)$$

On peut alors utiliser la première partie de la question (avec $u = \theta y$) et on obtient

$$\left| \Phi_Y(\theta) - 1 + \frac{\theta^2}{2} \mathbb{E}(Y^2) \right| \leq \sum_{y \in Y(\Omega)} \frac{|\theta y|^3}{6} \mathbb{P}(Y = y) = \frac{|\theta|^3}{6} \mathbb{E}(|Y|^3)$$

Avec la question 19, on peut conclure que

$$\boxed{\left| \Phi_Y(\theta) - 1 + \frac{\theta^2}{2} \mathbb{E}(Y^2) \right| \leq \frac{|\theta|^3}{6} (\mathbb{E}(Y^4))^{3/4}}$$

Remarque : on obtient une constante $1/6$ meilleure que celle demandée. La constante $1/3$ apparaît sans doute si on traite séparément les parties réelle et imaginaire.

21. On écrit alors que

$$\left| \Phi_Y(\theta) - \exp\left(-\frac{\mathbb{E}(Y^2)\theta^2}{2}\right) \right| \leq \left| \Phi_Y(\theta) - 1 + \frac{\theta^2}{2} \mathbb{E}(Y^2) \right| + \left| 1 - \frac{\theta^2}{2} \mathbb{E}(Y^2) - \exp\left(-\frac{\mathbb{E}(Y^2)\theta^2}{2}\right) \right|$$

Le premier terme du membre de droite est majoré avec la question 20.
 Pour le second, on utilise l'inégalité de Taylor-Lagrange qui donne pour $v \geq 0$

$$|e^{-v} - 1 + v| \leq \frac{v^2}{2}$$

Avec $v = \frac{\theta^2}{2} \mathbb{E}(Y^2)$, on majorer le second terme et avec la question 19, on obtient finalement (en reprenant la constante $1/3$ du sujet)

$$\left| \Phi_Y(\theta) - \exp\left(-\frac{\mathbb{E}(Y^2)\theta^2}{2}\right) \right| \leq \frac{|\theta|^3}{3} (\mathbb{E}(Y^4))^{3/4} + \frac{\theta^4}{8} \mathbb{E}(Y^4)$$

Remarque : à ce niveau du problème, et d'autant plus que le résultat n'est pas donné par l'énoncé, on n'a pas à s'étendre plus sur l'inégalité obtenue plus haut avec Taylor-Lagrange.

F. Convergence vers une gaussienne

22. On procède par récurrence sur n , l'hypothèse étant que pour tout choix des z_i et u_i de module ≤ 1 on a l'inégalité voulue.

- Pour $n = 1$, le résultat est immédiat et c'est une égalité. Pour $n = 2$, on écrit que

$$|z_1 z_2 - u_1 u_2| = |(z_1 - u_1)z_2 + u_1(z_2 - u_2)| \leq |z_1 - u_1||z_2| + |u_1||z_2 - u_2|$$

ce qui donne le résultat au rang 2 (puisque $|z_2| \leq 1$ et $|u_1| \leq 1$).

- Supposons le résultat acquis jusqu'à un rang $n \geq 2$ (récurrence généralisée qui est inutile en fait ici). On se donne des u_i, z_i de module ≤ 1 . Comme $|z_1 \dots z_n| \leq 1$ et $|u_1 \dots u_n| \leq 1$ ainsi que $|z_{n+1}| \leq 1$ et $|u_{n+1}| \leq 1$, on peut appliquer le résultat au rang 2 :

$$\left| \prod_{k=1}^{n+1} z_k - \prod_{k=1}^{n+1} u_k \right| \leq \left| \prod_{k=1}^n z_k - \prod_{k=1}^n u_k \right| + |z_{n+1} - u_{n+1}|$$

L'hypothèse au rang n donne alors le résultat au rang $n + 1$.

$$\text{Si } |z_1|, \dots, |z_n|, |u_1|, \dots, |u_n| \leq 1, \left| \prod_{k=1}^n z_k - \prod_{k=1}^n u_k \right| \leq \sum_{k=1}^n |z_k - u_k|$$

23. On a $\mathbb{E}(Z_k) = \frac{1}{1-e^{-kt}}$. La question 14 utilisée avec $a = k$ et $b = -k\mathbb{E}(Z_k)$ donne

$$\Phi_{Y_k}(\theta) = \frac{(1 - e^{-kt}) \exp\left(-i\theta \frac{ke^{-kt}}{1-e^{-kt}}\right)}{1 - e^{-kt+ik\theta}}$$

On a ainsi

$$\prod_{k=1}^n \Phi_{Y_k}(\theta) = \prod_{k=1}^n \frac{1 - e^{-kt}}{1 - e^{-kt+ik\theta}} \exp\left(-i\theta \sum_{k=1}^n \frac{ke^{-kt}}{1 - e^{-kt}}\right)$$

Avec le résultat de la question 3 donnant $P(z)$ comme limite d'un produit, on a

$$\lim_{n \rightarrow +\infty} \prod_{k=1}^n \frac{1 - e^{-kt}}{1 - e^{-kt+ik\theta}} = \frac{P(e^{-t}e^{i\theta})}{P(e^{-t})}$$

On remarque aussi que

$$m_t = S_{1,1}(t) = \sum_{k=1}^{\infty} \frac{ke^{-kt}}{1 - e^{-kt}}$$

Par continuité de l'exponentielle, on a donc

$$\lim_{n \rightarrow +\infty} \exp \left(-i\theta \sum_{k=1}^n \frac{ke^{-kt}}{1 - e^{-kt}} \right) = \exp(-i\theta m_t)$$

On a ainsi montré que

$$h(t, \theta) = \lim_{n \rightarrow +\infty} \prod_{k=1}^n \Phi_{Y_k}(\theta)$$

Par ailleurs, on a

$$\sigma_t^2 = \sum_{k=1}^{\infty} \frac{k^2 e^{-kt}}{(1 - e^{-kt})^2}$$

et par continuité de l'exponentielle (et propriété de morphisme)

$$e^{-\frac{\sigma_t^2 \theta^2}{2}} = \lim_{n \rightarrow +\infty} \prod_{k=1}^n \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right)$$

Comme $|\Phi_{Y_k}(\theta)| \leq 1$ et $0 \leq \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \leq 1$, on peut utiliser la question 22 pour affirmer que

$$\left| \prod_{k=1}^n \Phi_{Y_k}(\theta) - \prod_{k=1}^n \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \right| \leq \sum_{k=1}^n \left| \Phi_{Y_k}(\theta) - \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \right|$$

Or, $\mathbb{E}(Y_k^2) = k^2 \mathbb{V}(Z_k) = k^2 \frac{e^{-kt}}{(1 - e^{-kt})^2}$ et ainsi

$$\left| \prod_{k=1}^n \Phi_{Y_k}(\theta) - \prod_{k=1}^n \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \right| \leq \sum_{k=1}^n \left| \Phi_{Y_k}(\theta) - \exp \left(-\frac{\mathbb{E}(Y_k^2) \theta^2}{2} \right) \right|$$

On peut alors utiliser la question 21 :

$$\left| \prod_{k=1}^n \Phi_{Y_k}(\theta) - \prod_{k=1}^n \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \right| \leq \sum_{k=1}^n \left(\frac{|\theta|^3}{3} (\mathbb{E}(Y_k^4))^{3/4} + \frac{\theta^4}{8} \mathbb{E}(Y_k^4) \right)$$

La question 18 donne une majoration de $\mathbb{E}(\frac{1}{k^4} Y_k^4)$ et ainsi

$$\begin{aligned} & \left| \prod_{k=1}^n \Phi_{Y_k}(\theta) - \prod_{k=1}^n \exp \left(\frac{-\theta^2 k^2 e^{-kt}}{2(1 - e^{-kt})^2} \right) \right| \\ & \leq \sum_{k=1}^n \left(\frac{K^{3/4} |\theta|^3}{3} \left(k^4 \frac{e^{-kt}}{(1 - e^{-kt})^4} \right)^{3/4} + \frac{\theta^4}{8} \left(k^4 \frac{e^{-kt}}{(1 - e^{-kt})^4} \right) \right) \end{aligned}$$

En faisant tendre n vers l'infini, on voit apparaître $S_{3,3/4}$ et $S_{4,1}$ dans le membre de droite et $h(t, \theta) - e^{-\frac{\sigma_t^2 \theta^2}{2}}$ dans le membre de gauche. On peut conclure que

$$\left| h(t, \theta) - e^{-\frac{\sigma_t^2 \theta^2}{2}} \right| \leq K^{3/4} \frac{|\theta|^3}{3} S_{3,3/4}(t) + K \frac{\theta^4}{8} S_{4,1}(t)$$

Remarque : l'énoncé fait disparaître les constantes $1/3$ et $1/8$. On a donc une inégalité plus précise.

24. En utilisant la question 11, on obtient (quand $t \rightarrow 0^+$)

$$\sigma_t^2 = S_{2,1}(t) = \frac{1}{t^3} \int_0^\infty \frac{x^2 e^{-x}}{(1 - e^{-x})^2} + O\left(\frac{1}{t^2}\right)$$

Avec le résultat admis en question 2, le membre de droite équivaut à $\frac{\pi^2}{3t^3}$. Comme on peut élever un équivalent à une puissance constante, on conclut que

$$\sigma_t \underset{t \rightarrow 0^+}{\sim} \frac{\pi}{\sqrt{3}t^{3/2}}$$

De façon similaire, on a

$$m_t = S_{1,1}(t) = \frac{\pi^2}{6t^2} + O\left(\frac{1}{t}\right)$$

et donc, puisque $1/\sigma_t = O(t^{3/2})$,

$$\frac{1}{\sigma_t} \left(m_t - \frac{\pi^2}{6t^2} \right) = O(\sqrt{t}) \rightarrow 0$$

ce qui donne

$$\lim_{t \rightarrow 0^+} \zeta(t, u) = 1$$

Par ailleurs, la question 23 avec $\theta = u/\sigma_t$ indique que

$$\left| h\left(u, \frac{u}{\sigma_t}\right) - e^{-u^2/2} \right| \leq K^{3/4} \frac{|u|^3}{\sigma_t^3} S_{3,3/4}(t) + K \frac{u^4}{\sigma_t^4} S_{4,1}(t)$$

D'après la question 11, $S_{3,3/4}(t) = O(t^{-4})$ et comme $\sigma_t = O(t^{-3/2})$, on a $\frac{S_{3,3/4}(t)}{\sigma_t^3} = O(\sqrt{t})$.

De la même façon, $\frac{S_{4,1}(t)}{\sigma_t^4} = O(t)$.

Le majorant ci-dessus est donc de limite nulle en 0^+ et $h(u, \frac{u}{\sigma_t}) \rightarrow e^{-u^2/2}$. On a finalement

$$\lim_{t \rightarrow 0^+} j(t, u) = e^{-u^2/2}$$

25. Posons $g(\theta) = \frac{1 - \cos(\theta)}{\theta^2}$ qui est bien définie sur $[-\pi, \pi] \setminus \{0\}$ et strictement positive sur cet ensemble.

Quand $\theta \rightarrow 0$, $g(\theta) \rightarrow \frac{1}{2}$ et g est donc prolongeable par continuité en 0.

On obtient une fonction continue sur le segment $[-\pi, \pi]$ et qui ne s'annule pas. Elle admet donc un minimum $\alpha > 0$ (théorème de compacité) :

$$\boxed{\exists \alpha > 0, \forall \theta \in [-\pi, \pi], 1 - \cos(\theta) \geq \alpha \theta^2}$$

On note que

$$|h(t, \theta)| = \left| \frac{P(e^{-t} e^{i\theta})}{P(e^{-t})} \right|$$

Quand $t \rightarrow 0^+$, $e^{-t} \rightarrow 1^-$ et il existe t_1 tel que $\forall t \in]0, t_1]$, $e^{-t} \in [1/2, 1[$. La question 9 indique que pour $t \in]0, t_1]$,

$$|h(t, \theta)| \leq \exp\left(-\frac{1 - \cos(\theta)}{6(1 - e^{-t})^3}\right) \quad \text{ou} \quad \exp\left(-\frac{1}{3(1 - e^{-t})}\right)$$

Comme $e^u \geq 1 + u$ pour tout u par convexité de $\exp$ (courbe au dessus de sa tangente au point d'abscisse 0), on a donc

$$|h(t, \theta)| \leq \exp\left(-\frac{1 - \cos(\theta)}{6t^3}\right) \quad \text{ou} \quad \exp\left(-\frac{1}{3t}\right)$$

et avec le début de la question, en se limitant à $\theta \in [-\pi, \pi]$,

$$|h(t, \theta)| \leq \exp\left(-\frac{\alpha\theta^2}{6t^3}\right) \quad \text{ou} \quad \exp\left(-\frac{|\theta|^{2/3}}{3\pi^{2/3}t}\right)$$

Avec la question 24 et l'équivalent trouvé pour σ_t , il existe $t_0 \leq t_1$ tel que

$$\forall t \in]0, t_0], \quad \frac{\pi}{2\sqrt{3}t^{3/2}} \leq \sigma_t \leq \frac{2\pi}{\sqrt{3}t^{3/2}}$$

Ceci donne des constantes β et γ telles que

$$\forall t \in]0, t_0], \quad \beta\sigma_t^2 \leq \frac{\alpha}{6t^3} \quad \text{et} \quad \gamma\sigma_t^{2/3} \leq \frac{1}{3\pi^{2/3}t}$$

et donne le résultat voulu :

$$\forall t \in]0, t_0], \quad |h(t, \theta)| \leq e^{-\beta(\sigma_t\theta)^2} \quad \text{ou} \quad e^{-\gamma(\sigma_t|\theta|)^{2/3}}$$

26. On a

$$\int_{-\pi\sigma_t}^{\pi\sigma_t} j(t, u) \, du = \int_{\mathbb{R}} G(t, u) \, du \quad \text{avec} \quad G(t, u) = \zeta(t, u)h(t, \frac{u}{\sigma_t})\mathbf{1}_{[-\pi\sigma_t, \pi\sigma_t]}$$

On cherche dans cette question à intervertir une limite et une intégrale. On fait pour cela appel au théorème de convergence dominée.

- Pour tout $u \in \mathbb{R}$, si t est assez petit, $u \in [-\pi\sigma_t, \pi\sigma_t]$ et donc $G(t, u) = j(t, u) \rightarrow e^{-u^2/2}$.
- Pour tous $t > 0$ et $u \in [-\pi\sigma_t, \pi\sigma_t]$, $|G(t, u)| \leq |h(t, \frac{u}{\sigma_t})| \leq e^{-\beta u^2} + e^{-\gamma|u|^{2/3}}$ car $|u/\sigma_t| \leq \pi$ et on peut utiliser la question 25. Le majorant reste valable si $|u| > \pi\sigma_t$. C'est une fonction intégrable sur $\mathbb{R}$ (continu et dominé par $1/u^2$ aux voisinages des infinis).

Le théorème s'applique et donne $\int_{\mathbb{R}} e^{-u^2/2} \, du$ comme limite pour l'intégrale quand $t \rightarrow 0^+$. On a montré que

$$\lim_{t \rightarrow 0^+} \int_{-\pi\sigma_t}^{\pi\sigma_t} j(t, u) \, du = \sqrt{2\pi}$$

Le problème avec l'expression initiale à étudier est que la variable t apparaît à la fois dans les bornes et sous l'intégrale. Or, le cours ne permet de conclure (sous de bonnes hypothèses) que si la variable est dans les bornes (on a alors une étude d'existence d'intégrale) ou dans l'intégrale (théorème d'interversion). La méthode utilisée ci-dessus et qui permet de se débarrasser de la variable t dans les bornes est assez courante et gagne à être connue.

G. La conclusion

27. La formule (1) avec $t = \frac{\pi}{\sqrt{6n}}$ donne

$$p_n = \frac{e^{\pi\sqrt{\frac{n}{6}}} P(e^{-\frac{\pi}{\sqrt{6n}}})}{2\pi} \int_{-\pi}^{\pi} e^{im_t\theta} e^{-in\theta} h\left(\frac{\pi}{\sqrt{6n}}, \theta\right) d\theta$$

Le changement de variable $u = \sigma_t \theta$ donne, après simplification,

$$p_n = \frac{e^{\pi\sqrt{\frac{n}{6}}} P(e^{-\frac{\pi}{\sqrt{6n}}})}{2\pi\sigma_t} \int_{-\pi\sigma_t}^{\pi\sigma_t} j\left(\frac{\pi}{\sqrt{6n}}, u\right) du$$

Le terme σ_t (qui dépend de n) équivaut quand $n \rightarrow +\infty$ à $\frac{\pi}{\sqrt{3}} \left(\frac{\pi}{\sqrt{6n}}\right)^{-3/2} = \frac{(6n)^{3/4}}{\sqrt{3\pi}}$.

Le terme $P(e^{-\frac{\pi}{\sqrt{6n}}})$ équivaut (l'énoncé l'admet) à $\frac{1}{\sqrt{2}(6n)^{1/4}} \exp(\pi\sqrt{n}/\sqrt{6})$.

Il reste à tout multiplier pour obtenir

$$p_n \sim \frac{\exp(\pi\sqrt{\frac{2n}{3}})}{4\sqrt{3n}}$$

Remarque : dans cette ultime question de synthèse, on pourrait développer les calculs. Il ne semble pas, cependant, que de tels détails soient utiles aux yeux du correcteur.



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2022

DEUXIÈME ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 4 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES II - MP

L'énoncé de cette épreuve comporte 6 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Autour des exponentielles de matrices

Dans tout le sujet, le corps $\mathbf{K}$ sera $\mathbf{R}$ ou $\mathbf{C}$, et n est un entier naturel supérieur ou égal à 2.

On note $\|\cdot\|$ une norme sur l'espace vectoriel $\mathcal{M}_n(\mathbf{K})$, vérifiant les propriétés

$$\|I_n\| = 1. \quad (N_1)$$

$$\forall (A, B) \in (\mathcal{M}_n(\mathbf{K}))^2 \quad \|AB\| \leq \|A\| \|B\|. \quad (N_2)$$

On rappelle que l'exponentielle d'une matrice A de $\mathcal{M}_n(\mathbf{K})$ est la matrice, notée e^A , ou bien $\exp(A)$, définie par

$$e^A = \sum_{k=0}^{+\infty} \frac{A^k}{k!}.$$

On rappelle que, pour tout $A \in \mathcal{M}_n(\mathbf{K})$, l'application

$$f_A : \mathbf{R} \rightarrow \mathcal{M}_n(\mathbf{K}), \quad t \mapsto f_A(t) = e^{tA}$$

est de classe $\mathcal{C}^1$ sur $\mathbf{R}$, avec

$$\forall t \in \mathbf{R} \quad f'_A(t) = A e^{tA} = e^{tA} A.$$

On admettra que, si A et B sont deux matrices semblables de $\mathcal{M}_n(\mathbf{K})$, plus précisément si on a $B = P^{-1}AP$ avec $P \in \mathrm{GL}_n(\mathbf{K})$, alors

$$e^B = P^{-1} e^A P.$$

Si A et B sont deux matrices de $\mathcal{M}_n(\mathbf{K})$, on définit leur **crochet de Lie** par

$$[A, B] = AB - BA.$$

La partie 4 du problème est indépendante des parties 2 et 3.

1 Questions préliminaires

On se donne deux matrices A et B dans $\mathcal{M}_n(\mathbf{K})$. On suppose dans les questions 1) et 2) que A et B commutent.

1 ▷ Montrer que les matrices A et e^B commutent.

On définit une application

$$\begin{aligned} g : \mathbf{R} &\rightarrow \mathcal{M}_n(\mathbf{K}) \\ t &\longmapsto g(t) = e^{t(A+B)} e^{-tB}. \end{aligned}$$

2 ▷ Montrer que l'application g , et l'application f_A définie en préambule, sont solutions d'un même problème de Cauchy. En déduire une démonstration de la relation

$$\forall t \in \mathbf{R} \quad e^{t(A+B)} = e^{tA} e^{tB}. \quad (1)$$

3 ▷ Réciproquement, on suppose la relation (1) satisfaite. En dérivant deux fois cette relation par rapport à la variable réelle t , montrer que les matrices A et B commutent.

4 ▷ Pour toute matrice $A \in \mathcal{M}_n(\mathbf{K})$, prouver la relation $\|e^A\| \leq e^{\|A\|}$.

5 ▷ Montrer que $\det(e^A) = e^{\text{tr}(A)}$.

2 Formule de Trotter-Kato

Dans cette partie, on note A et B deux matrices quelconques de $\mathcal{M}_n(\mathbf{K})$. L'objectif est de prouver la relation

$$\lim_{k \rightarrow +\infty} \left(e^{\frac{A}{k}} e^{\frac{B}{k}} \right)^k = e^{A+B} \quad \text{ou} \quad \lim_{k \rightarrow +\infty} \left(\exp\left(\frac{A}{k}\right) \exp\left(\frac{B}{k}\right) \right)^k = \exp(A+B). \quad (2)$$

Pour tout k entier naturel non nul, on pose

$$X_k = \exp\left(\frac{A}{k}\right) \exp\left(\frac{B}{k}\right) \quad \text{et} \quad Y_k = \exp\left(\frac{A+B}{k}\right).$$

6 ▷ Prouver les majorations

$$\forall k \in \mathbf{N}^* \quad \|X_k\| \leq \exp\left(\frac{\|A\| + \|B\|}{k}\right) \quad \text{et} \quad \|Y_k\| \leq \exp\left(\frac{\|A\| + \|B\|}{k}\right).$$

On introduit la fonction

$$\begin{aligned} h : \mathbf{R} &\longrightarrow \mathcal{M}_n(\mathbf{K}) \\ t &\longmapsto h(t) = e^{tA} e^{tB} - e^{t(A+B)} \end{aligned}$$

7 ▷ Montrer que

$$X_k - Y_k = O\left(\frac{1}{k^2}\right) \text{ lorsque } k \rightarrow +\infty.$$

8 ▷ Vérifier la relation

$$X_k^k - Y_k^k = \sum_{i=0}^{k-1} X_k^i (X_k - Y_k) Y_k^{k-i-1}.$$

En déduire la relation (2).

3 Vers les algèbres de Lie

Dans cette partie, $\mathbf{K} = \mathbf{R}$. Pour tout n entier naturel, $n \geq 2$, on introduit l'ensemble, dit **groupe spécial linéaire** :

$$\mathrm{SL}_n(\mathbf{R}) = \{M \in \mathcal{M}_n(\mathbf{R}) \mid \det(M) = 1\}.$$

Si G est un sous-groupe fermé de $\mathrm{GL}_n(\mathbf{R})$, on introduit son **algèbre de Lie** :

$$\mathcal{A}_G = \{M \in \mathcal{M}_n(\mathbf{R}) \mid \forall t \in \mathbf{R} \quad e^{tM} \in G\}.$$

L'ensemble $\mathrm{SL}_n(\mathbf{R})$, ainsi que le groupe orthogonal $\mathrm{O}_n(\mathbf{R})$, sont bien des sous-groupes fermés de $\mathrm{GL}_n(\mathbf{R})$. On ne demande pas de le démontrer.

9 ▷ Déterminer $\mathcal{A}_G$ lorsque $G = \mathrm{SL}_n(\mathbf{R})$.

10 ▷ Si $G = \mathrm{O}_n(\mathbf{R})$, montrer que $\mathcal{A}_G = \mathcal{A}_n(\mathbf{R})$, ensemble des matrices antisymétriques.

Dans les questions 11) à 14), G est un sous-groupe fermé quelconque de $\mathrm{GL}_n(\mathbf{R})$.

11 ▷ En utilisant la partie 2, montrer que $\mathcal{A}_G$ est un sous-espace vectoriel de $\mathcal{M}_n(\mathbf{R})$.

12 ▷ Soient $A \in \mathcal{A}_G$ et $B \in \mathcal{A}_G$. Montrer que l'application

$$\begin{aligned} u : \mathbf{R} &\longrightarrow \mathcal{M}_n(\mathbf{R}) \\ t &\longmapsto u(t) = e^{tA} \cdot B \cdot e^{-tA} \end{aligned}$$

est à valeurs dans $\mathcal{A}_G$.

13 ▷ En déduire que $\mathcal{A}_G$ est stable par le crochet de Lie, i.e.

$$\forall A \in \mathcal{A}_G, \forall B \in \mathcal{A}_G, [A, B] \in \mathcal{A}_G.$$

On rappelle que, si M est une matrice de $\mathcal{M}_n(\mathbf{R})$, on dit que M est **tangente** à G en I_n s'il existe $\varepsilon > 0$ et une application $\gamma :]-\varepsilon, \varepsilon[\rightarrow G$, dérivable, telle que $\gamma(0) = I_n$ et $\gamma'(0) = M$. L'ensemble des matrices tangentes à G en I_n est appelé **espace tangent** à G en I_n , et noté $\mathcal{T}_{I_n}(G)$.

On rappelle aussi que l'application $\det : \mathcal{M}_n(\mathbf{R}) \rightarrow \mathbf{R}$ est différentiable en tout point, par exemple parce qu'elle est polynomiale.

14 ▷ Prouver l'inclusion $\mathcal{A}_G \subset \mathcal{T}_{I_n}(G)$.

15 ▷ Soit $M \in \mathcal{M}_n(\mathbf{R})$, que l'on pourra aussi considérer comme matrice complexe, soit l'application $\delta_M : \mathbf{R} \rightarrow \mathbf{R}, t \mapsto \delta_M(t) = \det(I_n + tM)$. En utilisant un développement limité à l'ordre 1, montrer que δ_M est dérivable en 0 et calculer $\delta'_M(0)$.

16 ▷ Montrer que la différentielle au point I_n de l'application $\det : \mathcal{M}_n(\mathbf{R}) \rightarrow \mathbf{R}$ est la forme linéaire "trace".

17 ▷ Montrer que, dans les cas particuliers $G = \mathrm{SL}_n(\mathbf{R})$ et $G = \mathrm{O}_n(\mathbf{R})$, on a $\mathcal{T}_{I_n}(G) = \mathcal{A}_G$.

4 Comportement asymptotique

Étude d'un exemple

On considère deux nombres complexes distincts α et β . On suppose qu'une matrice $A \in \mathcal{M}_3(\mathbf{C})$ admet α pour valeur propre simple, β pour valeur propre double.

18 ▷ Montrer que A est semblable à une matrice de la forme

$$T = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & a \\ 0 & 0 & \beta \end{pmatrix}$$

où a est un certain nombre complexe. Calculer T^n pour n entier naturel, puis e^{tT} pour t réel. En déduire une condition nécessaire et suffisante sur α et β pour que l'on ait $\lim_{t \rightarrow +\infty} e^{tA} = 0_3$.

Cas général

Dans tout ce qui suit, $\mathbf{K} = \mathbf{C}$. On pose $E = \mathbf{C}^n$. L'espace vectoriel E , identifié à $\mathcal{M}_{n,1}(\mathbf{C})$, peut être muni d'une quelconque norme notée $\|\cdot\|_E$, on rappelle qu'elles sont toutes équivalentes. On se donne $A \in \mathcal{M}_n(\mathbf{C})$ une matrice carrée à coefficients complexes, et on note u l'endomorphisme de $\mathbf{C}^n$ canoniquement associé à cette matrice. On s'intéresse au comportement asymptotique de la fonction f_A introduite dans le préambule, et à celui des fonctions vectorielles solutions du système différentiel linéaire à coefficients constants $X' = AX$. Pour tout t réel et pour $(i, j) \in \llbracket 1, n \rrbracket^2$, on notera $v_{i,j}(t)$ le coefficient d'indices (i, j) de la matrice e^{tA} . Ainsi,

$$\forall t \in \mathbf{R} \quad f_A(t) = e^{tA} = (v_{i,j}(t))_{1 \leq i, j \leq n} \in \mathcal{M}_n(\mathbf{C}).$$

Pour toute valeur propre λ de la matrice A , on note m_λ sa multiplicité, et on introduit le sous-espace vectoriel

$$F_\lambda = \text{Ker}((A - \lambda I_n)^{m_\lambda}) = \text{Ker}((u - \lambda \text{Id}_E)^{m_\lambda}).$$

On posera aussi $\alpha = \max_{\lambda \in \text{Sp}(A)} \text{Re}(\lambda)$.

19 ▷ Montrer que, si $\lim_{t \rightarrow +\infty} f_A(t) = 0_n$, alors $\alpha < 0$.

20 ▷ Montrer que $\mathbf{C}^n = \bigoplus_{\lambda \in \text{Sp}(A)} F_\lambda$.

21 ▷ En déduire l'existence de trois matrices P , D et N dans $\mathcal{M}_n(\mathbf{C})$ telles que :

$$\begin{aligned} P &\text{ est inversible,} \\ D &\text{ est diagonale,} \\ N &\text{ est nilpotente,} \\ ND &= DN, \\ A &= P(D + N)P^{-1}, \\ \chi_A &= \chi_D. \end{aligned}$$

22 ▷ En déduire qu'il existe un entier naturel p tel que, pour tout $(i, j) \in \llbracket 1, n \rrbracket^2$, on ait

$$v_{i,j}(t) = O(t^p e^{\alpha t}) \text{ lorsque } t \rightarrow +\infty.$$

23 ▷ Étudier la réciproque de la question 19).

24 ▷ On suppose, *dans cette question seulement*, que les valeurs propres de la matrice A ont toutes des parties réelles positives ou nulles. Montrer que, si $X \in \mathbf{C}^n$, on a

$$\lim_{t \rightarrow +\infty} e^{tA} X = 0 \iff X = 0.$$

Dans les questions qui suivent, on introduit les polynômes suivants :

$$P_s(X) = \prod_{\substack{\lambda \in \text{Sp}(A) \\ \text{Re}(\lambda) < 0}} (X - \lambda)^{m_\lambda},$$

$$P_i(X) = \prod_{\substack{\lambda \in \text{Sp}(A) \\ \text{Re}(\lambda) > 0}} (X - \lambda)^{m_\lambda},$$

$$P_n(X) = \prod_{\substack{\lambda \in \text{Sp}(A) \\ \text{Re}(\lambda) = 0}} (X - \lambda)^{m_\lambda},$$

et les sous-espaces $E_s = \text{Ker}(P_s(A))$, $E_i = \text{Ker}(P_i(A))$ et $E_n = \text{Ker}(P_n(A))$ de $E = \mathbf{C}^n$. Les indices s, i, n signifient respectivement *stable, instable et neutre*.

25 ▷ Après avoir justifié que $E = E_s \oplus E_i \oplus E_n$, montrer que

$$E_s = \{X \in E \mid \lim_{t \rightarrow +\infty} e^{tA} X = 0\}.$$

On prouverait de même, mais ce n'est pas demandé, que

$$E_i = \{X \in E \mid \lim_{t \rightarrow -\infty} e^{tA} X = 0\}.$$

26 ▷ Montrer que

$$E_n = \{X \in E \mid \exists C \in \mathbf{R}_+^* \exists p \in \mathbf{N} \forall t \in \mathbf{R} \quad \|e^{tA} X\|_E \leq C (1 + |t|)^p\}.$$

E_n est donc l'ensemble des vecteurs X de $\mathbf{C}^n$ tels que la fonction vectorielle $t \mapsto e^{tA} X$ ait un comportement polynomial en $-\infty$ et $+\infty$.

FIN DU PROBLÈME

SOLUTION

Questions préliminaires

1. Comme A et B commutent, A commute avec tout polynôme en B . En particulier

$$\forall p \in \mathbb{N}, A \sum_{k=0}^p \frac{B^k}{k!} = \sum_{k=0}^p \frac{B^k}{k!} A$$

Les applications $M \mapsto MA$ et $M \mapsto AM$ étant continues (linéaires en dimension finie), le passage à la limite $p \rightarrow +\infty$ donne $Ae^B = e^B A$.

$$\boxed{\text{Si } AB = BA \text{ alors } Ae^B = e^B A}$$

Remarque : c'est le début du sujet et il faut absolument donner tous les arguments pour mettre le correcteur en confiance. Ici, par exemple, l'argument de continuité. Le rapport indique clairement qu'un calcul formel (avec des sommes infinies) et sans argument topologique (la continuité) n'était pas rémunéré. Le fait que si A et B commutent alors A commute avec $P(B)$ pour tout $P \in \mathbb{K}[X]$ pourrait être détaillé en commençant par le cas $P = X^k$ puis par un argument de linéarité. Il me semble cependant que citer ce résultat devrait suffire.

2. g est dérivable sur $\mathbb{R}$ par théorèmes d'opération (ou par composition de $(M, N) \mapsto MN$ qui est bilinéaire en dimension finie et des fonctions f et g). De plus

$$g'(t) = (A + B)e^{t(A+B)}e^{-tB} + e^{t(A+B)}(-B)e^{-tB}$$

Comme B et $t(A + B)$ commutent, il en est de même de B et $e^{t(A+B)}$ et donc

$$g'(t) = (A + B)e^{t(A+B)}e^{-tB} - Be^{t(A+B)}e^{-tB} = Ag(t)$$

Ceci est un système différentiel dont l'inconnue est $t \in \mathbb{R} \mapsto g(t) \in \mathcal{M}_n(\mathbb{K})$. Ce système est linéaire et à coefficients constants puisqu'il s'écrit

$$g'(t) = \varphi(g(t)) \text{ avec } \varphi : A \mapsto AM$$

et que φ est une application linéaire qui ne dépend pas de t .

De plus $g(0) = I_n$ et g est en fait la solution d'un problème de Cauchy. Le préambule indique que f_A est une autre solution de ce problème de Cauchy.

Un problème de Cauchy pour une équation linéaire à coefficients continus (ici constants) admettant une unique solution, on a $f_A = g$ i.e.

$$\forall t \in \mathbb{R}, e^{t(A+B)}e^{-tB} = e^{tA}$$

On peut appliquer cela avec $A = 0$ (qui commute avec B) pour obtenir $e^{tB}e^{-tB} = I_n$ i.e. e^{tB} et e^{-tB} sont inverses l'une de l'autre. En multipliant à droite par e^{tB} , on trouve que

$$\boxed{\text{Si } AB = BA, \text{ alors } \forall t, e^{t(A+B)} = e^{tA}e^{tB}}$$

Remarque : cette question est un peu piégeante car on ne peut a priori pas utiliser le fait que si deux matrices M et N commutent alors $\exp(M)\exp(N) = \exp(M+N)$ puisque c'est l'objet de la question ! Il faut donc réussir à montrer que $\exp(-M)$ est l'inverse de $\exp(M)$ autrement qu'en faisant le produit. Par ailleurs, il faut bien comprendre de quel problème de Cauchy on parle. Dans le cours, classiquement, on a un système $X'(t) = AX(t)$ où $X : t \in \mathbb{R} \mapsto X(t) \in \mathbb{K}^n$ et la solution générale est $X : t \mapsto e^{tA}X_0$. Ce n'est pas le cas ici car l'inconnue n'est PAS à valeurs dans $\mathbb{K}^n$.

3. On peut dériver la relation (1) avec les mêmes arguments qu'en question 2 pour obtenir

$$\forall t \in \mathbb{R}, (A+B)e^{t(A+B)} = Ae^{tA}e^{tB} + e^{tA}Be^{tB}$$

et en redérivant

$$\forall t \in \mathbb{R}, (A+B)^2 e^{t(A+B)} = A^2 e^{tA}e^{tB} + Ae^{tA}Be^{tB} + Ae^{tA}Be^{tB} + e^{tA}B^2 e^{tB}$$

Pour $t = 0$, la relation donne

$$A^2 + AB + BA + B^2 = A^2 + 2AB + B^2$$

et ainsi $AB = BA$.

Si pour tout t on a $e^{t(A+B)} = e^{tA}e^{tB}$, alors $AB = BA$

4. On montre par récurrence sur k que la propriété

$$\|A^k\| \leq \|A\|^k$$

est vraie pour tout k .

- Initialisation : le résultat est vrai au rang 0 car $\|A^0\| = \|I_n\| = 1 = \|A\|^0$.
- Hérédité : supposons le résultat vrai jusqu'à un rang $k \geq 0$ (récurrence forte en réalité inutile ici). On a alors

$$\|A^{k+1}\| = \|A^k \cdot A\| \leq \|A^k\| \|A\| \leq \|A\|^k \|A\| = \|A\|^{k+1}$$

et le résultat est vrai au rang $k+1$.

Avec l'inégalité triangulaire, on a donc

$$\forall p \in \mathbb{N}, \left\| \sum_{k=0}^p \frac{A^k}{k!} \right\| \leq \sum_{k=0}^p \left\| \frac{A^k}{k!} \right\| \leq \sum_{k=0}^p \frac{\|A\|^k}{k!} \leq e^{\|A\|}$$

Comme $M \mapsto \|M\|$ est continue, faire tendre p vers $+\infty$ donne

$\forall A \in \mathcal{M}_n(\mathbb{K}), \|e^A\| \leq e^{\|A\|}$

Remarque : on ne peut pas directement travailler avec des sommes infinies à ce niveau du problème. Et il faut encore bien donner l'argument de continuité pour le passage à la limite. On notera qu'on utilise ici la propriété (N_1) . Souvent on considère des normes ayant seulement la propriété (N_2) de sous-multiplicativité, ce qui ne suffit pas ici (la formule ne serait vraie que pour $k \geq 1$ sans (N_1)).

5. Soit $A \in \mathcal{M}_n(\mathbb{K})$. A est $\mathbb{C}$ -trigonalisable et il existe $P \in GL_n(\mathbb{C})$ telle que $P^{-1}AP = T$ est triangulaire supérieure. On a alors $e^A = Pe^T P^{-1}$ et, le déterminant étant invariant par similitude,

$$\det(e^A) = \det(e^T)$$

Comme T est triangulaire supérieure, il en est de même de toutes ses puissances et donc, l'ensemble des matrices triangulaires étant un fermé puisque c'est un sous-espace de dimension finie, e^T est triangulaire et son déterminant est le produit de ses coefficients diagonaux.

Or, les coefficients diagonaux de T^k sont les puissance k -ièmes de ceux de T et donc ceux de $\exp(T)$ sont les exponentielles de ceux de T :

$$\det(e^T) = \prod_{k=1}^n e^{T_{i,i}} = \exp\left(\sum_{k=1}^n T_{i,i}\right) = e^{\text{Tr}(T)}$$

Enfin, la trace est un invariant de similitude et donc

$$\boxed{\forall A \in \mathcal{M}_n(\mathbb{K}), \det(e^A) = e^{\text{Tr}(A)}}$$

Remarque : le raisonnement fait ici utilise de manière cruciale le fait que le corps $\mathbb{K}$ est inclus dans $\mathbb{C}$. Pour le généraliser à un corps quelconque, il faudrait savoir que tout corps est inclus dans un autre "algébriquement clos".

Formule de Trotter-Kato

6. Par sous-multiplicativité de la norme (propriété (N_2)) et question 4, on a

$$\|X_k\| \leq \|e^{\frac{A}{k}}\| \|e^{\frac{B}{k}}\| \leq e^{\frac{\|A\|}{k}} e^{\frac{\|B\|}{k}}$$

Il reste à utiliser la propriété de morphisme de l'exponentielle (d'une variable réelle ici) pour conclure que

$$\boxed{\forall k \in \mathbb{N}, \|X_k\| \leq \exp\left(\frac{\|A\| + \|B\|}{k}\right)}$$

Pour Y_k , on utilise d'abord la question 4 puis l'inégalité triangulaire et la croissance de $\exp$ (considérée comme fonction d'une variable réelle) :

$$\|Y_k\| \leq e^{\frac{\|A+B\|}{k}} \leq e^{\frac{\|A\| + \|B\|}{k}}$$

$$\boxed{\forall k \in \mathbb{N}, \|Y_k\| \leq \exp\left(\frac{\|A\| + \|B\|}{k}\right)}$$

7. Avec les notations de l'énoncé, on a $X_k - Y_k = h(\frac{1}{k})$. Comme h est de classe C^2 sur $\mathbb{R}$, on a

$$h(x) = h(0) + xh'(0) + \int_0^x (x-t)h''(t) dt$$

Il s'agit de la formule de Taylor avec reste intégrale, valable pour des fonctions d'une variable réelle et à valeurs dans un espace vectoriel de dimension finie en travaillant avec

les fonctions coordonnées.

On a $h(0) = 0$. Comme plus haut, $h'(t) = Ae^{tA}e^{tB} + e^{tA}Be^{tB} - (A+B)e^{t(A+B)}$ et donc $h'(0) = 0$. Ainsi

$$X_k - Y_k = h\left(\frac{1}{k}\right) = \int_0^{1/k} \left(\frac{1}{k} - t\right) h''(t) dt$$

et donc

$$\|X_k - Y_k\| \leq \int_0^{1/k} \left(\frac{1}{k} - t\right) \|h''(t)\| dt$$

Quand f est une fonction continue sur $[a, b]$ et à valeur dans l'espace vectoriel normé $(E, \|\cdot\|)$, on a $\left\|\int_a^b f\right\| \leq \int_a^b \|f\|$.

h'' est de classe C^1 sur le compact $[0, 1]$ et est bornée sur ce compact. En notant M un majorant de la norme, on a

$$\|X_k - Y_k\| \leq M \int_0^{1/k} \left(\frac{1}{k} - t\right) dt = \frac{M}{2k^2}$$

ce qui montre que

$$X_k - Y_k = O\left(\frac{1}{k^2}\right) \text{ quand } k \rightarrow +\infty$$

Remarque : pour traiter cette question, il faut savoir ce que signifie la notation "grand O " dans le cas de vecteurs. C'est simplement la même chose que les normes des quantités (de même que pour des quantités réelles, $f(t) = O(g(t))$ et $|f(t)| = O(|g(t)|)$ sont des phrases équivalentes).

8. On a

$$\begin{aligned} \sum_{i=0}^{k-1} X_k^i (X_k - Y_k) Y_k^{k-i-1} &= \sum_{i=0}^{k-1} X_k^{i+1} Y_k^{k-i-1} - \sum_{i=0}^{k-1} X_k^i Y_k^{k-i} \\ &= \sum_{j=1}^k X_k^j Y_k^{k-j} - \sum_{i=0}^{k-1} X_k^i Y_k^{k-i} \end{aligned}$$

par un simple changement d'indice. Les termes se télescopent et il reste

$$X_k^k - Y_k^k = \sum_{i=0}^{k-1} X_k^i (X_k - Y_k) Y_k^{k-i-1}$$

On a tout d'abord $(e^M)^k = e^{kM}$ car M commute avec elle-même. Ainsi, $Y_k^k = e^{A+B}$. Avec le début de la question, on a donc (on utilise l'inégalité triangulaire et le caractère sous-multiplicatif de la norme)

$$\|X_k^k - e^{A+B}\| \leq \sum_{i=0}^{k-1} \|X_k^i (X_k - Y_k) Y_k^{k-i-1}\| \leq \sum_{i=0}^{k-1} \|X_k\|^i \|X_k - Y_k\| \|Y_k\|^{k-i-1}$$

Avec la question 6, on a donc $((e^{x/k})^{k-1} = e^{\frac{x(k-1)}{k}} \leq e^x$ quand $x \geq 0$)

$$\|X_k^k - e^{A+B}\| \leq \|X_k - Y_k\| \sum_{i=0}^{k-1} (e^{\frac{\|A\|+\|B\|}{k}})^{k-1} \leq k\|X_k - Y_k\| e^{\|A\|+\|B\|}$$

La question 7 indique que le majorant est de limite nulle et on a prouvé que

$$\lim_{k \rightarrow +\infty} \left(e^{\frac{A}{k}} e^{\frac{B}{k}} \right)^k = e^{A+B}$$

On prendra soin de ne pas écrire une limite avant de savoir si celles-ci existe.

Vers les algèbres de Lie

9. Il s'agit de trouver les $M \in \mathcal{M}_n(\mathbb{R})$ telles que e^{tM} est de déterminant 1 pour tout t . D'après la question 5, ceci équivaut à la nullité de la trace de tM pour tout t . Si cette propriété a lieu alors $t = 1$ donne M de trace nulle. Réciproquement, si M est de trace nulle, c'est le cas de tM pour tout t par linéarité de la trace. Ainsi

$$\mathcal{A}_{SL_n(\mathbb{R})} = \{M \in \mathcal{M}_n(\mathbb{R}), \text{Tr}(M) = 0\} = \ker(\text{Tr})$$

10. On procède par double inclusion.

- Supposons $e^{tM} \in O_n(\mathbb{R})$ pour tout t . On a donc

$$\forall t \in \mathbb{R}, (e^{tM})^T e^{tM} = I_n$$

Comme pour tout k , $(M^k)^T = (M^T)^k$, on a $(e^M)^T = e^{M^T}$ (égalité des sommes partielles et continuité du passage à la transposée) et donc

$$\forall t \in \mathbb{R}, e^{tM^T} e^{tM} = I_n$$

En dérivant cette relation, on trouve

$$\forall t \in \mathbb{R}, M^T e^{tM^T} e^{tM} + e^{tM^T} M e^{tM} = 0$$

En appliquant en $t = 0$, on trouve que M est antisymétrique.

- Réciproquement, on suppose M antisymétrique. On a alors

$$\forall t \in \mathbb{R}, (e^{tM})^T e^{tM} = e^{tM^T} e^{tM} = e^{-tM} e^{tM}$$

Comme M et $-M$ commutent ceci vaut $e^0 = I_n$ et $e^{tM} \in O_n(\mathbb{R})$.

$$\mathcal{A}_{O_n(\mathbb{R})} = \mathcal{A}_n(\mathbb{R})$$

11. Comme G est un sous-groupe de $GL_n(\mathbb{R})$, il contient I_n et donc $0 \in \mathcal{A}_G$. Soient $A, B \in \mathcal{A}_G$. On a donc e^{tA} et e^{tB} qui sont dans G pour tout réel t . En particulier, c'est vrai pour t/k ($k \in \mathbb{N}^*$ et $t \in \mathbb{R}$) et comme G est un groupe

$$\forall t \in \mathbb{R}, \forall k \in \mathbb{N}^*, \left(e^{\frac{tA}{k}} e^{\frac{tB}{k}} \right)^k \in G$$

Comme G est fermé, une limite de suite d'éléments de G est dans G et la formule de Trotter-Kano donne $e^{tA+tB} \in G$ et donc $A+B \in \mathcal{A}_G$.

Soient enfin $A \in \mathcal{A}_G$ et $\lambda \in \mathbb{R}$. On a $e^{t(\lambda A)} = e^{(t\lambda)A} \in G$ et donc $\lambda A \in \mathcal{A}_G$.

Par caractérisation des sous-espaces vectoriels,

$$\boxed{\mathcal{A}_G \text{ est un sous-espace vectoriel de } \mathcal{M}_n(\mathbb{R})}$$

Dans ce type de question, il ne faut oublier aucune des hypothèses, en particulier le fait que l'ensemble est non vide.

12. On suppose $A, B \in \mathcal{A}_G$ et $t \in \mathbb{R}$. Soit $x \in \mathbb{R}$. Comme e^{tA} et e^{-tA} sont des matrices inverses, on a (avec la formule $P^{-1}e^M P = e^{P^{-1}MP}$)

$$e^{xu(t)} = e^{tA} e^{xB} e^{-tA}$$

qui est un produit d'éléments de G et est donc dans G . Ainsi, $u(t) \in \mathcal{A}_G$ et

$$\boxed{u \text{ est à valeurs dans } \mathcal{A}_G}$$

Il ne faut pas oublier d'introduire un paramètre réel x pour suivre la définition de l'algèbre de Lie.

13. On se donne $A, B \in \mathcal{A}_G$ et on utilise la fonction u de la question précédente. On remarque que u est dérivable et que

$$\forall t \in \mathbb{R}, u'(t) = Ae^{tA}Be^{-tA} - e^{tA}BAe^{-tA} = e^{tA}[A, B]e^{-tA}$$

puisque A et e^{tA} commutent. Ainsi

$$\forall t \in \mathbb{R}, [A, B] = e^{-tA}u'(t)e^{tA}$$

et pour montrer que $[A, B] \in \mathcal{A}_G$, il suffit de montrer avec la question précédente (en changeant A en $-A$) que $u'(t) \in \mathcal{A}_G$.

Par définition de la dérivée,

$$u'(t) = \lim_{h \rightarrow 0} \frac{1}{h}(u(t+h) - u(t))$$

u est à valeurs dans l'espace vectoriel $\mathcal{A}_G$ et pour tous $t \in \mathbb{R}$ et $h \neq 0$, $\frac{1}{h}(u(t+h) - u(t)) \in \mathcal{A}_G$. $\mathcal{A}_G$ est un espace vectoriel de dimension finie et c'est donc un fermé. Une limite d'éléments de $\mathcal{A}_G$ est donc dans $\mathcal{A}_G$ et c'est ainsi le cas de $u'(t)$.

$$\boxed{\mathcal{A}_G \text{ est stable par crochet de Lie}}$$

Le terme "algèbre de Lie" est mal choisi car $\mathcal{A}_G$ n'est pas toujours une algèbre. On ne peut, ainsi, affirmer que $\mathcal{A}_G$ est stable par produit.

14. Soit $A \in \mathcal{A}_G$. Considérons l'application $\gamma : t \mapsto e^{tA}$. C'est une fonction à valeurs dans G (par définition de $\mathcal{A}_G$), dérivable et de dérivée $\gamma' : t \mapsto Ae^{tA}$. Ainsi $\gamma(0) = I_n$ et $\gamma'(0) = A$. On a donc montré que $A \in \mathcal{T}_{I_n}(G)$.

$$\boxed{\mathcal{A}_G \subset \mathcal{T}_{I_n}(G)}$$

15. M est $\mathbb{C}$ -trigonalisable via une matrice inversible $P : P^{-1}MP = T$ triangulaire supérieure et (déterminant invariant par similitude)

$$\delta_M(t) = \det(I_n + tT) = \prod_{k=1}^n (1 + tT_{k,k}) = 1 + t \sum_{k=1}^n T_{k,k} + o(t) \text{ quand } t \rightarrow 0$$

δ_M admet donc un DL à l'ordre en 0 ce qui donne sa dérivabilité en 0 avec

$$\delta'_M(0) = \sum_{k=1}^n T_{k,k} = \text{Tr}(T) = \text{Tr}(M)$$

Remarque : on rappelle que la dérivabilité en un point équivaut à l'existence d'un DL à l'ordre 1 en ce point. Cette équivalence est fautive pour les dérivées suivantes.

16. Comme $\det$ est différentiable en tout point, on a aussi

$$\delta'_M(t) = d\det(I_n + tM).M$$

et en particulier

$$d\det(I_n).M = \delta'_M(0) = \text{Tr}(M)$$

Ceci étant vrai pour tout M ,

$$d\det(I_n) = \text{Tr}$$

Remarque : on a ici interprété δ comme la composée de $\det$ et de $\varphi : t \mapsto I_n + tM$. La formule de différentiation composée donne $d(\det \circ \varphi)(t) = d\det(\varphi(t)) \circ d\varphi(t)$ ou en termes de dérivée (ce qui a un sens puisque la variable t est réelle), $(\det \circ \varphi)'(t) = d\det(\varphi(t))(\varphi'(t))$. Et ici $\varphi'(t) = M$.

17. On se place dans le cas où $G = SL_n(\mathbb{R})$ et on se donne $M \in \mathcal{T}_{I_n}(G)$. Il existe donc $\gamma :]-\varepsilon, \varepsilon[\rightarrow G$ associée (dérivable avec $\gamma(0) = I_n$ et $\gamma'(0) = M$). Pour tout $t \in]-\varepsilon, \varepsilon[$, on a

$$\det(\gamma(t)) = 1$$

La dérivation de cette relation donne

$$d\det(\gamma(t)).\gamma'(t) = 0$$

et en particulier, on a

$$\text{Tr}(M) = d\det(I_n).M = 0$$

ce qui montre que $M \in \mathcal{A}_{SL_n(\mathbb{R})}$. On a donc (l'autre inclusion a été prouvée en question 14)

$$\mathcal{T}_{I_n}(SL_n(\mathbb{R})) = \mathcal{A}_{SL_n(\mathbb{R})}$$

On se place dans le cas où $G = O_n(\mathbb{R})$ et on se donne $M \in \mathcal{T}_{I_n}(G)$. Il existe donc γ associée. Pour tout $t \in]-\varepsilon, \varepsilon[$, on a

$$\gamma(t)^T \gamma(t) = I_n$$

et en dérivant on trouve

$$\gamma'(t)^T \gamma(t) + \gamma(t)^T \gamma'(t) = 0$$

et en particulier (pour $t = 0$), $M^T + M = 0$ et donc $M \in \mathcal{A}_{O_n(\mathbb{R})}$. Ainsi (même remarque que ci-dessus)

$$\mathcal{T}_{I_n}(O_n(\mathbb{R})) = \mathcal{A}_{O_n(\mathbb{R})}$$

Comportement asymptotique

18. Par hypothèse, $\chi_A = (X - \alpha)(X - \beta)^2$ et ce polynôme annule A . Par théorème de décomposition des noyaux ($(X - \alpha)$ et $(X - \beta)^2$ sont premiers entre eux)

$$\mathbb{C}^3 = \ker(\chi_A(A)) = \ker(A - \alpha I_3) \oplus \ker((A - \beta I_3)^2)$$

$F_\alpha = \ker(A - \alpha I_3)$ et $F_\beta = \ker((A - \beta I_3)^2)$ sont stables par l'endomorphisme f canoniquement associé à A (comme noyaux de polynômes en f). Notons f_α et f_β les endomorphismes induits.

F_α est de dimension 1 (sous-espace propre associé à une valeur propre simple) et $f_\alpha = \alpha \text{Id}$. F_β est donc de dimension 2 et comme $(X - \beta)^2$ scindé annule f_β , il existe une base où f_β est représenté par une matrice triangulaire de taille 2 avec des β sur la diagonale (f est annulé par un polynôme scindé et donc trigonalisable et ses seules valeurs propres possibles sont les racines du polynôme).

Dans une base adaptée, f est ainsi représenté par une matrice du type T

$$A \text{ est semblable à } T = \begin{pmatrix} \alpha & 0 & 0 \\ 0 & \beta & a \\ 0 & 0 & \beta \end{pmatrix}$$

Ceci est un cas particulier d'un résultat général qui figure au programme. Pour un endomorphisme trigonalisable, les sous-espaces caractéristiques sont supplémentaires. Traduction matricielle : similitude à une matrice diagonale par blocs, chaque bloc diagonal étant triangulaire et à termes diagonaux égaux. C'est donc en fait une question de cours et, bien sûr, il n'était pas question de simplement évoquer le résultat général.

On montre facilement par récurrence que (en convenant que pour $n = 0$, $na\beta^{n-1}$ est nul)

$$\forall n \in \mathbb{N}, T^n = \begin{pmatrix} \alpha^n & 0 & 0 \\ 0 & \beta^n & na\beta^{n-1} \\ 0 & 0 & \beta^n \end{pmatrix}$$

En combinant linéairement pour calculer l'exponentielle de tT on voit apparaître les quantités

$$\sum_{k=0}^n \frac{(t\alpha)^k}{k!}, \sum_{k=0}^n \frac{(t\beta)^k}{k!}, a \sum_{k=1}^n \frac{k\beta^{k-1}t^k}{k!}$$

qui convergent vers $e^{t\alpha}$, $e^{t\beta}$ et $ate^{t\beta}$. On a donc

$$e^{tT} = \begin{pmatrix} e^{t\alpha} & 0 & 0 \\ 0 & e^{t\beta} & ate^{t\beta} \\ 0 & 0 & e^{t\beta} \end{pmatrix}$$

Supposons (étape d'analyse i.e. recherche de condition nécessaire) $e^{tA} \rightarrow 0$. Il existe P telle que $P^{-1}AP = T$ et $e^{tT} = P^{-1}e^{tA}P \rightarrow 0$ (par exemple par continuité de $M \mapsto P^{-1}MP$). Tous les coefficients de e^{tT} sont donc de limite nulle et ceci impose α et β de partie réelle < 0 ($|e^z| = e^{\text{Re}(z)}$).

Réciproquement, si cette condition a lieu alors tous les coefficients de e^{tT} sont de limite nulle quand $t \rightarrow +\infty$ (l'un d'entre eux par croissances comparées) et donc $e^{tA} = Pe^{tT}P^{-1}$ est aussi de limite nulle (multiplier par P ou P^{-1} est une opération continue).

$$\lim_{t \rightarrow +\infty} e^{tA} = 0_3 \iff (\operatorname{Re}(\alpha) < 0 \text{ et } \operatorname{Re}(\beta) < 0)$$

Une erreur fatale serait, bien sûr, d'oublier que α et β sont des complexes et d'écrire des inégalités dans $\mathbb{C}$.

19. Supposons que $f_A(t) \rightarrow 0$ quand $t \rightarrow +\infty$. Pour tout vecteur colonne X , on a donc $f_A(t)X \rightarrow 0$.

Il existe $\lambda \in \operatorname{Sp}(A)$ tel que $\alpha = \operatorname{Re}(\lambda)$. On note X un vecteur propre associé : $AX = \lambda X$. Une récurrence simple donne $A^k X = \lambda^k X$ et donc (combinaisons linéaires et passage à la limite) $e^{tA} X = e^{t\lambda} X$.

Ce vecteur est de limite nulle quand $t \rightarrow +\infty$ et donc

$$\|e^{tA} X\|_E = e^{t\alpha} \|X\|_E \rightarrow 0$$

Comme $\|X\|_E \neq 0$, on en déduit que $\alpha < 0$.

$$\text{Si } \lim_{t \rightarrow +\infty} f_A(t) = 0_n \text{ alors } \alpha < 0$$

20. On a (puisque tout polynôme est scindé sur $\mathbb{C}$ et puisque χ_A est unitaire)

$$\chi_A = \prod_{\lambda \in \operatorname{Sp}(A)} (X - \lambda)^{m_\lambda}$$

Comme χ_A annule A (théorème de Cayley-Hamilton) et que les $(X - \lambda)^{m_\lambda}$ sont deux à deux premiers entre eux, le théorème de décomposition des noyaux donne

$$\mathbb{C}^n = \bigoplus_{\lambda \in \operatorname{Sp}(A)} F_\lambda$$

Le rapport s'offusque que peu de candidats aient justifié que les $(X - \lambda)^{m_\lambda}$ sont deux à deux premiers entre eux. A ce niveau du problème, cela nous semble vraiment abusif. Que dire de plus sinon que deux de ces polynômes n'ont pas de facteur irréductible commun ? Cela ne nous semble pas apporter grand chose.

21. Notons u l'endomorphisme canoniquement associé à A . Comme u et $(u - \lambda \operatorname{Id}_E)^{m_\lambda}$ commutent, tous les F_λ sont stables par u . Dans une base $\mathcal{B}$ adaptée à la somme directe (obtenue en concaténant des bases $\mathcal{B}_\lambda$ des F_λ), on a

$$\operatorname{Mat}(u, \mathcal{B}) = \operatorname{diag}(A_\lambda)_{\lambda \in \operatorname{Sp}(A)} \text{ avec } A_\lambda = \operatorname{Mat}(u_\lambda, \mathcal{B}_\lambda)$$

où on note $u_\lambda \in \mathcal{L}(F_\lambda)$ l'endomorphisme induit par u sur F_λ . Comme $(X - \lambda)^{m_\lambda}$ annule u_λ , $u_\lambda - \lambda \operatorname{Id}_{F_\lambda}$ est nilpotent. Il existe donc une base de F_λ dans laquelle $u_\lambda - \lambda \operatorname{Id}_{F_\lambda}$ est représenté par une matrice strictement triangulaire supérieure.

Si on choisit pour $\mathcal{B}_\lambda$ la base évoquée ci-dessus, on obtient une base dans laquelle u est représenté par

$$M = \operatorname{diag}(A_\lambda)_{\lambda \in \operatorname{Sp}(A)} \text{ avec } A_\lambda = \begin{pmatrix} \lambda & ? & \dots & ? \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & ? \\ 0 & \dots & 0 & \lambda \end{pmatrix}$$

M et A représentent le même endomorphisme dans deux bases différentes et sont donc semblables via une matrice de passage P :

$$A = PMP^{-1}$$

En notant n_λ la taille du bloc A_λ , c'est à dire la dimension de F_λ , le polynôme caractéristique de M est le produit des $(X - \lambda)^{n_\lambda}$.

A et M ayant mêmes polynômes caractéristiques, on en déduit que $m_\lambda = n_\lambda$ et on peut écrire

$$A_\lambda = \lambda I_{m_\lambda} + N_\lambda$$

On peut alors poser

$$D = \text{diag}(\lambda I_{m_\lambda})_{\lambda \in \text{Sp}(A)} \quad \text{et} \quad N = \text{diag}(N_\lambda)_{\lambda \in \text{Sp}(A)}$$

ce qui nous donne une matrice diagonale et une matrice nilpotente (chaque bloc est nilpotent puisque triangulaire supérieur à diagonale nulle). Comme N_λ et λI_{m_λ} commutent, N et D commutent. Enfin, on a $\chi_A = \chi_D$.

$$A = P(D + N)P^{-1} \text{ avec } D \text{ diagonale, } N \text{ nilpotente, } DN = ND \text{ et } \chi_A = \chi_D$$

On a là la décomposition de Dunford. Elle dit qu'un endomorphisme u se décompose en la somme $u = d + n$ avec d diagonalisable et n nilpotent qui commutent. Cette décomposition est unique (ce qui n'est pas montré ici). De plus n et d sont des polynômes en u (idem). La difficulté, quand on a compris le processus, est de justifier que $m_\lambda = \dim(F_\lambda)$.

22. Comme D et N commutent, on a $e^{t(D+N)} = e^{tD}e^{tN}$. De plus

$$e^{tD} = \text{diag}(e^{t\lambda} I_{m_\lambda})_{\lambda \in \text{Sp}(A)}$$

et en notant N_∞ la norme infinie sur $\mathcal{M}_n(\mathbb{K})$ (maximum des modules des coefficients)

$$N_\infty(e^{tD}) \leq e^{t\alpha}$$

Toutes les normes étant équivalentes, il existe c tel que $\forall M, \|M\| \leq cN_\infty(M)$ et ainsi

$$\|e^{tD}\| \leq ce^{t\alpha}$$

Par ailleurs, comme N est nilpotente, on a $N^n = 0$ et donc

$$e^{tN} = \sum_{k=0}^{n-1} \frac{t^k N^k}{k!}$$

et ainsi

$$\|e^{tN}\| \leq \sum_{k=0}^{n-1} \frac{|t|^k \|N\|^k}{k!}$$

Pour $t \geq 1$ et $0 \leq k \leq n-1$, $|t|^k \leq t^{n-1}$ et donc

$$\forall t \geq 1, \|e^{tN}\| \leq t^{n-1} \sum_{k=0}^{n-1} \frac{\|N\|^k}{k!} = c' t^{n-1}$$

On a ainsi

$$\forall t \geq 1, \|e^{t(D+N)}\| \leq cc't^{n-1}e^{t\alpha}$$

De plus $e^{tA} = Pe^{t(D+N)}P^{-1}$ et donc

$$\forall t \geq 1, \|e^{tA}\| \leq cc'\|P\|\|P^{-1}\|t^{n-1}e^{\alpha t}$$

Toujours avec l'équivalence des normes, il existe d tel que $N_\infty(M) \leq d\|M\|$ pour tout M et finalement

$$\forall t \geq 1, N_\infty(e^{tA}) \leq dcc'\|P\|\|P^{-1}\|t^{n-1}e^{\alpha t} = O(t^{n-1}e^{\alpha t}) \text{ quand } t \rightarrow +\infty$$

Chaque coefficient d'une matrice étant de module plus petit que la norme infinie de la matrice, on a a fortiori

$$\boxed{\forall i, j \in \llbracket 1, n \rrbracket^2, v_{i,j}(t) = O(t^{n-1}e^{\alpha t})}$$

23. On suppose que $\alpha < 0$. Par croissances comparées $t^{n-1}e^{\alpha t} \rightarrow 0$ quand $t \rightarrow +\infty$. La question précédente montre que chaque coefficient $v_{i,j}$ est de limite nulle en $+\infty$ et il en est de même de $N_\infty(f_A(t))$. Les normes étant équivalentes, travailler avec la norme infinie est licite et on a la réciproque de la question 19.

$$\boxed{\lim_{t \rightarrow +\infty} f_A(t) = 0_n \iff \alpha < 0}$$

24. On peut d'abord remarquer que si $x = 0$ alors $e^{tA}x = 0$ est de limite nulle quand $t \rightarrow +\infty$.

Supposons que $e^{tA}x$ est de limite nulle quand $t \rightarrow +\infty$. On peut décomposer x sur la famille des F_λ et écrire

$$x = \sum_{\lambda \in \text{Sp}(A)} x_\lambda \text{ avec } x_\lambda \in F_\lambda$$

et on a

$$e^{tA}x = \sum_{\lambda \in \text{Sp}(A)} e^{tA}x_\lambda$$

Comme F_λ est stable par e^{tA} , la décomposition ci-dessus est celle sur la famille des F_λ . La projection sur un F_λ dans la direction de la somme des autres étant continue, on a alors

$$\forall \lambda \in \text{Sp}(A), e^{tA}x_\lambda \rightarrow 0$$

Remarque : on ne peut bien sûr en général pas passer du fait que la somme de quantités est de limite nulle au fait que chaque quantité l'est. L'argument de continuité donné ci-dessus est essentiel.

Fixons $\lambda \in \text{Sp}(A)$ et notons v la restriction de u à F_λ . $(X - \lambda)^{m_\lambda}$ annule v et en posant $w = v - \lambda \text{Id}_{F_\lambda}$ on a donc un endomorphisme nilpotent. La relation matricielle précédente donne

$$e^{t(\lambda \text{Id} + w)}(x_\lambda) \rightarrow 0$$

c'est à dire

$$e^{t\lambda}e^{tw}(x_\lambda) \rightarrow 0$$

Comme on suppose que les λ sont à valeurs propres positives, on a $|e^{t\lambda}| \geq 1$ pour $t \geq 0$ et ainsi (on divise par $e^{t\lambda}$ ce qui donne une quantité plus petite en norme)

$$e^{tw}(x_\lambda) \rightarrow 0$$

w est nilpotent et si, par l'absurde, $x_\lambda \neq 0$, il existe un plus grand indice $p \geq 0$ tel que $w^p(x_\lambda) \neq 0$. On a alors

$$e^{tw}(x_\lambda) = \sum_{k=0}^p \frac{t^k}{k!} w^k(x_\lambda)$$

et ainsi (pour $t \geq 0$ et par inégalité triangulaire)

$$\|e^{tw}(x_\lambda)\|_E \geq \frac{t^p}{p!} \|w^p(x_\lambda)\|_E - \sum_{k=0}^{p-1} \frac{t^k}{k!} \|w^k(x_\lambda)\|_E \underset{t \rightarrow +\infty}{\sim} \frac{t^p}{p!} \|w^p(x_\lambda)\|_E$$

Le minorant étant de limite infinie, on a une contradiction. On a donc nullité de tous les x_λ et aussi de x .

Si les valeurs propres sont toutes de partie réelle ≥ 0 , $\lim_{t \rightarrow +\infty} e^{tA}x = 0 \iff x = 0$

25. P_s, P_i et P_n sont deux à deux premiers entre eux (puisque'ils n'ont pas de racine complexe commune et puisque les irréductibles de $\mathbb{C}[X]$ sont les polynômes de degré 1) et $\chi_A = P_s P_i P_n$. Par théorème de décomposition des noyaux et puisque χ_A annule A , on a

$$E = E_s \oplus E_i \oplus E_n$$

Soit $X \in E$ que l'on décompose sur la famille précédente : $X = X_s + X_i + X_n$. Comme les trois espaces E_s, E_i, E_n sont stables par A (comme noyaux de polynômes en A) et donc aussi par e^{tA} (pour Z dans l'un des espaces $e^{tA}Z$ est encore dans l'espace considéré comme limite d'éléments de cet espace ; on rappelle qu'un sous-espace de dimension finie est fermé). On a donc

$$e^{tA}X = e^{tA}X_s + e^{tA}X_i + e^{tA}X_n$$

et ceci est la décomposition sur E_s, E_i, E_n .

La projection sur E_s de direction $E_i \oplus E_n$ étant continue, si $e^{tA}X \rightarrow 0$ alors $e^{tA}X_s \rightarrow 0$ et de même pour les deux autres. Réciproquement, si les trois $e^{tA}X_s, e^{tA}X_i, e^{tA}X_n$ sont de limite nulle, alors $e^{tA}X \rightarrow 0$.

Il faudrait en réalité travailler avec les endomorphismes induits u_s, u_i, u_n qui sont respectivement à valeurs propres de parties réelles $< 0, > 0$ ou $= 0$. La condition pour que $e^{tu}(x) \rightarrow 0$ est alors que $e^{tu_s}(x_s) \rightarrow 0$, $e^{tu_i}(x_i) \rightarrow 0$ et $e^{tu_n}(x_n) \rightarrow 0$. La première limite est toujours vraie (question 23). Les seconde et troisième ont lieu ssi $x_i = 0$ et $x_n = 0$ (question 24).

La condition est donc que $x \in E_s$.

$$E_s = \{X \in E \mid \lim_{t \rightarrow +\infty} e^{tA}X = 0\}$$

26. E_n est stable u et u induit sur E_n un endomorphisme u_n . Il est aussi stable par e^{tu} et la restriction de e^{tu} à E_n est e^{tu_n} . L'endomorphisme u_n a toutes ses valeurs propres qui sont imaginaires pures. La question 22 s'applique à cet endomorphisme avec $\alpha = 0$.

Dans cette question, on travaille matriciellement avec la norme infinie mais comme toutes les normes sont équivalentes en dimension finie, cette question montre aussi que pour $x \in E_n$, $\|e^{tu_n}(x)\|_E = O(t^p)$ quand $t \rightarrow +\infty$. Il en va de même en $-\infty$ en changeant u en $-u$. Pour un tel x , il existe un entier p et des constantes K, a telles que $\forall |t| \geq a$, $\|e^{tu_n}(x)\|_E \leq K|t|^p$. Sur le segment $[-a, a]$, la norme est bornée par une constante K' . En posant $C = \max(K, K')$ on a

$$\forall t \in \mathbb{R}, \|e^{tu}(x)\|_E \leq C(1 + |t|^p) \leq C(1 + |t|)^p$$

Réciproquement, on suppose avoir cette propriété pour x . On décompose $x = x_s + x_i + x_n$ sur $E_s \oplus E_i \oplus E_n$ et on a

$$e^{tu}(x) = e^{tu}(x_s) + e^{tu}(x_i) + e^{tu}(x_n)$$

Quand $t \rightarrow +\infty$, $\|e^{tu}(x_s)\|_E \rightarrow 0$ d'après la question 25 et le début de la question donne $\|e^{tu}(x_n)\|_E$ qui est à croissance au plus polynomiale en $+\infty$. L'hypothèse indique que $\|e^{tu}(x_i)\|_E$ est aussi à croissance au plus polynomiale en $+\infty$. En reprenant et adaptant la question 24, cela entraîne $x_i = 0$. En changeant u en $-u$, on a de même $x_s = 0$ et donc $x \in E_n$.

$$E_n = \{x \in E \mid \exists c > 0, \exists p \in \mathbb{N}^*, \forall t \in \mathbb{R}, \|e^{tA}x\|_E \leq c(1 + |t|)^p\}$$



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2022

ÉPREUVE D'INFORMATIQUE MP

Durée de l'épreuve : 3 heures

L'usage de la calculatrice et de tout dispositif électronique est interdit.

Cette épreuve concerne uniquement les candidats de la filière MP.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE - MP

L'énoncé de cette épreuve comporte 9 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Préliminaires

Présentation du sujet

L'épreuve est composée d'un problème unique, comportant 27 questions. Dans ce problème, nous étudions un mécanisme pour lire un mot qui a été brouillé (par exemple *corekte*) et proposer un mot ciblé comme correction de ce mot (par exemple *correct*). Ce type de traitement est couramment utilisé dans le contexte de la vérification orthographique, de la reconnaissance vocale, de la lecture optique ou encore de la conception de moteurs de recherche tolérant aux requêtes mal formulées.

Après cette section de préliminaires, le problème est divisé en trois sections. Dans la première section (page 1), nous étudions comment mesurer des erreurs commises à la saisie d'un mot par une méthode de programmation dynamique. Dans la deuxième section (page 4), nous étudions comment stocker un corpus de mots par une méthode arborescente et comment fouiller ce corpus à l'aide d'un automate déjà construit. Dans la troisième section (page 7), nous étudions comment construire un filtre de fouille par une méthode inspirée de la théorie des automates.

Dans tout l'énoncé, un même identificateur écrit dans deux polices de caractère différentes désignera la même entité, mais du point de vue mathématique pour la police en italique (par exemple n ou n') et du point de vue informatique pour celle en romain avec espacement fixe (par exemple `n` ou `nprime`).

Travail attendu

Pour répondre à une question, il sera permis de réutiliser le résultat d'une question antérieure, même sans avoir réussi à établir ce résultat.

Il faudra coder des fonctions à l'aide du langage de programmation OCaml, en reprenant l'en-tête de fonction fourni par le sujet, sans nécessairement recopier la déclaration des types. Quand l'énoncé demande de coder une fonction, sauf demande explicite de l'énoncé, il n'est pas nécessaire de justifier que celle-ci est correcte ou que des préconditions sont satisfaites.

Le barème tient compte de la clarté des programmes : nous recommandons de choisir des noms de variables intelligibles ou encore de structurer de longs codes par des blocs ou par des fonctions auxiliaires dont on décrit le rôle.

1 Une mesure des erreurs de saisie

1.1 Une fonction mystère

La constante entière `max_int` désigne le plus grand entier représentable par OCaml. Nous nous donnons la fonction `mystere` suivante.

```
let rec mystere z = match z with
(* La fonction mystere calcule ... *)
| [] -> max_int
| [a] -> a
| a::b::y -> mystere ((if a <= b then a else b)::y);;
```

- ☐ 1 – Donner la signature de la fonction `mystere`. Justifier brièvement.

□ 2 – Dire si, quelle que soit l'entrée z respectant le typage, le calcul de `mystere z` se termine et le démontrer. Préciser, le cas échéant, le nombre d'appels à la fonction `mystere`.

□ 3 – Compléter sommairement le commentaire de la ligne 2. Énoncer une propriété qui caractérise exactement la valeur de retour de la fonction `mystere`. Démontrer cette propriété.

Dans la question suivante, le terme *complexité en espace* désigne un ordre de grandeur asymptotique de l'espace utilisé en mémoire lors de l'exécution d'un algorithme pour stocker tant l'entrée que des résultats intermédiaires et la valeur de retour.

□ 4 – Quelle est la complexité en espace de l'appel `mystere z`? Est-elle optimale?

1.2 Distance d'édition de Levenshtein

Nous fixons un alphabet de 27 *symboles* $\Sigma = \{a, b, \dots, y, z, \$\}$ qui se représentent par le type `char`. L'ensemble des *mots* sur l'alphabet Σ se note Σ^* ; la longueur d'un mot $w \in \Sigma^*$ se note $|w|$. Dans toute cette sous-section, nous fixons deux mots : le mot $b = b_1 \dots b_m$, dit *brouillé*, de longueur m et le mot $c = c_1 \dots c_n$, dit *ciblé*, de longueur n .

Définition : Nous appelons *distance* entre un mot brouillé $b \in \Sigma^*$ et un mot ciblé $c \in \Sigma^*$, et nous notons $\text{dist}(b, c)$, le nombre minimum de symboles qu'il faut supprimer, insérer ou substituer à un autre symbole pour transformer le mot brouillé b en le mot ciblé c . On remarque que $\text{dist}(b, c) = \text{dist}(c, b)$.

Par exemple, la distance entre le mot brouillé $b = \text{corekte}$ et le mot ciblé $c = \text{correct}$ vaut 3 : en effet, on peut insérer un `r`, substituer un `c` au `k` et supprimer le `e` final pour passer du mot b au mot c ; par ailleurs, on peut vérifier que cette transformation est impossible en effectuant deux opérations ou moins.

Nous notons $\text{préf}_i(b)$ le *préfixe de longueur i* du mot b , c'est-à-dire le mot des i premiers symboles de b . Pour $i = 0$, il s'agit du mot vide ϵ . Pour tous les indices i compris entre 0 et m et pour j compris entre 0 et n , nous notons $d_{i,j} = \text{dist}(\text{préf}_i(b), \text{préf}_j(c))$ la distance entre les préfixes $\text{préf}_i(b)$ et $\text{préf}_j(c)$.

□ 5 – Pour tous les entiers i compris entre 0 et m et j compris entre 0 et n , déterminer les distances $d_{i,0}$ et $d_{0,j}$.

□ 6 – Pour tous les entiers i compris entre 0 et $m - 1$ et j compris entre 0 et $n - 1$, exprimer la distance $d_{i+1,j+1}$ en fonction des distances $(d_{i',j'})_{\substack{0 \leq i' \leq i \\ 0 \leq j' \leq j}}$.

Épreuve d'option informatique MP 2022

Indication Ocaml : Un élément de type `char` se déclare entre deux apostrophes : par exemple, on code `'a'` pour définir le symbole `a`. Les mots de Σ^* se représentent par le type `char list`. Nous déclarons

```
type mot = char list;;
```

Nous rappelons la syntaxe des fonctions suivantes :

- `List.length`, de type `'a list -> int` : la longueur d'une liste ℓ est `List.length l` ;
- `Array.make`, de type `int -> 'a -> 'a array` : un tableau de longueur n dont chaque case est initialisée avec la valeur v s'obtient par `Array.make n v`. Dans un tableau, les indices sont numérotés à partir de 0. On accède au coefficient en position i du tableau a par l'expression `a.(i)`, on le modifie par l'instruction `a.(i) <- v`.

- 7 – Écrire une fonction `array_of_mot (w:mot) : char array` dont la valeur de retour est un tableau formé des symboles du mot w écrits dans le même ordre.

Indication Ocaml : Nous rappelons la syntaxe de la fonction suivante :

- `Array.make_matrix`, de type `int -> int -> 'a -> 'a array array` : une matrice de taille $s \times t$ dont toutes les cases sont initialisées avec la valeur v s'obtient par `Array.make_matrix s t v`. Dans une matrice, les indices sont numérotés à partir de 0. On accède au coefficient en position (i, j) de la matrice a par l'expression `a.(i).(j)`, on le modifie par l'instruction `a.(i).(j) <- v`.

- 8 – Écrire une fonction `distance (b:mot) (c:mot) : int` qui calcule par mémoïsation la distance `dist(b, c)`.

Indication Ocaml : Nous rappelons la syntaxe de la fonction suivante :

- `List.filter`, de type `('a -> bool) -> 'a list -> 'a list` : la sous-liste des éléments x de la liste ℓ tels que le prédicat $p(x)$ vaut `true` s'obtient par `filter p l`.

- 9 – Soit $n_{\max}$ un entier. Exprimer la complexité en temps de la fonction `distance b c` en fonction des longueurs m et n des mots b et c . En déduire la complexité de l'instruction `List.filter (fun c -> (distance b c) <= k) lc` ; où ℓ_c est une liste de mots ciblés, chacun de longueur inférieure ou égale à $n_{\max}$, et où k est un entier naturel.

- 10 – Soit k la distance `dist(b, c)`. Montrer qu'il existe un entier r , des mots $b_0, b_1, \dots, b_r$ appartenant chacun à Σ^* , des mots $c_0, c_1, \dots, c_r$ appartenant chacun à Σ^* et des symboles $x_0, x_1, \dots, x_{r-1}$ appartenant chacun à Σ tels que

$$b = b_0 x_0 b_1 \dots x_{r-1} b_r \quad \text{et} \quad c = c_0 x_0 c_1 \dots x_{r-1} c_r$$

et

$$k = \sum_{i=0}^r \max(|b_i|, |c_i|).$$

2 Fouille dans un trie

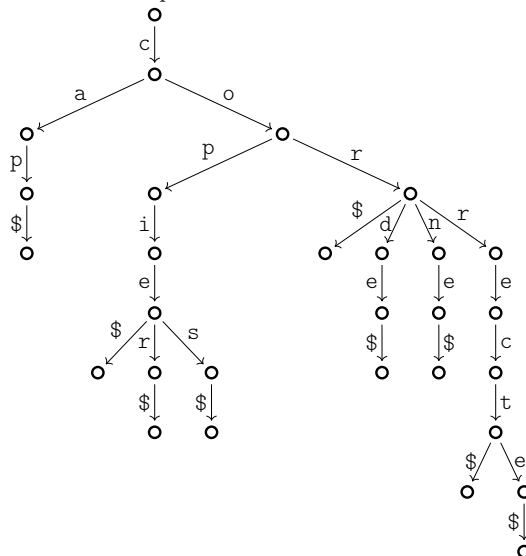
2.1 Représentation d'un corpus de mots par un trie

Définition : Un *trie* est un type particulier d'arbre dont chaque arête est orientée de la racine vers les feuilles et est étiquetée par un symbole de Σ . Le trie vide est formé d'un seul sommet et de zéro arête. Lorsqu'une arête est étiquetée par le symbole \$, son extrémité finale doit être le trie vide. La taille d'un trie t , notée $|t|$, est le nombre de ses arêtes.

On dit qu'un mot $c = c_1 \dots c_n \in (\Sigma \setminus \{\$\})^n$ *appartient* à un trie s'il existe un chemin, constitué de $n + 1$ arêtes, issu de la racine et dont les arêtes successives ont pour étiquettes respectives les symboles $c_1, c_2, \dots, c_n$ et \$. Le symbole \$ indique la présence d'un mot et joue le rôle de symbole terminateur.

Dans cette partie, les tries sont utilisés afin de représenter un corpus de mots ciblés.

FIGURE 1 – Exemple de trie de taille 28 contenant 9 mots.



□ 11 – Donner l'ensemble des mots du trie représenté à la figure 1.

Dans ce sujet, le terme *dictionnaire* désigne en général un type de données abstrait qui contient des associations entre une clé et une valeur.

□ 12 – Nommer deux structures de données concrètes, qui réalisent le type abstrait *dictionnaire*. Pour chacune d'entre elles, rappeler sans justifier la complexité en temps de l'opération *insertion*.

Épreuve d'option informatique MP 2022

Indication Ocaml : Nous définissons un type polymorphe `'a char_map` qui réalise une structure de données persistante de dictionnaire associant des clés de type `char` à des valeurs de type quelconque `'a`. Nous disposons de la constante et de la fonction suivante :

- `CharMap.empty`, de type `'a char_map`, qui désigne le dictionnaire vide.
- `CharMap.add`, de type `char -> 'a -> 'a char_map -> 'a char_map`, telle que la valeur de retour de `CharMap.add x t d` est un dictionnaire contenant les associations du dictionnaire d ainsi qu'une association supplémentaire entre la clé x et la valeur t . Si la clé x était déjà associée dans le dictionnaire d , l'ancienne association de x disparaît.

Pour représenter les tries, nous définissons le type

```
type trie = Node of trie char_map ;;
```

- ☐ 13 – Définir deux constantes `trie_vide` et `trie_motvide`, de type `trie`, qui réalisent respectivement le trie vide et le trie contenant le mot vide ϵ .
- ☐ 14 – Écrire une fonction `trie_singleton (x:char) : trie` qui construit un trie contenant le mot formé d'un seul symbole $x \in \Sigma \setminus \{\$ \}$.

Indication Ocaml : Nous utilisons un type polymorphe `'a option` défini par

```
type 'a option =
  | None
  | Some of 'a;;
```

qui permet de représenter des valeurs de type `'a` parfois non définies. Nous complétons le type `'a char_map` par la fonction suivante :

- `CharMap.find`, de type `char -> 'a char_map -> 'a option`, telle que l'appel de `CharMap.find x d` renvoie, en temps constant, `Some t` si la clé x est associée à la valeur t dans le dictionnaire d et renvoie `None` s'il n'existe pas d'association de clé x dans le dictionnaire d .

- ☐ 15 – Écrire une fonction `trie_mem (c:mot) (Node tcm:trie) : bool` qui teste si le mot $c \in (\Sigma \setminus \{\$ \})^*$ appartient au trie $t = \text{Node } tcm$.
- ☐ 16 – Écrire une fonction `trie_add (c:mot) (Node tcm:trie) : trie` dont la valeur de retour est un trie contenant les mêmes mots que le trie $t = \text{Node } tcm$ ainsi que le mot $c \in (\Sigma \setminus \{\$ \})^*$.
- ☐ 17 – Nous construisons le trie représenté à la figure 1 en déclarant d'abord la constante `trie_vide` (question 13), puis en appliquant neuf fois la fonction `trie_add` (question 16). Compte tenu du caractère persistant du type `'a char_map`, combien d'exemplaires de `trie_vide` coexiste-t-il une fois la construction terminée ? Expliquer.

Définition : Un trie est dit *élagué* si toute feuille est précédée d'une arête d'étiquette \$.

Indication Ocaml : Nous complétons le type 'a char_map par les fonctions suivantes :

- CharMap.is_empty, de type 'a char_map -> bool, qui teste si un dictionnaire est le dictionnaire vide.
- CharMap.filter_map, de type (char -> 'a -> 'a option) -> 'a char_map -> 'a char_map, telle que CharMap.filter_map f d renvoie le dictionnaire d' restreint aux associations entre la clé x et la valeur t' où, d'une part, il existe dans le dictionnaire d une association entre la clé x et la valeur t et, d'autre part, $f(t)$ vaut Some tprime. Si le dictionnaire d contient un couple clé et valeur (x, t) mais que $f(t)$ vaut None, alors le dictionnaire d' ne contient pas d'association de clé x . En voici une illustration :

d	x_1	$\mapsto$	t_1	d'	x_1	$\mapsto$	t'_1	$f(t_1) = \text{Some } t1\text{prime}$
	x_2	$\mapsto$	t_2					$f(t_2) = \text{None}$
	x_3	$\mapsto$	t_3		x_3	$\mapsto$	t'_3	$f(t_3) = \text{Some } t3\text{prime}$
	$\vdots$		$\vdots$		$\vdots$		$\vdots$	

□ 18 – Compléter le code suivant afin que la valeur de retour de `trie_trim t` soit un trie élagué contenant les mêmes mots que le trie $t = \text{Node } tcm$.

```
let rec trie_trim (Node tcm:trie) : trie =
  let filtre (x:char) (y:trie) : trie option =
    (* a completer *)
  in
  Node(CharMap.filter_map filtre tcm);;
```

2.2 Filtrage dans un trie

□ 19 – Soient $\mathbf{b} \in (\Sigma \setminus \{\$\})^*$ un mot brouillé, t un trie contenant un ensemble de mots ciblés et k un entier naturel. On suppose avoir engendré la liste $\ell_{\mathbf{b}}$ des mots de $(\Sigma \setminus \{\$\})^*$ à distance inférieure ou égale à k du mot $\mathbf{b}$. Montrer que la liste $\ell_{\mathbf{b}}$ compte $O(|\mathbf{b}|^k)$ éléments. En déduire la complexité en temps de l'instruction `List.filter (fun bb -> trie_mem bb t) lb;;`.

Définition : Nous appelons *système de transitions* sur l'alphabet Σ la donnée d'un triplet $(Q, \hat{q}, \Delta)$ où

- Q est un ensemble (fini ou non), dit *ensemble des états*,
- $\hat{q} \in Q$ est un état, dit *état initial*,
- $\Delta \subseteq Q \times \Sigma \times Q$ est une relation, dite *relation de transition*.

Un mot $\mathbf{c} = c_1 \dots c_n \in \Sigma^n$ est *accepté* par le système de transitions $(Q, \hat{q}, \Delta)$ s'il existe une suite d'états $(q_j)_{0 \leq j \leq n}$ telle que l'état q_0 égale l'état initial $\hat{q}$ et, pour tout j compris entre 0 et $n-1$, le triplet (q_j, c_{j+1}, q_{j+1}) appartient à la relation de transition Δ .

Épreuve d'option informatique MP 2022

On peut voir un système de transitions comme un automate éventuellement infini dont tous les états sont finals.

□ 20 – Dessiner, sans justifier, un système de transitions fini qui accepte les mots $w \in \Sigma^*$ n'ayant pas le mot `ccmp` comme facteur (c'est-à-dire que le mot w n'est pas accepté si et seulement s'il contient quatre symboles consécutifs valant `c`, `c`, `m` et `p`).

Nous disons qu'un système de transitions $(Q, \hat{q}, \Delta)$ est *déterministe* s'il existe une fonction partiellement définie $\delta : Q \times \Sigma \rightarrow Q$ telle que l'on ait

$$\Delta = \{(q, x, \delta(q, x)) ; (q, x) \in Q \times \Sigma \text{ et } \delta(q, x) \text{ est défini}\}.$$

Nous notons alors $(Q, \hat{q}, \delta)$ ce système.

Indication Ocaml : Afin de représenter des systèmes de transitions déterministes, nous convenons de nous appuyer sur un type `etat` pour représenter l'ensemble des états Q . Ce type sera explicité ultérieurement. Nous déclarons

```
type syst_trans = etat -> char -> etat option;;
```

pour représenter la fonction de transition δ . Pour tout couple $(q, x) \in Q \times \Sigma$, si `delta` est de type `syst_trans`, on accède à l'état image $q' = \delta(q, x)$ par l'expression `delta q x` qui vaut alors `Some qprime` si la transition est bien définie ou bien `None` si la transition n'est pas définie.

□ 21 – Écrire une fonction `trie_filter (qchapeau:etat) (delta:syst_trans) (Node tcm:trie) : trie` qui renvoie un trie contenant les mots du trie $t = \text{Node tcm}$ acceptés par le système de transitions déterministe $(Q, \hat{q}, \delta)$. Il n'est pas demandé de renvoyer un trie élagué.

□ 22 – Un système de transitions déterministe $(Q, \hat{q}, \delta)$ étant fixé, quelle est la complexité en temps du calcul `trie_filter qchapeau delta t` en fonction du trie t ? On suppose que l'exécution de la fonction de transition δ s'effectue en temps constant.

3 Système de transitions des voisins d'un mot brouillé

Dans toutes les questions restantes, les lettres b et c désignent systématiquement deux mots de longueur m et n de $(\Sigma \setminus \{\$\})^*$. La lettre k désigne un entier naturel. L'écriture $c\$$ désigne la concaténation du mot c et du symbole $\$$; nous parlons alors de *mot prolongé*.

L'objectif de cette section est de construire un système de transitions non déterministe qui, à partir d'un entier naturel k et d'un mot brouillé b , accepte n'importe quel préfixe du mot prolongé $c' = c\$$ où c est un mot de $(\Sigma \setminus \{\$\})^*$ tel que $\text{dist}(b, c) \leq k$ et aucun autre mot n'est accepté.

Définition : Nous appelons *transformations élémentaires* les $(k+3)$ appellations **suppr**, **subs** et **h -ins-puis-id**, avec $0 \leq h \leq k$; nous notons $\mathcal{T}$ leur ensemble.

Soient $\mathbf{c}' = c'_1 \dots c'_n \in \Sigma^*$ un mot de longueur n et $\mathbf{b}' \in \Sigma^*$ un mot de longueur m . Nous disons qu'une suite $\tau = (\tau_1, \tau_2, \dots, \tau_n)$ de n transformations élémentaires est un *script de transformation* du mot $\mathbf{c}'$ en le mot $\mathbf{b}'$ s'il existe une factorisation $\beta_1 \beta_2 \dots \beta_n$ du mot $\mathbf{b}'$ telle que pour tout entier j compris entre 1 et n ,

- (1) β_j est un mot de Σ^* ,
- (2) lorsque $\tau_j = \mathbf{suppr}$, alors le mot β_j est le mot vide ϵ ,
- (3) lorsque $\tau_j = \mathbf{subs}$, alors le mot β_j est de longueur 1 et, en l'identifiant à un symbole, il est distinct du symbole c'_j ,
- (4) lorsque $\tau_j = \mathbf{h-ins-puis-id}$, alors le mot β_j est de longueur $h+1$ et le dernier symbole de β_j vaut le symbole c'_j .

Nous observons que la factorisation du mot $\mathbf{b}'$ en $\beta_1 \beta_2 \dots \beta_n$ est unique et ne dépend que du script τ .

Voici un exemple de script de transformation entre le mot $\mathbf{c}' = \text{correct}\$$ et le mot $\mathbf{b}' = \text{incorekte}\$$.

$(c'_j)_{1 \leq j \leq 8}$	c	o	r	r	e	c	t	\$
$(\beta_j)_{1 \leq j \leq 8}$	inc	o	r	ϵ	e	k	t	e\$
$(\tau_j)_{1 \leq j \leq 8}$	2-ins-puis-id	0-ins-puis-id	0-ins-puis-id	suppr	0-ins-puis-id	subs	0-ins-puis-id	1-ins-puis-id

Définition : Le *coût* d'une transformation élémentaire est précisé par le tableau suivant :

Transformation élémentaire	suppr	subs	h -ins-puis-id
Coût	1	1	h

Le coût d'un script de transformation est la somme des coûts des transformations élémentaires constituant le script.

Dans l'exemple ci-dessus, le coût du script vaut 5 (ou encore $2 + 0 + 0 + 1 + 0 + 1 + 0 + 1$).

Définition : Nous utilisons le terme *k -script* pour raccourcir l'expression « script de transformation de coût inférieur ou égal à k ».

□ 23 – Montrer que la distance $\text{dist}(\mathbf{b}, \mathbf{c})$ est inférieure ou égale à k si et seulement s'il existe un k -script du mot prolongé $\mathbf{c}\$$ vers le mot prolongé $\mathbf{b}\$$.

Dans les questions 24 et 25, la lettre j désigne un entier avec $0 \leq j \leq n$ et $\tau = (\tau_1, \tau_2, \dots, \tau_j) \in \mathcal{T}^j$ un k -script depuis le préfixe $\text{préf}_j(\mathbf{c}\$)$ du mot prolongé $\mathbf{c}\$$ vers un certain préfixe $\mathbf{p}$ du mot prolongé $\mathbf{b}\$$. On appelle $\mathbf{s}$ le suffixe de $\mathbf{b}\$$ tel que $\mathbf{b}\$$ se factorise en $\mathbf{b}\$ = \mathbf{p}\mathbf{s}$.

□ 24 – Si l'on a $0 \leq j < n$, par quelles appellations $\tau_{j+1} \in \mathcal{T}$ est-il possible de compléter le k -script τ pour que $(\tau_1, \tau_2, \dots, \tau_{j+1})$ soit un k -script depuis le préfixe $\text{préf}_{j+1}(\mathbf{c}\$)$ vers un certain préfixe du mot prolongé $\mathbf{b}\$$? On exprimera sa réponse en fonction du $(j+1)^{\text{e}}$ symbole c_{j+1} de $\mathbf{c}\$$, du suffixe $\mathbf{s}$ et du coût κ du script τ .

□ 25 – Si l'on a $j = n$, par quelles appellations $\tau_{n+1} \in \mathcal{T}$ et sous quelles conditions est-il possible de compléter le k -script τ pour que $(\tau_1, \tau_2, \dots, \tau_{n+1})$ soit un k -script depuis le mot prolongé $\mathbf{c}\$$ vers le mot prolongé $\mathbf{b}\$$?

Indication Ocaml : Nous précisons le type `etat` en déclarant

```
type etat = mot * int;;
```

□ 26 – Le mot brouillé $\mathbf{b} \in (\Sigma \setminus \{\$, \})^*$ étant toujours fixé, décrire en OCaml un système de transitions $(Q, \hat{q}, \Delta)$ qui accepte tout préfixe du mot $\mathbf{c}\$,$ avec $\mathbf{c} \in (\Sigma \setminus \{\$, \})^*$, tel que $\text{dist}(\mathbf{b}, \mathbf{c}) \leq k$ et qui n'accepte aucun autre mot. On définira une fonction `etat_initial (b:mot) (k:int) : etat` qui construit l'état initial $\hat{q}$ et une fonction `delta (k:int) (q:etat) (x:char) : etat list` qui renvoie la liste des états q' tels que l'on ait $(q, x, q') \in \Delta$.

□ 27 – Comment adapter la fonction `trie_filter`, écrite à la question 21, pour qu'elle fonctionne avec le système de transitions de la question 26 ? On ne demande pas de code. Préciser la complexité en temps. Pourquoi peut-on souhaiter adapter la fonction `trie_filter` plutôt que de déterminer le système de transitions ?

FIN DE L'ÉPREUVE

SOLUTION

1. Elle est de type $int\ list \rightarrow int$ car l'entrée est une liste (filtrage par motif *match with* et le premier cas est la liste vide $[]$, la sortie est un entier car la liste vide renvoie max_int qui l'est, les éléments sont des nombres car ils sont comparés via $a \leq b$ et ils sont entiers car le cas $[a] \rightarrow a$ renvoie un élément de cet liste qui doit être de même nature que max_int).

Commentaires : *Question de cours, elle ne doit pas poser de problème particulier.*

2. On procède par récurrence sur la longueur de la liste z . Si elle est de longueur 0, elle est vide donc *mystere* z renvoie immédiatement max_int (premier cas de filtrage) donc la propriété est vraie au rang 0. Si elle est de longueur 1, elle est réduite à un élément $[a]$ donc *mystere* z renvoie immédiatement a (deuxième cas du filtrage) donc la propriété est vraie au rang 1. Considérons un entier $n \geq 2$ et supposons la propriété vérifiée pour l'entier $n - 1$. Si z est une liste de longueur n , elle possède donc au moins 2 éléments donc elle s'écrit $z = a : : b : : y$. Si $a \leq b$ (resp. $a > b$) alors *mystere* est exécutée sur la liste $a : : y$ (resp. $b : : y$) qui est de longueur $n - 1$ (troisième cas du filtrage) donc elle s'arrête par hypothèse de récurrence. Ainsi, la propriété est vérifiée au rang n ce qui prouve l'hérédité et donc la récurrence. La fonction *mystere* est appelée n fois où n est la longueur de la liste d'entrée z .

Commentaires : *Il est attendu une véritable preuve et non pas des déclarations d'intention ou des affirmations non fondées. Il faut procéder par récurrence et s'appuyer sur l'intégralité du code (factuellement). Ce type de question est toujours difficile pour les candidats. Entraînez vous sur cette question qui est relativement simple.*

3. La fonction *mystere* calcule le minimum de la liste z si elle est non vide et renvoie max_int sinon. Ceci est la propriété demandée par la question.
Si z est de longueur 0, l'exécution de *mystere* z renvoie max_int (premier cas du filtrage).
Si z est de longueur 1, elle est de la forme $[a]$, l'exécution de *mystere* z renvoie a (deuxième cas du filtrage) qui est bien la plus petite valeur de z .
Soit $n \geq 2$ et supposons la propriété vérifiée pour toute liste de longueur $n - 1$. Considérons une liste z de longueur $n \geq 1$ donc elle contient au moins deux éléments c'est-à-dire qu'elle est de la forme $a : : b : : z$ avec z une liste. Si on note c la plus petite valeur entre a et b alors *mystere* z appelle *mystere* sur la liste $c : : z'$ qui est de longueur $n - 1$. Par hypothèse de récurrence, cet appel sur cette liste renvoie la plus petite valeur de la liste $c : : z'$ qui aussi la plus petite valeur de $a : : b : : z'$ c'est-à-dire de z . La propriété est prouvée au rang n ce qui achève la récurrence.

Commentaires : *Ce que calcule mystere est de difficulté standard. Pour la preuve, même remarque que précédemment.*

4. Soit n la longueur de la liste z . L'exécution de *mystere* z amène n appels de *mystere* et le stockage de chaque appel nécessite une quantité fixe de mémoire $O(1)$ (l'appel de *mystere*, le stockage de $c = \min(a, b)$ et le pointage de c vers la liste y , qui est préalablement stocké via z , pour créer la liste d'entrée de *mystere*) donc un stockage mémoire en $nO(1) = O(n)$. Cette complexité est optimale car il est indispensable de stocker les n éléments de la liste z en entrée dont le coût est en $nO(1) = O(n)$ (chaque élément a un stockage en $O(1)$ et il fait faire pointer les éléments vers le reste de la liste via $: :$).

Commentaires : Question hors programme (depuis 2021). Les candidats lors du concours ont fréquemment confondu complexité en espace (c'est-à-dire en mémoire) et en temps.

5. $d_{i,0} = i$ (pour passer d'un mot à i lettres à un mot à 0 lettres, il suffit d'effacer une à une les lettres, on ne peut pas faire moins!).
 $d_{0,j} = j$ (pour passer d'un mot vide à un mot à j lettres, il suffit d'ajouter une à une les lettres, on ne peut pas faire moins!).

Commentaires : Question élémentaire.

6. Considérons deux mots b' et c' respectivement préfixe de longueur $i + 1$ de b et de longueur $j + 1$ de c donc ils s'écrivent

$$b' = b_0 \cdots b_i, c' = c_0 \cdots c_j$$

avec b_k et c_k les lettres d'indice k de b et c . Pour passer de b' à c' , nous allons distinguer différents cas.

Premier cas $b_i = c_i$ donc il faut juste passer de $b'' = b_0 \cdots b_{i-1}$ à $c'' = c_0 \cdots c_{j-1}$. Le nombre minimum d'opérations est alors $d_{i,j}$ donc $d_{i+1,j+1} = d_{i,j}$.

Deuxième cas $b_i \neq c_j$. Il faut alors faire au moins une opération élémentaire. Nous allons encore distinguer des cas.

Premier sous-cas. On remplace b_i par c_j (une opération) donc il faut maintenant passer du mot $b' = b_0 \cdots b_{i-1}c_j$ au mot $c = c_0 \cdots c_j$ ce qui revient à passer du mot $b'' = b_0 \cdots b_{i-1}$ (de longueur i) au mot $c' = c_0 \cdots c_{j-1}$ (de longueur j) Le coût minimal de ce passage est $d_{i,j}$ donc $d_{i+1,j+1} = 1 + d_{i,j}$.

Deuxième sous-cas. On efface b_i (une opération) donc il faut maintenant passer du mot $b' = b_0 \cdots b_{i-1}$ (de longueur i) au mot c' (de longueur $j + 1$). Le coût minimal de ce passage est $d_{i,j+1}$ donc $d_{i+1,j+1} = 1 + d_{i,j+1}$.

Troisième sous-cas. On effectue le passage du mot b' (de longueur $i + 1$) au mot $c'' = c_0 \cdots c_{j-1}$ (de longueur j), ce qui nécessite $d_{i+1,j}$ opérations puis on insère c_j au mot c'' (1 opération). Ainsi, on passe du mot b' au mot c' en $1 + d_{i+1,j}$ opération. Ainsi, $d_{i+1,j+1} = 1 + d_{i+1,j}$.

Au final, il faudra prendre la plus petite de toutes les valeurs précédentes c'est-à-dire

$$d_{i+1,j+1} = \begin{cases} \min(d_{i,j}, 1 + d_{i+1,j}, 1 + d_{i,j+1}) & \text{si } c_i = c_j \\ \min(1 + d_{i,j}, 1 + d_{i+1,j}, 1 + d_{i,j+1}) & \text{si } c_i \neq c_j \end{cases}$$

Commentaires : Question très difficile, surtout à l'époque où la programmation dynamique n'était pas au programme. Même actuellement, elle reste une question difficile. Au final, peu de candidats ont sut trouvé la réponse. Surtout que l'énoncé rend la question encore plus difficile (exprimer $d_{i+1,j+1}$ en fonction des $(d_{\alpha,\beta})_{0 \leq \alpha \leq i, 0 \leq \beta \leq j}$ et des $(d_{\alpha,j+1})_{0 \leq \alpha \leq i}$ et $(d_{i+1,\beta})_{0 \leq \beta \leq j}$).

7.

```
let array_of_mot w =
  let n = List.length w in
  let t = Array.make n (List.hd w) in
  let rec aux i w = match w with
    | [] -> t
```

```
|c::q -> t.(i)<-c ; aux (i+1) q
in aux 0 w;;
```

Commentaires : Question élémentaire.

8.

```
let distance b c =
  let tb = array_of_mot b and tc = array_of_mot c in
  let nb = Array.length tb and nc = Array.length tc in
  let d = Array.make_matrix (nb+1) (nc+1) 0 in
  for i = 0 to nb do
    d.(i).(0) <- i done;
  for j = 0 to nc do
    d.(0).(j) <- j done;
  for i=0 to nb-1 do
    for j=0 to nc-1 do
      if tb.(i+1)==tc.(j+1) then d.(i+1).(j+1) <- min (min d.(i).(j) (1+d.(i+1).(j))) (1+d.(i).(j+1));
      else d.(i+1).(j+1) <- 1 + min (min d.(i).(j) d.(i+1).(j)) d.(i).(j+1);
    done;done;
  d.(nb).(nc);;
```

Commentaires : Question à l'origine inaccessible si la réponse à la question 6 n'est pas produite ! Pour ceux qui l'ont obtenu, la rédaction fut par récursion pure ce qui amène une complexité exponentielle. L'utilisation de la mémorisation (mémoriser les calculs $d_{\alpha,\beta}$ pour rendre la complexité polynomiale) est peu exploitée. Travailler ce type de question qui va probablement devenir un standard vu qu'il est désormais au programme de l'ITC et de l'option informatique.

9. La complexité de *distance b c* est en $O(mn)$ (les deux boucles imbriquées, les opérations qu'elles contiennent sont en nombre bornés).

La commande *List.filter (fun c -> (distance b c) <= k) lc* exécute la fonction *distance b c* au plus $|lc|$ fois. Comme *c* a une longueur au plus $n_{\max}$ durant les appels (ce sont les éléments de la liste *lc*) et *b* étant de longueur *n*, chaque appel de *distance* a un coût en $O(n_{\max} m)$ et il est répété $|lc|$ fois d'où une complexité totale en $|lc| O(n_{\max} m) = O(|lc| n_{\max} m)$ (le test booléen $\leq k$ ayant un coût borné).

Commentaires : Question inaccessible vu que la plupart des candidats n'ont pas su traiter les questions 6 et 8. Si ces questions sont traitées, cette question est de difficulté standard.

10. Si *b* et *c* n'ont aucune lettre commune alors on choisit $r = 1$, $b_0 = b$ et $c_0 = c$, les éléments $(x_i)_{1 \leq i \leq r-1}$ n'existent pas (car $1 \leq r-1 = 0$ est impossible). Si $b = b_0 \cdots b_{n-1}$ est plus court que $c = c_0 \cdots c_{m-1}$ (i.e. $n \leq m$), il faut au moins m opérations pour passer de *b* à *c* (ne serait que pour faire apparaître toutes les lettres de *c* dans *b*). Or, si on change b_0 par c_0 , b_1 par c_1 , ..., b_n par c_n (donc *b* est devenu $c_1 \cdots c_n$) puis on insère à ce mot successivement les

lettres $b_{n+1}, \dots, b_{m-1}$, on vient de transformer le mot b en le mot c en m opérations. Ainsi, $d(b, c) = m = \max(|b|, |c|)$ dans ce cas. Si c est plus court que b , comme

$$d(b, c) = d(c, b) = \max(|c|, |b|) = \max(|b|, |c|),$$

Si b et c ont au moins une lettre commune, notons x_0 la première lettre que l'on lit dans b et c (dans la lecture de gauche à droite). Alors b et c s'écrivent $b = b_0 x_0 b'$ et $c = c_0 x_0 c'$ avec b_0 et c_0 sont deux préfixes de b (éventuellement vide) sans aucune lettre commune et b', c' deux suffixes de b et c (éventuellement vide). Pour passer de b à c , il est indispensable de transformer b_0 en c_0 donc il faut au moins $\max(|b_0|, |c_0|)$ opérations et il faut passer de b' à c' donc il faut au moins $d(b', c')$ opérations. Remarquons qu'il existe $\max(|b_0|, |c_0|)$ opérations pour passer de b_0 à c_0 (d'après le début de notre réponse) et $d(b', c')$ opérations pour passer de b' à c' (par définition de la distance de Levenshtein) d'où

$$(*) : d(b, c) = \max(|b_0|, |c_0|) + d(b', c').$$

En itérant ce processus à b' et c' (ou bien en fait une récurrence forte), on obtient l'existence d'un nombre fini $r \geq 1$ de lettres $x_0, \dots, x_{r-1}$ et de mots $b_0, \dots, b_r, c_0, \dots, c_r$ tels que

$$b = b_0 x_0 b_1 x_1 \cdots x_{r-1} b_r \text{ et } c = c_0 x_0 c_1 x_1 \cdots x_{r-1} c_r$$

et en itérant la formule $(*)$, on obtient

$$d(b, c) = \max(|b_0|, |c_0|) + \max(|b_1|, |c_1|) + \cdots + \max(|b_r|, |c_r|).$$

Commentaires : Question difficile car elle demande de la part du candidat la compréhension de ce qui est demandé. Lors de cette épreuve, peu de candidats y parvinrent.

11. La lecture depuis la racine jusqu'aux feuilles (où se trouvent \$) nous donne : cap, copie, copier, copies, cor, corde, corne, correct, correcte.

Commentaires : Question élémentaire bien traitée en général.

12. On peut utiliser une liste formée de couples $(cle, valeur)$. L'insertion a une complexité en $O(n)$ où n est le nombre d'éléments du dictionnaire (la liste la modélisant). On peut également utiliser un arbre binaire de recherche dont les noeuds sont de la forme $(cle, valeur)$, la comparaison se faisant sur les clés. L'insertion a une complexité en $O(h)$ où h est la hauteur de l'arbre de recherche le modélisant. Comme h varie entre $O(\log_2(n))$ (pour les arbres équilibrés par exemple) et $O(n)$ (pour un arbre de recherche quelconque), l'insertion a une complexité variant en $O(\log_2(n))$ (efficace) ou $O(n)$ (comme les listes).

Commentaires : Question qui fut difficile au moment de l'épreuve et peu de candidats y parvinrent. La structure de dictionnaire était au programme (mais vu succinctement). Comme elle est étudiée désormais au programme de l'ITC et que la notion s'est affermie au programme de l'option informatique, il est souhaitable de travailler cette notion et ses réalisations concrètes au programme (celle de la réponse ci-dessous).

- 13.

```
let trie_vide = Node empty;;

let trie_vide = Node (add '$' trie_vide empty) ;;
```

Commentaires : Question très difficile car cela nécessite de la part du candidat de bien comprendre l'implémentation des tries. Comme cette dernière est non conventionnelle et abstraite, un nombre faible de candidats parvint à écrire une réponse correcte. C'est dommage que le sujet n'est pas donné un exemple non trivial et pas trop difficile de trie afin d'éclairer les candidats car le codage informatique n'est guère difficile.

14.

```
let trie_singleton x =
  Node (add x trie_vide empty);;
```

Commentaires : Même remarque que précédemment.

15.

```
let rec trie_mem c t = match c,t with
| [] , Node d -> CharMap.find '$' d != None
| x::q , Node d -> match CharMap.find x d with
| None -> false
| Some t' -> trie_mem q t';;
```

Commentaires : Même remarque que précédemment.

16.

```
let nouveau t x d = match CharMap.find x d with
| None -> trie_vide
| Some t -> t;;

let rec trie_add c t = match c,t with
| [],Node d -> let d1 = add '$' trie_vide d in Node d1
| x::q , Node d -> let t' = nouveau t x d in
  let t1 = trie_add q t' in let d1 = add x t1 d in Node d1
;;
```

Commentaires : Même remarque que précédemment.

17. La question affirme déclarer d'abord la constante *trie_vide* (let *trie_vide* = *Node empty*;;) donc, par caractère immuable (persistant) de cette constante, tous les appels de *add* ne modifie jamais *trie_vide* donc il n'existe qu'un seul exemplaire de *trie_vide*.

Commentaires : Même remarque que précédemment.

18.

```
let rec trie_trim (Node tcm:trie) : trie =
  let filtre x y =
    if x == '$' then Some trie_vide else
      choix (trie_trim y)
  in Node(filter_map filtre tcm);;
```

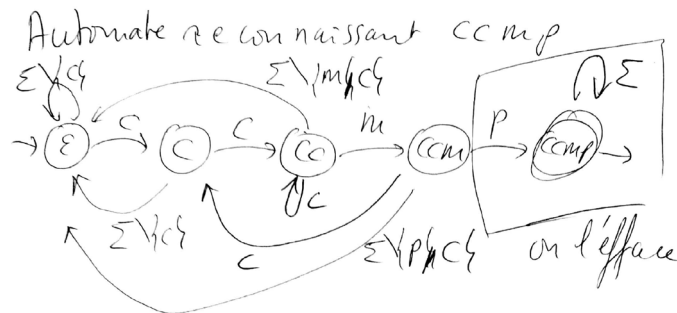
Commentaires : *Même remarque que précédemment.*

19. La distance entre b et c étant au plus de k , cela signifie qu'il existe au plus k opérations faisant passer du mot b au mot c . Une opération pouvant se positionner n'importe où dans b , il y a $|b|$ choix possibles pour la positionner (première lettre, deuxième, etc dernière). Chaque opération a un nombre fini de possibilités (insérer l'une des lettres de l'alphabet $\Sigma \setminus \{\$ \}$ qui est fixé donc $O(1)$ choix, idem pour la suppression ou l'insertion) donc $O(1)$ choix pour une position donnée. Ainsi, chaque opération a $|b| O(1) = O(|b|)$ choix et comme il y en a au plus k , cela fait au plus $(O(|b|))^k = O(|b|^k)$ choix de mots c . Par conséquent, la liste l_b est de longueur en $O(|b|^k)$.

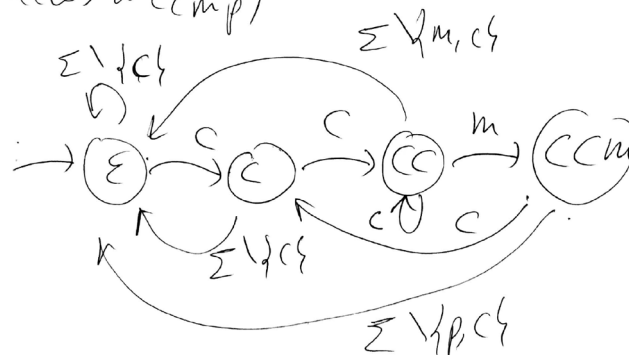
Comme la fonction `CharMap.find` a une complexité constante donc $O(1)$ (d'après l'énoncé), la fonction `fun bb-> trie_mem bb t` a une complexité en $O(|bb|)$ (elle appelle au plus $|bb|$ fois `CharMap.find`, les autres opérations ayant un coût borné et en nombre borné). Comme tous les éléments de la liste lb ont une longueur au plus en $|b| + k$ (ils s'obtiennent de b avec au plus k opérations donc dans le pire des cas k insertions), cette longueur est en $O(|b|)$, l'exécution de `fun bb-> trie_mem bb t` sur un élément bb de lb a une complexité en $O(O(b)) = O(|b|)$. Comme elle est répétée au plus $|lb| = O(|b|^k)$ fois dans l'instruction `List.filter (fun bb-> trie_mem bb t) lb`, la complexité totale est en $O(|b|) O(|b|^k) = O(|b|^{k+1})$.

Commentaires : *Le problème principal de cette question est un problème de combinatoire donc il s'avère toujours difficile pour les candidats. Seules les meilleures copies ont donné une réponse convenable.*

20.



système de transition reconnaissant
tous les mots sans le facteur $ccmp$
(car on efface cette possibilité, unique
accès à $ccmp$)



Commentaires : Si la réponse via un automate est classique, l'énoncé ne donne pas un automate. Au final, peu de candidats ont produit une réponse correcte.

21.

```
let rec trie_filter (qchapeau:etat) (delta:syst_trans) (Node
  tcm:trie) =
  let filtre2 x t =
    if x == '$' then Some trie_vide else
    match delta qchapeau x with
    None -> None
```

```
|Some q' -> Some (trie_filter q' delta t)
in Node(filter_map filtre2 tcm);;
```

Commentaires : *Même remarque qu'à la question 13.*

22. La fonction *delta* a une complexité constante d'après le sujet, donc en $O(1)$. La fonction *filtre2* x t s'exécute au plus le nombre de noeuds du trie t donc au plus $|t|$ fois. A chacun de ces appels, les opérations effectuées ont un coût borné (tests booléens, utilisation de *delta*, filtrage par motif, etc) donc la complexité totale est en $O(|t|)$.

Commentaires : *Question élémentaire mais ce ne fut pas le cas à cause de la remarque à la question 13. Comme quoi, un sujet intéressant peut facilement devenir inaccessible.*

23. Supposons qu'il existe un k -script $(\tau_1, \dots, \tau_n)$, $b' = \beta_1 \cdots \beta_n$ du mot prolongé $c\$$ au mot prolongé $b\$$. Dans ce cas, nous exécutons dans cet ordre les opérations $\tau_1, \dots, \tau_n$ sur les mots $\beta_1, \dots, \beta_n$ c'est-à-dire :

- si $\tau_j = \text{suppr}$, on supprime le mot β_j (de longueur 1 i.e. un caractère, le sujet manque de clarté sur ce sujet, dommage car cela joue un rôle clé dans la question!) que l'on supprime de c' donc une opération (suppression);
- si $\tau_j = \text{subs}$, on substitue à c'_j le caractère β_j donc une opération (substitution);
- si $\tau_j = \text{h-ins-puis-id}$, on insère avant le caractère c'_j les h caractères formant le mot β_j (par exemple, si $\beta_j = uvw \cdots$, on insère $uvw \cdots$ avant c'_j) donc cela fait h opérations (insertion).

A l'issue de ces transformations successives, on a transformé le mot c' en le mot b' en au plus $|u|$ final, on parvient à passer du mot c' au mot b' en un nombre d'opérations égale au coût du script donc au plus k .

Réciproquement, supposons que la distance de b à c donc il existe au plus k opérations consécutives qui font passer du mot c' au mot b' . Chaque caractère c_j de c' subit au plus 1 opération (s'il en subit plus d'une, il suffit d'en retirer une pour passer de c' à b' avec moins de $k - 1$ opérations!) donc s'il subit une opération, elle est unique.

Pour chaque entier j , si c_j est supprimé lors d'une de ces opérations, on pose $\beta_j = \varepsilon$ et $\tau_j = \text{suppr}$. Si c_j est substitué au moins une fois par un caractère b_k de b , on pose $\beta_j = b_k$ et $\tau_j = \text{subs}$. Si c_j n'est jamais supprimé ou substitué, on note h le nombre de lettres qui auront été insérées entre les caractères c_{j-1} et c_j , on note d_j le mot formé par ces h lettres. On pose alors $\beta_j = d_j$ et $\tau_j = \text{h-ins-puis-id}$. Il est manifeste que le coût de ce script est au plus k .

Commentaires : *Question difficile qui fut peu réussie. Probablement parce que les candidats étaient fatigués par le reste du sujet et la tentative de compréhension de la structure de trie. C'est dommage.*

24. On procède par disjonction de cas.

- Supposons que c_{j+1} est une lettre de s . Si on désigne par h le nombre de lettres $m_1, \dots, m_h$ précédents c_{j+1} (vu pour la première fois) dans s alors, en insérant ces h lettres avant c_{j+1} , on passe du préfixe de c de longueur $j + 1$ au préfixe $pm_1 \cdots m_h$ de $b\$$. On pose alors $\tau_{j+1} = \text{h-ins-puis-id}$ si $\kappa + h \leq k$. En effet, le coût de cette transformation étant h , le coût du script $(\tau_1, \dots, \tau_{j+1})$ vaut $\kappa + h$ et pour le retenir, ce coût ne peut dépasser k .

- Supposons que c_{j+1} n'est pas une lettre de s .
 - Si s est vide, on pose $\tau_{j+1} = \text{suppr}$ comme cela on passe du préfixe de c de longueur $j+1$ au préfixe p de $b\$$. Le coût du script $(\tau_1, \dots, \tau_{j+1})$ est alors κ donc inférieur ou égal à k .
 - Si s est non vide, il possède une première lettre m . On pose $\tau_{j+1} = \text{subs}$ (dont le coût est 1) si $\kappa + 1 \leq k$ comme cela on passe du préfixe de c de longueur $j+1$ au préfixe pm de $b\$$. Dans ce cas, le coût du script $(\tau_1, \dots, \tau_{j+1})$ est $\kappa + 1$ (d'où l'hypothèse $\kappa + 1 \leq k$) de coût inférieur à k .

Commentaires : Question très difficile. Seules quelques excellentes copies peuvent donner une réponse correcte.

- 25.** Raisonnement semblable à la question précédente nous donne : Si s est vide, on pose $\tau_{j+1} = \text{suppr}$ si $\kappa + 1 \leq k$. Si s est non vide de longueur h avec $\kappa + h \leq k$, on pose $\tau_{j+1} = \text{sub}$.

Commentaires : Même remarque que précédemment.

- 26 et 27.** Ces questions sont beaucoup trop difficile à ce stade du sujet et pour une durée de 3 heures. Aucun candidat n'a fournit une brève de réponse correcte à ces questions. Les corriger n'a aucun sens donc je m'en abstiens.

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES 1

Durée : 4 heures

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de deux exercices et d'un problème.

EXERCICE I

Dans cet exercice d'informatique commune, on se propose d'écrire des algorithmes dans le but de faire du calcul matriciel et plus particulièrement afin d'utiliser les matrices d'adjacence d'un graphe. Les algorithmes demandés doivent être écrits en langage **Python**. On sera très attentif à la rédaction et notamment à l'indentation du code. L'usage de toute librairie est **interdit**.

Notation

Les matrices sont carrées et représentées par des listes dont les éléments correspondent aux lignes de la matrice. Par exemple, la matrice $A = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ est représentée par la liste $[[1,2],[3,4]]$.

Dans la suite, pour définir la matrice d'adjacence $A \in \mathcal{M}_n(\mathbb{R})$ d'un graphe ayant n sommets, on numérote ses sommets de 0 à $n-1$.

- Q1.** Écrire une fonction *produit*(A,B) prenant en arguments deux matrices carrées A et B de mêmes dimensions et qui renvoie AB le produit de la matrice A par la matrice B .
- Q2.** Écrire une fonction *orienté*(A) prenant en argument la matrice d'adjacence A d'un graphe et qui retourne *True* si le graphe est orienté et *False* sinon.
- Q3.** On admet que le nombre de chemins de longueur p reliant i et j dans un graphe de matrice d'adjacence A est égal au coefficient d'indice (i,j) de la matrice A^p .
Écrire une fonction *distance*(A,i,j) où A est la matrice d'adjacence d'un graphe et qui renvoie le nombre minimal d'arêtes que l'on doit parcourir pour atteindre le sommet j depuis le sommet i (on suppose qu'un tel chemin existe).

On considère deux tables : CLIENTS et PARTENAIRES. La première contient des informations sur les clients et la deuxième permet d'identifier qui sont les partenaires des clients.

La table CLIENTS contient les attributs suivants :

- id : identifiant d'un individu (entier), clé primaire ;
- nom (chaîne de caractères) ;
- prenom (chaîne de caractères) ;
- ville (chaîne de caractères) ;
- email (chaîne de caractères).

La table PARTENAIRES contient les attributs suivants :

- id : identifiant de suivi (entier), clé primaire ;
- id_client : identifiant du client représenté par l'attribut **id** dans la table CLIENTS (entier) ;
- partenaire : nom du partenaire (chaîne de caractères).

- Q4.** Écrire une requête SQL permettant d'extraire les identifiants de tous les clients provenant de la ville de « Toulouse ».
- Q5.** Écrire une requête SQL permettant d'extraire les emails de tous les clients ayant « SCEI » comme partenaire.

EXERCICE II

On définit la fonction $f : (x, y) \mapsto x^2 - 2xy + 2y^2 + e^{-x}$ sur $\mathbb{R}^2$.

Q6. Établir que l'équation $e^{-x} = x$ admet une unique solution sur $\mathbb{R}$.

Q7. Démontrer que f possède un unique point critique $(x_0, y_0) \in \mathbb{R}^2$.

Q8. À l'aide de la matrice hessienne, démontrer que f admet un extremum local en (x_0, y_0) .
Est-ce un minimum ou un maximum ?

PROBLÈME

Dans tout le problème, α est un réel appartenant à l'intervalle $]0, 1[$. On pose :

$$I(\alpha) = \int_0^1 \frac{x^{\alpha-1}}{1+x} dx \quad \text{et} \quad J(\alpha) = \int_1^{+\infty} \frac{x^{\alpha-1}}{1+x} dx.$$

Partie I - Calcul d'une intégrale à l'aide d'une série

Q9. Démontrer que $x \mapsto \frac{x^{\alpha-1}}{1+x}$ est intégrable sur $]0, 1[$ et sur $[1, +\infty[$.

Q10. Démontrer que $J(\alpha) = I(1-\alpha)$.

On se propose maintenant d'écrire $I(\alpha)$ sous forme d'une somme de série.

Q11. 1^{re} tentative

Pour tout $x \in]0, 1[$, on pose $f_n(x) = (-1)^n x^{n+\alpha-1}$. Montrer que :

$$\forall x \in]0, 1[, \quad \frac{x^{\alpha-1}}{1+x} = \sum_{n=0}^{+\infty} f_n(x).$$

La série de fonctions $\sum f_n$ converge-t-elle uniformément sur $]0, 1[$?

Q12. 2^e tentative

Pour tout $x \in]0, 1[$, on pose :

$$S_n(x) = \sum_{k=0}^n (-1)^k x^{k+\alpha-1}.$$

À l'aide du théorème de convergence dominée, montrer que :

$$I(\alpha) = \lim_{n \rightarrow +\infty} \int_0^1 S_n(x) dx.$$

En déduire une expression de $I(\alpha)$ sous forme d'une somme de série.

Q13. En déduire que :

$$I(\alpha) + J(\alpha) = \int_0^{+\infty} \frac{x^{\alpha-1}}{1+x} dx = \frac{1}{\alpha} + 2\alpha \sum_{n=1}^{+\infty} \frac{(-1)^n}{\alpha^2 - n^2}.$$

On admet la formule suivante :

$$\forall x \in \mathbb{R}, \quad \cos(\alpha x) = \frac{\sin(\pi\alpha)}{\pi} \left(\frac{1}{\alpha} + \sum_{n=1}^{+\infty} (-1)^n \frac{2\alpha \cos(nx)}{\alpha^2 - n^2} \right).$$

Q14. Démontrer que :

$$\int_0^{+\infty} \frac{x^{\alpha-1}}{1+x} dx = \frac{\pi}{\sin(\alpha\pi)}.$$

Partie II - Lien avec la fonction Gamma

Dans toute la suite, on pose :

$$\forall x \in]0, +\infty[, \quad \Gamma(x) = \int_0^{+\infty} t^{x-1} e^{-t} dt,$$

et

$$\forall x \in [0, +\infty[, \quad f_\alpha(x) = \int_0^{+\infty} \frac{t^{\alpha-1}}{t+1} e^{-xt} dt.$$

Q15. Démontrer que Γ est bien définie sur $]0, +\infty[$.**Q16. Démontrer que f_α est bien définie et continue sur $[0, +\infty[$.**

Q17. Démontrer que f_α est de classe C^1 sur $]0, +\infty[$ et calculer sa dérivée.

Q18. Déterminer $\lim_{x \rightarrow +\infty} f_\alpha(x)$.

Q19. Démontrer que $t \mapsto \frac{e^{-t}}{t^\alpha}$ est intégrable sur $]0, +\infty[$. En déduire :

$$\lim_{x \rightarrow +\infty} \int_x^{+\infty} \frac{e^{-t}}{t^\alpha} dt.$$

Partie III - Vers la formule des compléments

Q20. Pour tout $x \in]0, +\infty[$, démontrer que :

$$f_\alpha(x) - f'_\alpha(x) = \frac{\Gamma(\alpha)}{x^\alpha}.$$

Q21. Pour tout $x \in]0, +\infty[$, on pose :

$$g_\alpha(x) = \Gamma(\alpha) e^x \int_x^{+\infty} \frac{e^{-t}}{t^\alpha} dt.$$

Vérifier que g_α est une solution particulière de l'équation différentielle $y - y' = \frac{\Gamma(\alpha)}{x^\alpha}$.

En déduire que $\forall x \in]0, +\infty[, f_\alpha(x) = g_\alpha(x)$.

Q22. En déduire que :

$$\int_0^{+\infty} \frac{t^{\alpha-1}}{t+1} dt = \Gamma(\alpha) \int_0^{+\infty} \frac{e^{-t}}{t^\alpha} dt.$$

Q23. Démontrer l'identité suivante (formule des compléments) :

$$\Gamma(\alpha)\Gamma(1-\alpha) = \frac{\pi}{\sin(\alpha\pi)}.$$

Q24. En déduire la valeur de l'intégrale de Gauss :

$$\int_0^{+\infty} e^{-t^2} dt.$$

FIN

5/5

SOLUTION

Q1. Deux programmes possibles (en créant directement la liste de listes ayant la bonne taille ou en la créant au fil de l'eau via *append*).

```
def produit(A,B):
    n=len(A)
    C=[[0 for j in range(n)] for i in range(n)]
    for i in range(n):
        for j in range(n):
            S=0
            for k in range(n):
                S = S + A[i][k]*B[k][j]
            C[i][j]=S
    return C

def produit(A,B):
    n=len(A)
    C=[]
    for i in range(n):
        temp=[]
        for j in range(n):
            temp.append(0)
        C.append(temp)
    for i in range(n):
        for j in range(n):
            S=0
            for k in range(n):
                S = S + A[i][k]*B[k][j]
            C[i][j]=S
    return C
```

Commentaires : Question de niveau standard ... si le candidat se rappelle de la formule du produit matriciel ! Au final, la question s'avère très discriminante.

Q2.

```
def oriente(A):
    n=len(A)
    for i in range(n):
        for j in range(i+1):
            if A[i][j] != A[j][i]:
                return True
    return False
```

Commentaires : Question de programmation élémentaire. Néanmoins, le sujet ne rappelant pas la définition d'un graphe orienté via sa matrice d'adjacence (non orienté : $\forall (i,j)$, si l'arête (i,j) est présente, l'arête (j,i) aussi c'est-à-dire $A[i][j] = A[j][i]$) et comme il s'agit d'une épreuve de mathématiques et comme il faut faire une négation, cette question s'avère sélective !

Q3.

```
def distance(A,i,j):
    n=len(A)
    temp = A
    for p in range(1,n+1):
        if temp[i][j] > 0 :
            return p
        temp = produit(temp,A)
    return (-1)
```

Commentaires : Question de niveau standard. Les erreurs à ne pas commettre : utiliser l'opération d'exponentiation $A ** k$ (car elle n'agit que sur des nombres, pas des matrices), éviter d'écrire $B \leftarrow A$ puis itérer l'opération $B \leftarrow B * B$ qui calcule A^{2^p} !

Q4.

```
SELECT id FROM clients WHERE ville = "Toulouse"
```

Commentaires : Question élémentaire de SQL. On peut s'interroger sur la pertinence de placer une question de SQL dans une épreuve de mathématiques. Est-ce que les candidats ont révisé l'ITC pour une épreuve de maths dont le cahier des charges était d'avoir des questions de Python contextualisés aux mathématiques ! On est loin. Je crains que cette question soit au final très discriminante et n'a aucun sens dans le cadre de cette épreuve.

Q5.

```
SELECT mail FROM clients JOIN partenaires ON clients.id=
    partenaires.id_client WHERE partenaires.partenaire = "scai"
```

Commentaires : Même remarque que précédemment.

Q6. La fonction $\varphi(x) = e^{-x} - x$ est continue et strictement sur $\mathbb{R}$ (comme somme de deux telles fonctions e^{-x} et $-x$) donc elle réalise une bijection de $] -\infty, +\infty[$ sur

$$\varphi(]-\infty, +\infty[) = \left] \lim_{+\infty} \varphi, \lim_{-\infty} \varphi \right[=]-\infty, +\infty[.$$

Comme $0 \in \varphi(]-\infty, +\infty[) =]-\infty, +\infty[$, l'équation

$$\varphi(x) = 0 \Leftrightarrow e^{-x} = x$$

admet une et une seule solution sur $\mathbb{R}$.

Commentaires : Question de difficulté standard. Il est attendu du candidat une argumentation rigoureuse (puisque la question est très classique et de niveau rudimentaire) : justifier correctement la bijectivité via le théorème de bijection, préciser l'intervalle image (qui n'a aucun rapport avec le fait que la fonction soit à valeurs dans $\mathbb{R}$! donc la détermination correcte des limites aux bornes $\pm\infty$).

Q7. La fonction f est de classe C^∞ sur $\mathbb{R}^2$ comme somme de telles fonctions ($x^2 - 2xy + 2y^2$ est une fonction polynômiale donc C^∞ et e^{-x} est C^∞). Un point $A = (x_0, y_0)$ est un point critique de f si et seulement si

$$\begin{aligned} \begin{cases} \frac{\partial f}{\partial x}(x_0, y_0) = 0 \\ \frac{\partial f}{\partial y}(x_0, y_0) = 0 \end{cases} &\Leftrightarrow \begin{cases} 2x_0 - 2y_0 - e^{-x_0} = 0 \\ -2x_0 + 4y_0 = 0 \end{cases} \\ &\Leftrightarrow \begin{cases} L_2 : 2y_0 = x_0 \\ 2x_0 - x_0 - e^{-x_0} = 0 \end{cases} \Leftrightarrow \begin{cases} e^{-x_0} = x_0 \\ y_0 = \frac{x_0}{2} \end{cases} \end{aligned}$$

Si α est l'unique solution de l'équation $e^{-x} = x$ (cf. q6) alors $(\alpha, \frac{\alpha}{2})$ est l'unique point critique de f .

Commentaires : Question de niveau standard.

Q8. Commençons par calculer les dérivées secondes de f puis évaluons les en l'unique point critique (α, α) de f (cf. question 7)

$$\begin{aligned} \frac{\partial^2 f}{\partial x^2}(x, y) &= \frac{\partial}{\partial x}(2x - 2y - e^{-x}) = 2 + e^{-x} \\ \frac{\partial^2 f}{\partial x \partial y}(x, y) &= \frac{\partial}{\partial x}(-2x + 4y) = -2 \\ \frac{\partial^2 f}{\partial y^2}(x, y) &= \frac{\partial}{\partial y}(-2x + 4y) = 4 \end{aligned}$$

La matrice hessienne de f en (α, α) vaut alors

$$H_f(\alpha, \alpha) = \begin{pmatrix} \frac{\partial^2 f}{\partial x^2}(\alpha, \alpha) & \frac{\partial^2 f}{\partial x \partial y}(\alpha, \alpha) \\ \frac{\partial^2 f}{\partial x \partial y}(\alpha, \alpha) & \frac{\partial^2 f}{\partial y^2}(\alpha, \alpha) \end{pmatrix} = \begin{pmatrix} 2 + e^{-\alpha} & -2 \\ -2 & 4 \end{pmatrix}.$$

Son déterminant vaut

$$(2 + e^{-\alpha})4 - (-2) \times (-2) = 4(1 + e^{-\alpha}) > 0$$

Par conséquent, (α, α) est un extrémum local de f ($H_f(\alpha, \alpha)$ est une matrice inversible et les deux valeurs propres de $H_f(\alpha, \alpha)$ sont de même signe car leur produit est le déterminant de $H_f(\alpha, \alpha)$ qui est positif). La trace de $H_f(\alpha, \alpha)$ étant $6 + e^{-\alpha}$ qui est positif, ses deux valeurs propres sont positives donc (α, α) est un minimum local de f .

Commentaires : Question qui demande au candidat de connaître la hessienne d'une fonction et de la condition suffisante pour savoir s'il s'agit d'un extrémum local ou pas. Pour cette raison, la question est très discriminante (malgré sa faible technicité).

Q9. La fonction $f : x \mapsto \frac{x^{\alpha-1}}{1+x}$ est continue sur $]0, 1]$ et elle est équivalente en 0 à $x^{\alpha-1} = \frac{1}{x^{1-\alpha}}$ qui est intégrable sur cet intervalle (intégrale de Riemann de paramètre $1 - \alpha < 1$) donc f aussi.

La fonction f est continue sur $[1, +\infty[$ et elle est équivalente en $+\infty$ à $\frac{x^{\alpha-1}}{x} = \frac{1}{x^{2-\alpha}}$ qui est

intégrable sur cet intervalle (intégrale de Riemann de paramètre $2 - \alpha > 2 - 1 = 1$) donc f aussi.

Commentaires : Question de difficulté standard. La question étant une application simple du cours et traditionnelle, les correcteurs sont intransigeants sur la rigueur de l'argumentation. Si vous n'avez pas mentionné certains des éléments de ma réponse, retravaillez ce type de question. Elles sont traditionnelles aux concours (CCINP, Mines, Centrale) et les correcteurs suivent les mêmes exigences indépendamment des concours. Au final, cette question s'avère toujours sélective bien que traditionnelle et élémentaire !

Q10. On effectue le changement de variable $t = \frac{1}{x}$ qui réalise une bijection strictement décroissante de $[1, +\infty[$ sur $]0, 1]$. On a $x = \frac{1}{t}$ donc

$$dx = \left(\frac{1}{t}\right)' dt = -\frac{dt}{t^2}.$$

Quand $x = 1$, $t = 1$, quand $x \rightarrow +\infty$, $x \rightarrow 0$. On peut alors écrire :

$$\begin{aligned} J(\alpha) &= \int_1^{+\infty} \frac{x^{\alpha-1}}{1+x} dx = \int_1^0 \frac{\left(\frac{1}{t}\right)^{\alpha-1}}{1+\frac{1}{t}} \left(-\frac{dt}{t^2}\right) = \int_0^1 \frac{dt}{t^{\alpha-1} \times \frac{t+1}{t} \times t^2} \\ &= \int_0^1 \frac{dt}{t^\alpha (1+t)} = \int_0^1 \frac{t^{-\alpha}}{1+t} dt = I(1-\alpha). \end{aligned}$$

Commentaires : Question très discriminante car le candidat doit imaginer la stratégie (changement de variable, le deviner et le mettre convenablement en place).

Q11. D'après les séries géométriques, on a pour tout $x \in]0, 1[$,

$$\frac{1}{1+x} = \sum_{n=0}^{+\infty} (-x)^n = \sum_{n=0}^{+\infty} (-1)^n x^n \Rightarrow \frac{x^{\alpha-1}}{1+x} = \sum_{n=0}^{+\infty} (-1)^n x^{n+\alpha-1}.$$

Supposons que cette série converge uniformément sur $]0, 1[$. Pour chaque entier n , la fonction

$$f_n : x \mapsto (-1)^n x^{n+\alpha-1}$$

admet pour limite quand $x \rightarrow 1$ le nombre $(-1)^n$. Dans ce cas, le théorème de double limite (permutation limite-série) affirme que

$$\sum_{n=0}^{+\infty} \lim_{x \rightarrow 1} f_n(x) = \lim_{x \rightarrow 1} \sum_{n=0}^{+\infty} f_n(x) \Leftrightarrow \sum_{n=0}^{+\infty} (-1)^n = \lim_{x \rightarrow 1} \frac{1}{1+x} = \frac{1}{2}$$

ce qui est absurde car la série $\sum_n (-1)^n$ diverge (son terme général ne tend pas vers 0). Par conséquent, la série $\sum_n f_n$ ne converge pas uniformément sur $]0, 1[$.

Commentaires : La première question est élémentaire. La seconde est difficile et destinée aux meilleures copies.

Q12. Pour tout entier n , la fonction S_n est continue donc intégrable sur $[0, 1]$. Elle converge simplement sur $]0, 1[$ vers $f : x \mapsto \frac{x^{\alpha-1}}{1+x}$. Comme il s'agit d'une série alternée, le critère spécial des séries alternées montre que

$$\forall n \in \mathbb{N}, \forall x \in]0, 1[, |S_n(x)| \leq |f_0(x)| = x^{\alpha-1} = \frac{1}{x^{1-\alpha}} = \varphi(x).$$

La fonction φ est indépendante de n et intégrable sur $]0, 1[$ (intégrale de Riemann de paramètre $1 - \alpha < 1$). Par conséquent, on peut appliquer le théorème de convergence dominée d'où l'égalité :

$$\lim_{n \rightarrow \infty} \int_0^1 S_n(x) dx = \int_0^1 \lim_{n \rightarrow +\infty} S_n(x) dx = \int_0^1 \frac{x^{\alpha-1}}{1+x} dx = I(\alpha).$$

Or, pour tout entier n , par linéarité de l'intégrale, on a :

$$\int_0^1 S_n(x) dx = \sum_{k=0}^n (-1)^k \int_0^1 x^{k+\alpha-1} dx = \sum_{k=0}^n (-1)^k \left[\frac{x^{k+\alpha}}{k+\alpha} \right]_0^1 = \sum_{k=0}^n \frac{(-1)^k}{k+\alpha}.$$

Par conséquent, on a

$$\lim_{n \rightarrow +\infty} \int_0^1 S_n(x) dx = \sum_{k=0}^{+\infty} \frac{(-1)^k}{k+\alpha}.$$

Commentaires : Question plutôt difficile pour ce concours même si le sujet indique la méthode à suivre. Encore faut-il que le candidat connaisse le théorème de convergence dominée ... pour les séries (ou voit bien une série comme une suite de somme partielle) et parvienne à trouver une domination indépendante de n (en évitant de poser par exemple $\varphi(x) = n$, qui est fausse, ou $nx^{\alpha-1}$, qui est vraie, ... qui dépend de n , etc). Si vous avez bloqué sur cette question ou donné une réponse fausse, n'hésitez pas à la retravailler, elle intervient relativement fréquemment à ce concours.

Q13. Par la relation de Chasles, il est immédiat que

$$I(\alpha) + J(\alpha) = \int_0^{+\infty} \frac{x^{\alpha-1}}{1+x} dx.$$

Comme $J(\alpha) = I(1-\alpha)$ et en utilisant la formule établie à la question précédente, on a :

$$I(\alpha) + J(\alpha) = \sum_{k=0}^{+\infty} \frac{(-1)^k}{k+\alpha} + \sum_{k=0}^{+\infty} \frac{(-1)^k}{k+1-\alpha}.$$

En effectuant le changement de variable $p = k+1$ (donc $k = p-1$) dans la seconde somme, on obtient :

$$\sum_{k=0}^{+\infty} \frac{(-1)^k}{k+1-\alpha} = \sum_{p=1}^{+\infty} \frac{(-1)^{p-1}}{p-\alpha} = - \sum_{k=1}^{+\infty} \frac{(-1)^k}{k-\alpha}.$$

On peut alors écrire :

$$\begin{aligned}
 I(\alpha) + J(\alpha) &= \sum_{k=0}^{+\infty} \frac{(-1)^k}{k+\alpha} - \sum_{k=1}^{+\infty} \frac{(-1)^k}{k-\alpha} = \frac{(-1)^0}{0+\alpha} + \sum_{k=1}^{+\infty} \frac{(-1)^k}{k+\alpha} - \sum_{k=1}^{+\infty} \frac{(-1)^k}{k-\alpha} \\
 &= \frac{1}{\alpha} + \sum_{k=1}^{+\infty} (-1)^k \left[\frac{1}{k+\alpha} - \frac{1}{k-\alpha} \right] = \frac{1}{\alpha} + \sum_{k=1}^{+\infty} (-1)^k \frac{k-\alpha-(k+\alpha)}{(k+\alpha)(k-\alpha)} \\
 &= \frac{1}{\alpha} + \sum_{k=1}^{+\infty} (-1)^k \frac{-2\alpha}{k^2-\alpha^2} = \frac{1}{\alpha} + 2\alpha \sum_{k=1}^{+\infty} \frac{(-1)^k}{\alpha^2-k^2}
 \end{aligned}$$

Commentaires : La première égalité est élémentaire. Pour la seconde, le candidat doit maîtriser ses calculs et, comme la réponse est fournie par le sujet, les correcteurs seront très attentifs aux calculs. Ils sanctionneront lourdement les "bidouilles" et autres erreurs. Au final, la deuxième question est sélective.

Q14. En choisissant $x = 0$ dans la formule admise à la question précédente, comme $\cos(0) = 1$, on a

$$1 = \frac{\sin(\alpha\pi)}{\pi} \left(\frac{1}{\alpha} + 2\alpha \sum_{k=1}^{+\infty} \frac{(-1)^k}{\alpha^2 - k^2} \right) = \frac{\sin(\alpha\pi)}{\pi} \int_0^{+\infty} \frac{x^{\alpha-1}}{1+x} dx$$

d'où la formule attendue.

Commentaires : Question destinée aux bonnes copies. Le candidat doit être apte à faire une synthèse des différents résultats prouvés ou admis pour le sujet et faire des choix judicieux.

Q15. Soit $x \in]0, +\infty[$. La fonction $g : t \mapsto e^{-t}t^{x-1}$ est continue sur $]0, +\infty[$. Son équivalent en 0 est $t^{x-1} = \frac{1}{t^{1-x}}$ qui est positive et intégrable sur $]0, 1]$ (intégrale de Riemann de paramètre $1-x < 1$) donc g l'est aussi. On dispose également de la domination suivante en $+\infty$: $g(t) \underset{t \rightarrow +\infty}{=} o\left(\frac{1}{t^2}\right)$ car $t^2g(t) = t^{x+1}e^{-t} \xrightarrow{t \rightarrow +\infty} 0$ (d'après les croissances comparées). La fonction $t \mapsto \frac{1}{t^2}$ étant positive et intégrable sur $[1, +\infty[$ (intégrale de Riemann de paramètre $2 > 1$) donc g l'est aussi. Par conséquent, $\int_0^{+\infty} g$ converge ce qui assure l'existence de $\Gamma(x)$.

Commentaires : Il s'agit d'une question de cours (la fonction Γ et son domaine de définition sont au programme officiel). La question étant une application simple du cours et traditionnelle, les correcteurs sont intransigeants sur la rigueur de l'argumentation. Si vous n'avez pas mentionné certains des éléments de ma réponse, retravaillez ce type de question. Elles sont traditionnelles aux concours (CCINP, Mines, Centrale) et les correcteurs suivent les mêmes exigences indépendamment des concours. Au final, cette question s'avère toujours sélective bien que traditionnelle et élémentaire !

Q16. Existence. Soit $x \in]0, +\infty[$. On applique les mêmes arguments qu'à la question précédente avec la $g : t \mapsto \frac{t^{\alpha-1}}{1+t} e^{-xt}$ (son équivalent en 0 est $t^{\alpha-1}$ et elle est négligeable en $+\infty$

devant $\frac{1}{t^2}$) donc f_α est bien définie sur $]0, +\infty[$.

Continuité. Considérons la fonction $h : (x, t) \mapsto \frac{t^{\alpha-1}}{1+t} e^{-xt}$. Pour chaque $x > 0$, la fonction $t \mapsto h(x, t)$ est continue sur $]0, +\infty[$. Pour chaque $t > 0$, la fonction $x \mapsto h(x, t)$ est continue sur $]0, +\infty[$. Vérifions la domination. Soit $[a, b]$ un segment de $]0, +\infty[$ (i.e. $a > 0$) alors

$$\forall x \in [a, b], \forall t \in]0, +\infty[, |h(x, t)| \leq \frac{t^{\alpha-1}}{1+t} e^{-at} = h(a, t).$$

La fonction $t \mapsto h(a, t)$ est indépendante de x et intégrable sur $]0, +\infty[$ (d'après l'existence) donc la fonction

$$x \mapsto \int_0^{+\infty} h(x, t) dt = f_\alpha(x)$$

est continue sur $]0, +\infty[$.

Commentaires : Même remarque que précédemment. Pensez à faire des dominations par rapport à x uniquement sur des segments. Cela distinguera très fortement les candidats entre eux.

Q17. Considérons la fonction $h : (x, t) \mapsto \frac{t^{\alpha-1}}{1+t} e^{-xt}$. Pour chaque $t > 0$, la fonction $x \mapsto h(x, t)$ est de classe C^1 sur $]0, +\infty[$. Pour chaque $x > 0$, les fonctions $t \mapsto h(x, t)$ et

$$t \mapsto \frac{\partial h}{\partial x}(x, t) = -\frac{t^\alpha}{1+t} e^{-xt}$$

sont continues sur $]0, +\infty[$. Pour chaque $x > 0$, l'intégrale $\int_0^{+\infty} h(x, t) dt$ converge. Vérifions la domination. Soit $[a, b]$ un segment de $]0, +\infty[$ (i.e. $a > 0$) alors

$$\forall x \in [a, b], \forall t \in]0, +\infty[, \left| \frac{\partial h}{\partial x}(x, t) \right| \leq \frac{t^\alpha}{1+t} e^{-at} = \varphi(t).$$

La fonction φ est indépendante de x et intégrable sur $]0, +\infty[$ (d'après l'existence de f_α en remplaçant par α par $\alpha + 1$ qui reste supérieur à -1) donc la fonction

$$x \mapsto \int_0^{+\infty} h(x, t) dt = f_\alpha(x)$$

est de classe C^1 sur $]0, +\infty[$. Sa dérivée est définie par :

$$\forall x > 0, f'_\alpha(x) = \int_0^{+\infty} \frac{\partial h}{\partial x}(x, t) dt = - \int_0^{+\infty} \frac{t^\alpha}{1+t} e^{-xt} dt.$$

Commentaires : Même remarque que précédemment.

Q18. Considérons la fonction $h : (x, t) \mapsto \frac{t^{\alpha-1}}{1+t} e^{-xt}$. Pour chaque $x > 0$, la fonction $t \mapsto h(x, t)$ est continue sur $]0, +\infty[$. Pour chaque $t > 0$, $\lim_{x \rightarrow +\infty} h(x, t) = 0$. Vérifions la domination.

$$\forall x \in [1, +\infty[, \forall t \in]0, +\infty[, |h(x, t)| \leq \frac{t^{\alpha-1}}{1+t} e^{-t} = h(1, t).$$

La fonction $t \mapsto h(1, t)$ est indépendante de x et intégrable sur $]0, +\infty[$ (d'après l'existence de f_α) donc, d'après le théorème de convergence dominée, on a :

$$\lim_{x \rightarrow +\infty} f_\alpha(x) = \lim_{x \rightarrow +\infty} \int_0^{+\infty} h(x, t) dt = \int_0^{+\infty} \lim_{x \rightarrow +\infty} h(x, t) dt = \int_0^{+\infty} 0 dt = 0.$$

Commentaires : Même remarque que précédemment. Par contre, il ne faut pas faire de domination par rapport à x sur les segments car, sur un segment donné, x ne peut pas tendre vers $+\infty$! Cette erreur est très classique aux concours.

Q19. La fonction $g : t \mapsto \frac{e^{-t}}{t^\alpha}$ est continue sur $]0, +\infty[$. Son équivalent en 0 est $\frac{1}{t^\alpha}$ qui est positive et intégrable sur $]0, 1]$ (intégrale de Riemann de paramètre $\alpha < 1$) donc g l'est aussi. On dispose également de la domination suivante en $+\infty$: $g(t) \underset{t \rightarrow +\infty}{=} o\left(\frac{1}{t^2}\right)$ car $t^2 g(t) = t^{2-\alpha} e^{-t} \xrightarrow[t \rightarrow +\infty]{} 0$ (d'après les croissances comparées). La fonction $t \mapsto \frac{1}{t^2}$ étant positive et intégrable sur $[1, +\infty[$ (intégrale de Riemann de paramètre $2 > 1$) donc g l'est aussi d'où son intégrabilité sur $]0, +\infty[$. Par conséquent, le reste de son intégrale tend vers 0 en $+\infty$ c'est-à-dire :

$$\lim_{x \rightarrow +\infty} \int_x^{+\infty} \frac{e^{-t}}{t^\alpha} dt = 0.$$

Commentaires : Question de difficulté standard. La question étant une application simple du cours et traditionnelle, les correcteurs sont intransigeants sur la rigueur de l'argumentation. Si vous n'avez pas mentionné certains des éléments de ma réponse, retravaillez ce type de question. Elles sont traditionnelles aux concours (CCINP, Mines, Centrale) et les correcteurs suivent les mêmes exigences indépendamment des concours. Au final, cette question s'avère toujours sélective bien que traditionnelle et élémentaire ! Remarquez que les candidats étant peu à l'aise avec l'intégration n'était pas à la fête dans ce sujet... Cela peut arriver donc vous devez être prêt à maîtriser les fondamentaux des chapitres fondamentaux du programme de deuxième année.

Q20. Soit $x > 0$. D'après la question 17, on a :

$$f_\alpha(x) - f'_\alpha(x) = \int_0^{+\infty} \frac{e^{-xt}}{1+t} (t^{\alpha-1} + t^\alpha) dt = \int_0^{+\infty} \frac{t^{\alpha-1} e^{-xt}}{1+t} (1+t) dt = \int_0^{+\infty} t^{\alpha-1} e^{-xt} dt.$$

En effectuant le changement de variable $u = xt$ ($t = \frac{u}{x}$, $dt = \frac{du}{x}$, $t \rightarrow 0$ alors $u \rightarrow 0$, $t \rightarrow +\infty$

alors $u \rightarrow +\infty$) à cette dernière intégrale, on obtient

$$f_\alpha(x) - f'_\alpha(x) = \int_0^{+\infty} \left(\frac{u}{x}\right)^{\alpha-1} e^{-u} \frac{du}{x} = \frac{1}{x^\alpha} \int_0^{+\infty} u^{\alpha-1} e^{-u} du = \frac{\Gamma(\alpha)}{x^\alpha}.$$

Commentaires : Question de difficulté standard. Le changement de variable sera probablement discriminant entre les candidats (le sujet ne le mentionne pas).

Q21. La fonction

$$H : x \mapsto \int_x^{+\infty} \frac{e^{-t}}{t^\alpha} dt = - \int_{+\infty}^x \frac{e^{-t}}{t^\alpha} dt$$

est une primitive de la fonction $h : x \mapsto -\frac{e^{-x}}{x^\alpha}$ donc la dérivée de H est h . Comme $g_\alpha = \Gamma(\alpha) e^x H$, on peut écrire pour tout réel $x > 0$:

$$\begin{aligned} g'_\alpha(x) &= \Gamma(\alpha) e^x H(x) + \Gamma(\alpha) e^x H'(x) = g_\alpha(x) + \Gamma(\alpha) e^x \left(-\frac{e^{-x}}{x^\alpha}\right) \\ &= g_\alpha(x) - \frac{\Gamma(\alpha)}{x^\alpha} \Rightarrow g_\alpha(x) - g'_\alpha(x) = \frac{\Gamma(\alpha)}{x^\alpha}. \end{aligned}$$

Posons $\varphi = f_\alpha - g_\alpha$ qui est de classe C^1 sur $]0, +\infty[$ et vérifie l'équation différentielle

$$\varphi - \varphi' = f_\alpha - f'_\alpha - (g_\alpha - g'_\alpha) = \frac{\Gamma(\alpha)}{x^\alpha} - \frac{\Gamma(\alpha)}{x^\alpha} = 0.$$

Par conséquent, il existe un réel C tel que

$$\forall x > 0, \varphi(x) = C e^x.$$

Or, les fonctions f_α et g_α tendent vers 0 en $+\infty$ donc φ aussi. Comme $\lim_{x \rightarrow +\infty} e^x = +\infty$, la seule possibilité est que

$$C = 0 \Leftrightarrow \varphi = 0 \Leftrightarrow f_\alpha = g_\alpha$$

sur l'intervalle $]0, +\infty[$.

Commentaires : Le point clé de la première question est de penser systématiquement au mot primitive lorsque l'on est face à une fonction de la forme $x \mapsto \int_a^x f$. Si cela ne vous est pas venu à l'esprit, réviser ce résultat de MPSI et vous devez l'avoir à votre esprit toujours, écrit comme oral de tous les concours. Sans lui, un nombre considérable de parties de sujet vous échappera alors que les questions sont rudimentaires. C'est le cas ici. La question est au finale très discriminante. La seconde question est destinée aux bonnes copies (voire très bonnes). Un nombre significatif de candidats invoque l'unicité des solutions d'équations différentielle ... mais il faut l'égalité des conditions initiales ! Ici, nous n'en disposons pas. Il faut alors se ramener à une équation homogène pour avancer.

Q22. Comme $f_\alpha = g_\alpha$ (question précédente), que la fonction f_α est continue en 0 (cf. question 16) et

$$\lim_{x \rightarrow 0} g_\alpha = \Gamma(\alpha) \int_0^{+\infty} \frac{e^{-t}}{t^\alpha} dt$$

(car cette dernière intégrale converge) donc, en faisant tendre x vers 0 dans la relation $f_\alpha = g_\alpha$, on obtient l'égalité voulue.

Commentaires : Question de difficulté standard.

Q23. En utilisant les formules des questions 14 et 22, on a :

$$\begin{aligned} \frac{\pi}{\sin(\pi\alpha)} &= \int_0^{+\infty} \frac{x^{\alpha-1}}{1+x} dx = \Gamma(\alpha) \int_0^{+\infty} \frac{e^{-t}}{t^\alpha} dt = \Gamma(\alpha) \int_0^{+\infty} e^{-t} t^{-\alpha} dt \\ &= \Gamma(\alpha) \int_0^{+\infty} e^{-t} t^{(1-\alpha)-1} dt = \Gamma(\alpha) \Gamma(1-\alpha). \end{aligned}$$

Commentaires : Question très discriminante car il s'agit d'une question de synthèse.

Q24. On choisit $\alpha = \frac{1}{2}$ dans la formule de la question précédente donc

$$\left(\Gamma\left(\frac{1}{2}\right) \right)^2 = \pi \Leftrightarrow \Gamma\left(\frac{1}{2}\right) = \sqrt{\pi}$$

(car Γ est positive). En faisant le changement de variable $x = t^2$ dans l'intégrale souhaitée ($t = \sqrt{x} = x^{1/2}$, $dt = \frac{1}{2}x^{-1/2} = \frac{1}{2\sqrt{x}}$, $t = 0$ alors $x = 0$, $t \rightarrow +\infty$ alors $x \rightarrow +\infty$), on obtient :

$$\int_0^{+\infty} e^{-t^2} dt = \int_0^{+\infty} e^{-x} \frac{dx}{2\sqrt{x}} = \frac{1}{2} \Gamma\left(\frac{1}{2}\right) = \frac{\sqrt{\pi}}{2}.$$

Commentaires : Question destinée aux bonnes copies (autonomie, initiative et technicité du candidat sont simultanément requises).

ÉPREUVE SPÉCIFIQUE - FILIÈRE MP

MATHÉMATIQUES 2

Durée : 4 heures

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de deux exercices et d'un problème tous indépendants.

EXERCICE 1

On note $E = \mathbb{R}_2[X]$.

Dans cet exercice, on pourra utiliser sans démonstration que pour tout entier naturel n , la fonction $x \mapsto x^n e^{-x}$ est intégrable sur $[0, +\infty[$ et $\int_0^{+\infty} x^n e^{-x} dx = n!$.

Q1. Démontrer que l'on définit un produit scalaire sur E en posant, pour tout couple (P, Q) de polynômes de E , $\langle P | Q \rangle = \int_0^{+\infty} P(x)Q(x)e^{-x} dx$. On notera $\| \cdot \|$ la norme euclidienne associée.

Q2. Déterminer le projeté orthogonal de X^2 sur $F = \mathbb{R}_1[X]$ noté $P_F(X^2)$.

Q3. Justifier que $\|X^2 - P_F(X^2)\|^2 = \|X^2\|^2 - \|P_F(X^2)\|^2$ puis calculer le réel :

$$\inf_{(a,b) \in \mathbb{R}^2} \int_0^{+\infty} (x^2 - ax - b)^2 e^{-x} dx.$$

EXERCICE 2

Soit $p \in]0, 1[$, $q = 1 - p$. Soit X et Y deux variables aléatoires indépendantes à valeurs dans $\mathbb{N}$ définies sur un même espace probabilisé et suivant la même loi définie par :

$$\forall k \in \mathbb{N}, P(X = k) = P(Y = k) = pq^k.$$

On considère les variables aléatoires Z et T définies par $Z = \sup(X, Y)$ et $T = \inf(X, Y)$.

Q4. Pour tout couple (m, n) d'entiers naturels, déterminer $P((Z = m) \cap (T = n))$ en distinguant trois cas : $m > n$, $m < n$ et $m = n$.

Q5. En déduire la loi de la variable aléatoire Z .

PROBLÈME

Dans ce problème, E est un $\mathbb{C}$ -espace vectoriel de dimension finie.

Partie I

Q6. Un exemple

Vérifier que la matrice $A = \begin{pmatrix} 3 & 2 \\ 2 & 3 \end{pmatrix}$ est diagonalisable.

Démontrer que les matrices $\Pi_1 = \frac{1}{2} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}$ et $\Pi_2 = \frac{1}{2} \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}$ sont des matrices de projecteur puis calculer $\Pi_1 + 5\Pi_2$, $\Pi_1 + \Pi_2$ et $\Pi_1\Pi_2$.

Q7. On rappelle le lemme de décomposition des noyaux :

Si $P_1, P_2, \dots, P_r$ sont des éléments de $\mathbb{C}[X]$ deux à deux premiers entre eux de produit égal à T , si u est un endomorphisme de E alors :

$$\text{Ker}[T(u)] = \text{Ker}(P_1(u)) \oplus \text{Ker}(P_2(u)) \oplus \dots \oplus \text{Ker}(P_r(u)).$$

L'objet de cette question est de démontrer le cas particulier $r = 2$.

Soit u un endomorphisme de E et soit P et Q deux polynômes premiers entre eux.

Justifier que $\text{Ker}(P(u)) \subset \text{Ker}[(PQ)(u)]$ (de même, on a : $\text{Ker}(Q(u)) \subset \text{Ker}[(PQ)(u)]$).

Démontrer que : $\text{Ker}[(PQ)(u)] = \text{Ker}(P(u)) \oplus \text{Ker}(Q(u))$.

Dans la suite du problème, on pourra utiliser librement le lemme de décomposition des noyaux.

Q8. Soit u un endomorphisme de E et soit π_u son polynôme minimal.

On suppose que $\pi_u = P_1^{k_1} P_2^{k_2}$ où les polynômes P_1 et P_2 sont premiers entre eux. On pose, pour tout entier $i \in \{1, 2\}$, $Q_i = \frac{\pi_u}{P_i^{k_i}}$.

Justifier qu'il existe deux polynômes R_1 et R_2 de $\mathbb{C}[X]$ tels que $R_1 Q_1 + R_2 Q_2 = 1$.

Pour la suite de cette partie, on notera $\pi_u = P_1^{k_1} P_2^{k_2} \dots P_m^{k_m}$ la décomposition en facteurs premiers du polynôme minimal et on admettra que, si pour tout entier $i \in \{1, 2, \dots, m\}$, $Q_i = \frac{\pi_u}{P_i^{k_i}}$, il existe des polynômes de $\mathbb{C}[X]$ tels que $R_1 Q_1 + R_2 Q_2 + \dots + R_m Q_m = 1$.

Q9. On pose alors pour tout entier $i \in \{1, 2, \dots, m\}$, $p_i = R_i(u) \circ Q_i(u)$.

Démontrer que pour tout couple (i, j) d'entiers distincts de $\{1, 2, \dots, m\}$, on a les trois résultats suivants :

$$p_i \circ p_j = 0,$$

$$\sum_{i=1}^m p_i = id_E,$$

et chaque p_i est un projecteur de E .

Les p_i seront appelés projecteurs associés à u .

Q10. Soit u un endomorphisme de E et soit χ_u son polynôme caractéristique : $\chi_u = \prod_{i=1}^m (X - \lambda_i)^{\alpha_i}$

(avec les λ_i deux à deux distincts et les α_i des entiers naturels non nuls) et pour tout entier

$i \in \{1, 2, \dots, m\}$, $N_i = \text{Ker}(u - \lambda_i id_E)^{\alpha_i}$ le sous-espace caractéristique associé à λ_i .

Justifier que $E = N_1 \oplus N_2 \oplus \dots \oplus N_m$.

Q11. Démontrer que $E = \text{Im } p_1 \oplus \text{Im } p_2 \oplus \dots \oplus \text{Im } p_m$.

Q12. Démontrer que pour tout entier $i \in \{1, 2, \dots, m\}$, $\text{Im } p_i = N_i$.

Partie II

Dans toute cette partie, on suppose que l'endomorphisme u est diagonalisable et on note $\lambda_1, \lambda_2, \dots, \lambda_m$ ses valeurs propres distinctes.

Q13. Quel est alors le polynôme minimal π_u de u ?

Q14. On note toujours, pour tout entier $i \in \{1, 2, \dots, m\}$, $Q_i = \frac{\pi_u}{P_i}$ où $P_i = X - \lambda_i$, et on pose

$$\theta_i = \frac{1}{Q_i(\lambda_i)}.$$

Donner, sans détails, la décomposition en éléments simples de $\frac{1}{\pi_u}$ puis démontrer que les

projecteurs associés à u sont, pour tout entier $i \in \{1, 2, \dots, m\}$, $p_i = \frac{Q_i(u)}{Q_i(\lambda_i)}$.

Q15. Démontrer que $X = \sum_{i=1}^m \frac{\lambda_i Q_i(X)}{Q_i(\lambda_i)}$ puis que $u = \sum_{i=1}^m \lambda_i p_i$ (décomposition spectrale de u).

Q16. Exemple : on considère la matrice $A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ 1 & -1 & 1 & -1 \\ 1 & -1 & -1 & 1 \end{pmatrix}$.

- a) Justifier que la matrice A est diagonalisable et calculer la matrice A^2 .
- b) En déduire le polynôme minimal π_A de la matrice A puis la décomposition spectrale de la matrice A . On notera Π_1 et Π_2 les matrices des projecteurs associés.
- c) Calculer, pour tout entier naturel q , A^q en fonction des matrices Π_1 et Π_2 .

Q17. On note $\mathbb{C}[v]$ l'algèbre des polynômes d'un endomorphisme v d'un $\mathbb{C}$ -espace vectoriel de dimension finie.

Démontrer que la dimension de l'espace vectoriel $\mathbb{C}[v]$ est égal au degré du polynôme minimal π_v de l'endomorphisme v .

Q18. On revient au cas u diagonalisable avec $\pi_u = \prod_{i=1}^m (X - \lambda_i)$.

Démontrer que la famille $(p_1, p_2, \dots, p_m)$ des projecteurs associés à u est une base de l'espace vectoriel $\mathbb{C}[u]$.

Q19. Dans le cas d'un endomorphisme u non diagonalisable, la famille $(p_1, p_2, \dots, p_m)$ des projecteurs associés à u est-elle toujours une base de l'espace vectoriel $\mathbb{C}[u]$?

Q20. Nous avons vu que si u est un endomorphisme de E diagonalisable, il existe m endomorphismes non nuls p_i de E , tels que pour tout entier q on ait $u^q = \sum_{i=1}^m \lambda_i^q p_i$.

Nous allons étudier une « réciproque ».

Soit u un endomorphisme de E , $\mathbb{C}$ -espace vectoriel de dimension finie. On suppose qu'il existe m endomorphismes non nuls f_i de E et m complexes $\lambda_1, \lambda_2, \dots, \lambda_m$ distincts, tels que pour tout

entier naturel q on ait $u^q = \sum_{i=1}^m \lambda_i^q f_i$.

Démontrer que u est diagonalisable.

FIN

SOLUTION

Q1. Soient P et Q appartenant à $\mathbb{R}[X]$.

Existence. La fonction $x \mapsto P(x)Q(x)e^{-x}$ est intégrable sur $\mathbb{R}_+$ comme combinaison de linéaires de fonctions $x \mapsto x^n e^{-x}$ ($n \in \mathbb{N}$) qui le sont (d'après l'hypothèse de cet exercice) donc $\langle P | Q \rangle$ existe.

Symétrie. Comme

$$\langle P | Q \rangle = \int_0^{+\infty} P(x)Q(x)e^{-x}dx = \int_0^{+\infty} Q(x)P(x)e^{-x}dx = \langle Q | P \rangle,$$

on peut affirmer que $\langle | \rangle$ est symétrique.

Bilinéarité. Pour tout R appartenant à $\mathbb{R}[X]$ et tous réels λ, μ , on a :

$$\begin{aligned} \langle \lambda P + \mu Q | R \rangle &= \int_0^{+\infty} (\lambda P(x) + \mu Q(x))R(x)e^{-x}dx \\ &= \int_0^{+\infty} (\lambda P(x)R(x)e^{-x} + \mu Q(x)R(x)e^{-x})dx \\ &= \lambda \int_0^{+\infty} P(x)R(x)e^{-x}dx + \mu \int_0^{+\infty} Q(x)R(x)e^{-x}dx \\ &= \lambda \langle P | R \rangle + \mu \langle Q | R \rangle \end{aligned}$$

donc $P \mapsto \langle P | R \rangle$ est linéaire. Par symétrie de $\langle | \rangle$, $R \mapsto \langle P | R \rangle$ l'est aussi donc $\langle | \rangle$ est bilinéaire.

Positivité. $\langle P | P \rangle = \int_0^{+\infty} (P(x))^2 e^{-x} dx \geq 0$ (par positivité de la fonction à intégrer).

Définie. Si $\langle P | P \rangle = 0$ alors, la fonction $f : x \mapsto (P(x))^2 e^{-x}$ étant positive, continue et d'intégrale nulle sur $[0, +\infty[$, elle est nulle sur cet intervalle c'est-à-dire :

$$\forall x \in \mathbb{R}_+, (P(x))^2 e^{-x} = 0 \Leftrightarrow \forall x \in \mathbb{R}, P(x) = 0$$

(car e^{-x} ne s'annule pas sur $\mathbb{R}$). Ainsi, P admet une infinité de racines (tous les réels positifs) donc P est nul d'où le caractère défini de $\langle | \rangle$.

Conclusion : $\langle | \rangle$ est bien un produit scalaire.

Commentaires : Question de niveau standard. La difficulté pour les candidats est de ... se souvenir de toutes les propriétés à vérifier pour avoir un produit scalaire ! L'absence de rédaction de l'existence de $\langle P | Q \rangle$ sera fortement sanctionnée (soit une preuve directe soit en mentionnant le rappel du sujet : $x \mapsto x^n e^{-x}$ est intégrable). La séparation mal rédigée ou pas rédigée sera aussi lourdement sanctionnée. Au final, la question s'avère habituellement très discriminante. Dommage pour une question classique et traditionnelle. Si c'est votre cas, retravaillez la.

Q2. Notons P ce projeté. Par définition, il appartient à $\mathbb{R}_1[X]$, donc il s'écrit sous la forme $P = aX + b$, et $X^2 - P$ est orthogonal à $\mathbb{R}_1[X]$ c'est-à-dire qu'il est orthogonal à 1 et à X

(famille génératrice de cet espace vectoriel). Par conséquent, on a le système suivant :

$$\begin{cases} \langle X^2 - aX - b \mid 1 \rangle = 0 \\ \langle X^2 - aX - b \mid X \rangle = 0 \end{cases} \Leftrightarrow \begin{cases} a \langle X \mid 1 \rangle + b \langle 1 \mid 1 \rangle = \langle X^2 \mid 1 \rangle \\ a \langle X \mid X \rangle + b \langle 1 \mid X \rangle = \langle X^2 \mid X \rangle \end{cases}$$

(par linéarité du produit scalaire). Calculons ces différents produits scalaires en utilisant la

formule admise par le sujet $\int_0^{+\infty} x^n e^{-x} dx = n!$.

$$\begin{aligned} \langle 1 \mid 1 \rangle &= \int_0^{+\infty} e^{-x} dx = 0! = 1, \quad \langle X \mid 1 \rangle = \int_0^{+\infty} x e^{-x} dx = 1! = 1 \\ \langle X \mid X \rangle &= \int_0^{+\infty} x^2 e^{-x} dx = 2! = 2, \quad \langle X^2 \mid 1 \rangle = \int_0^{+\infty} x^2 e^{-x} dx = 2! = 2 \\ \langle X^2 \mid X \rangle &= \int_0^{+\infty} x^3 e^{-x} dx = 3! = 6. \end{aligned}$$

Nous disposons alors du système

$$\begin{cases} a + b = 2 \\ 2a + b = 6 \end{cases} \Leftrightarrow \begin{cases} L_2 - L_1 : a = 4 \\ 2L_1 - L_2 : b = -2 \end{cases}$$

donc $P = 4X - 2$ est le projeté orthogonal de X^2 sur $\mathbb{R}_1[X]$.

Commentaires : Cette question peut s'aborder comme dans le corrigé ci-dessus ou bien utiliser la forme $P = \langle P \mid \varepsilon_0 \rangle \varepsilon_0 + \langle P \mid \varepsilon_1 \rangle \varepsilon_1$. Par contre, le piège est de ne pas prendre $\varepsilon_0 = 1$ et $\varepsilon_1 = X$ mais une base orthonormée, via le procédé de Gram-Schmidt. La question étant assez calculatoire, songez à être soigneux dans les calculs. Un nombre significatif de candidats n'ont pas d'idée pour déterminer un projeté orthogonal ! La question s'avère très discriminante bien que classique et élémentaire.

Q3. Comme $X^2 - P_F(X^2)$ est orthogonal à $\mathbb{R}_1[X]$ et que $P_F(X^2)$ appartient à $\mathbb{R}_1[X]$, ces deux vecteurs sont orthogonaux donc, par le théorème de Pythagore, on a :

$$\|X^2\|^2 = \|(X^2 - P_F(X^2)) + P_F(X^2)\|^2 = \|X^2 - P_F(X^2)\|^2 + \|P_F(X^2)\|^2$$

d'où la formule attendue.

Par définition de la norme $\|\cdot\|$ associée au produit scalaire $\langle \cdot \mid \cdot \rangle$, on a :

$$\begin{aligned} \inf_{(a,b) \in \mathbb{R}^2} \int_0^{+\infty} (x^2 - ax - b)^2 e^{-x} dx &= \inf_{(a,b) \in \mathbb{R}^2} \|X^2 - aX - b\|^2 \\ &= \inf_{P \in \mathbb{R}_1[X]} \|X^2 - P\|^2 \end{aligned}$$

qui est par définition, le carré de la distance de X^2 à $\mathbb{R}_1[X]$. Comme $\mathbb{R}_1[X]$ est un espace vectoriel de dimension finie, le cours sur les espaces euclidiens montre que cette distance est

atteinte en le projeté orthogonal de X^2 sur $\mathbb{R}_1[X]$ c'est-à-dire que

$$\begin{aligned} \inf_{P \in \mathbb{R}_1[X]} \|X^2 - P\|^2 &= \|X^2 - P_F(X^2)\|^2 = \|X^2\|^2 - \|P_F(X^2)\|^2 \\ &= \int_0^{+\infty} x^4 e^{-x} dx - \int_0^{+\infty} (4x - 2)^2 e^{-x} dx \\ &= 4! - 16 \int_0^{+\infty} x^2 e^{-x} dx + 16 \int_0^{+\infty} x e^{-x} dx - 4 \int_0^{+\infty} e^{-x} dx \\ &= 24 - 16 \times 2 + 16 - 4 = 4 \end{aligned}$$

donc

$$\inf_{(a,b) \in \mathbb{R}^2} \int_0^{+\infty} (x^2 - ax - b)^2 e^{-x} dx = 4.$$

Commentaires : La première question pose problème à un nombre important de candidats. La seconde est destinée en pratique aux candidats solides sur le chapitre espaces euclidiens (faire le lien avec la notion de distance, utiliser la formule avec le projeté et finaliser les calculs).

Q4. Z est la plus grande valeur entre X et Y , T la plus petite valeur.

Premier cas. Si $n < m$ alors l'événement $(Z = m) \cap (T = n)$ signifie que la plus petite valeur est n et la plus grande est m . Cela se produit uniquement si $X = n$ et $Y = m$ ou si $X = n$ et $Y = m$ donc

$$\begin{aligned} P((Z = m) \cap (T = n)) &= P([(X = n) \cap (Y = m)] \cup [(X = m) \cap (Y = n)]) \\ &= P([(X = n) \cap (Y = m)]) + P([(X = m) \cap (Y = n)]) \quad (*) \\ &= P(X = n) P(Y = m) + P(X = m) P(Y = n) \quad (**) \\ &= pq^n pq^m + pq^m pq^n = 2p^2 q^{n+m} \end{aligned}$$

(*) car les deux évènements sont incompatibles.

(**) car les variables X et Y sont présumées indépendantes.

Deuxième cas. Si $n = m$ alors l'événement $(Z = n) \cap (T = n)$ signifie que la plus petite valeur est n et la plus grande est n donc les variables X et Y prennent la même valeur n .

$$\begin{aligned} P((Z = n) \cap (T = n)) &= P((X = n) \cap (Y = n)) = P(X = n) P(Y = n) \quad (**) \\ &= pq^n pq^n = p^2 q^{2n} \end{aligned}$$

Troisième cas. Si $m < n$ alors l'événement $(Z = m) \cap (T = n)$ signifie que la plus petite valeur est n et la plus grande est m , ce qui est absurde car $m < n$! Par conséquent, cet événement est impossible donc sa probabilité est nulle c'est-à-dire

$$P((Z = n) \cap (T = n)) = 0.$$

Conclusion : nous avons établi la formule

$$P((Z = m) \cap (T = n)) = \begin{cases} 0 & \text{si } m < n \\ p^2 q^{2n} & \text{si } n = m \\ 2p^2 q^{n+m} & \text{si } n < m \end{cases}.$$

Commentaires : Question relativement difficile pour ce concours. Le candidat est confronté à différentes difficultés : réinterpréter $(Z = m) \cap (T = n)$ à l'aide des variables X et Y ; distinguer les cas selon m et n ; justifier l'apparition des sommes et produits (incompatible et indépendance) indispensable à ce niveau d'étude.

Au final, cette question distinguera clairement les candidats selon la maîtrise de ces trois points ci-dessus. Soyez rigoureux et ne vous lancer pas dans des calculs sans réfléchir. C'est ce qui est attendu d'un ingénieur ou d'un scientifique de haut niveau. C'est ce qui est testé dans ces exercices des sujets de CCINP.

Q5. Comme X et Y prennent toutes les valeurs possibles de $\mathbb{N}$, Z aussi c'est-à-dire $Z(\Omega) = \mathbb{N}$. Soit $m \in \mathbb{N}$. D'après la formule donnant la loi marginale à l'aide de la loi conjointe, on a :

$$P(Z = m) = \sum_{n=0}^{+\infty} P((Z = n) \cap (T = n))$$

Tous les termes de cette somme sont nuls si $m < n$ donc il reste les termes $n \leq m$. On distingue le cas $n = m$ des cas $n \in \{0, \dots, m-1\}$. Pour que ces deux cas apparaissent, il faut que m soit supérieur à 1. Si $m = 0$, seul le premier cas peut se produire.

$$P(Z = 0) = P((Z = 0) \cap (T = 0)) = p^2 q^{2 \times 0} = p^2$$

et si $m \geq 1$ alors

$$\begin{aligned} P(Z = m) &= P((Z = m) \cap (T = m)) + \sum_{n=0}^{m-1} P((Z = m) \cap (T = n)) \\ &= p^2 q^{2m} + \sum_{n=0}^{m-1} 2p^2 q^{n+m} = p^2 q^{2m} + 2p^2 q^m \sum_{n=0}^{m-1} q^n \\ &= p^2 q^{2m} + 2p^2 q^m \frac{1 - q^m}{1 - q} = p^2 q^{2m} + 2p^2 q^m \frac{1 - q^m}{p} \\ &= p^2 q^{2m} + 2pq^m - 2pq^{2m}. \end{aligned}$$

Remarquons que cette formule reste vraie si $m = 0$! (on a de la chance).

Commentaires : Remarques similaires à la question précédente.

Q6. La matrice A est symétrique à coefficients réels donc elle est diagonalisable. Un calcul direct montre que $\Pi_1^2 = \Pi_1$, $\Pi_2^2 = \Pi_2$ donc Π_1 et Π_2 sont bien des projecteurs. En outre, on a :

$$\Pi_1 + \Pi_2 = I_2, \Pi_1 + 5\Pi_2 = A, \Pi_1 \Pi_2 = 0_2.$$

Commentaires : Question élémentaire sans difficulté particulière.

Q7. Soit $x \in \ker(P(u))$ c'est-à-dire $P(u)(x) = 0_E$. Comme $PQ = QP$ (les polynômes commutent), on a :

$$(PQ)(u) = (QP)(u) = Q(u) \circ P(u)$$

(d'après le cours) donc

$$(PQ)(u)(x) = Q(u)(P(u)(x)) = Q(u)(0_E) = 0_E$$

(car $Q(u)$ est un endomorphisme) d'où $x \in \ker((PQ)(u))$, ce qui prouve l'inclusion souhaitée.

Puisque P et Q sont premiers entre eux, le théorème de Bezout montre l'existence de deux polynômes A et B tels que

$$\begin{aligned} AP + BQ &= 1 \Rightarrow (AP)(u) + (BQ)(u) = \text{Id} \\ \Leftrightarrow A(u) \circ P(u) + B(u) \circ Q(u) &= \text{Id}. \end{aligned}$$

En évaluant en $x \in E$, on obtient l'égalité

$$x = A(u)(P(u)(x)) + B(u)(Q(u)(x))$$

En particulier, si $x \in \ker(P(u)) \cap \ker(Q(u))$, on a $P(u)(x) = 0_E$ et $Q(u)(x) = 0$ donc

$$x = A(u)(P(u)(x)) + B(u)(Q(u)(x)) = A(u)(0_E) + B(u)(0_E) = 0_E.$$

Nous venons de prouver l'égalité ensembliste

$$\ker(P(u)) \cap \ker(Q(u)) = \{0\}$$

donc la somme $\ker(P(u)) + \ker(Q(u))$ est directe.

En outre, si $x \in \ker((PQ)(u))$, on pose

$$x_1 = (AP)(u)(x) \text{ et } x_2 = (BQ)(u)(x),$$

on a manifestement $x = x_1 + x_2$ et

$$\begin{aligned} Q(u)(x_1) &= (QAP)(u)(x) = (AQP)(u)(x) = A(u)((QP)(u)(x)) \\ &= A(u)(0_E) = 0_E \end{aligned}$$

donc $x_1 \in \ker(Q(u))$. Par le même calcul, $x_2 \in \ker(P(u))$ d'où l'inclusion

$$\ker((PQ)(u)) \subset \ker(P(u)) + \ker(Q(u)).$$

L'autre inclusion est immédiate car nous avons établi que $\ker(P(u))$ et $\ker(Q(u))$ sont inclus dans $\ker((PQ)(u))$ donc leur somme aussi (car ce dernier ensemble est un espace vectoriel). Par conséquent, on a prouvé l'égalité ensembliste

$$\ker((PQ)(u)) = \ker(P(u)) + \ker(Q(u)) = \ker(P(u)) \oplus \ker(Q(u)).$$

Commentaires : Si la question est formellement une question de cours. Néanmoins, la première inclusion va être très discriminante entre les candidats. Seuls vont y parvenir les candidats ayant une maîtrise substantielle des polynômes d'endomorphismes. La seconde question est destinée aux bonnes copies en pratique et seules les meilleures copies répondront correctement à toute la question. Chacun des points (intersection nulle, chaque inclusion) sera fortement valorisée.

Q8. Par construction, on a

$$Q_1 = \frac{\pi_U}{P_1^{k_1}} = \frac{P_1^{k_1} P_2^{k_2}}{P_1^{k_1}} = P_2^{k_2} \text{ et } Q_2 = P_1^{k_1}$$

(par le même calcul). Comme P_1 et P_2 sont premiers entre eux, $P_1^{k_1}$ et $P_2^{k_2}$ le sont aussi (si D est un diviseur irréductible de $P_1^{k_1} = P_1 \times \cdots \times P_1$ alors, par le lemme de Gauss, il divise l'un de ces facteurs donc il divise P_1 , de même si D divise $P_2^{k_2}$, il divise P_2 donc s'il divise $P_1^{k_1}$ et $P_2^{k_2}$, il divise P_1 et P_2 sont il est constant, cqfd). Le théorème de Bezout montre alors l'existence de deux polynômes R_1 et R_2 tels que

$$R_1 Q_1 + R_2 Q_2 = 1.$$

Commentaires : Les questions d'arithmétique (dans les entiers ou les polynômes) s'avèrent toujours difficiles aux candidats à ce concours donc cette question sera très sélective. Le lemme de Gauss doit impérativement être invoqué.

Q9. D'après la relation de Bezout, on a :

$$\begin{aligned} \sum_{i=1}^m p_i &= \sum_{i=1}^m R_i(u) \circ Q_i(u) = \sum_{i=1}^m (R_i Q_i)(u) \\ &= \left(\sum_{i=1}^m R_i Q_i \right)(u) = 1(u) = u^0 = \text{Id}_E. \end{aligned}$$

Par définition, Q_i possède comme facteurs tous les $\Pi_\alpha^{k_\alpha}$ pour $\alpha \neq i$ et Q_j a pour facteur tous les $\Pi_\alpha^{k_\alpha}$ pour $\alpha \neq j$. Par conséquent, le produit $Q_i Q_j$ possède tous les facteurs $\Pi_\alpha^{k_\alpha}$ sans exception c'est-à-dire qu'il se factorise par P . Ainsi, il existe un polynôme $B_{i,j}$ tel que $Q_i Q_j = P Q_{i,j}$ donc on a :

$$\begin{aligned} p_i \circ p_j &= (R_i Q_i R_j Q_j)(u) = (R_i R_j Q_i Q_j)(u) \\ &= (R_i R_j B_{i,j} P)(u) = (R_i R_j B_{i,j}) \circ \underbrace{P(u)}_{=0} = 0_E. \end{aligned}$$

En composant l'égalité $\sum_{i=1}^m p_i = \text{Id}$ par p_j , comme $p_i \circ p_j = 0$ si $i \neq j$, on obtient l'égalité

$$p_j = \sum_{i=1}^m \underbrace{p_i \circ p_j}_{=0 \text{ si } i \neq j} = p_j \circ p_j = (p_j)^2$$

donc p_j est un projecteur.

Commentaires : La première égalité est élémentaire. La seconde et la troisième seront réussies uniquement dans les bonnes copies au minimum (plutôt les meilleures même à ce concours).

Q10. Les polynômes $P_i = (X - \lambda_i)^{\alpha_i}$ sont deux à deux premiers entre eux (car scindés sans racines communes) donc, d'après le lemme des noyaux, on a :

$$\begin{aligned} N_1 \oplus \cdots \oplus N_m &= \ker(P_1(u)) \oplus \cdots \oplus \ker(P_m(u)) \\ &= \ker((P_1 \cdots P_m)(u)) = \ker(\chi_u(u)). \end{aligned}$$

Or, d'après le théorème de Cayley-Hamilton, χ_u annule u donc $\chi_u(u) = 0$ d'où son noyau est E tout entier, ce qui prouve le résultat attendu.

Commentaires : Il s'agit d'une question de cours. Si elle vous a posé des difficultés, n'hésitez pas à la retravailler car elle est régulièrement utilisée à ce concours (dans les écrits).

Q11. Soit $x \in E$. D'après la question 9, on a :

$$p_1(x) + \cdots + p_m(x) = \text{Id}(x) = x$$

et comme chaque $p_i(x)$ appartient à $\text{Im}(p_i)$, nous venons de montrer que

$$E \subset \text{Im}(p_1) + \cdots + \text{Im}(p_m).$$

L'inclusion réciproque est immédiate car chaque $\text{Im}(p_i)$ est un sous-espace vectoriel de E donc leur somme est incluse dans E d'où l'égalité ensembliste :

$$E = \text{Im}(p_1) + \cdots + \text{Im}(p_m).$$

Vérifions que cette somme est directe. Soit $x_1 \in \text{Im}(p_1)$, ..., $x_m \in \text{Im}(p_m)$ tel que $\sum_{i=1}^m x_i = 0_E$.

En composant cette égalité par p_j , on obtient :

$$(*) : \sum_{i=1}^m p_j(x_i) = 0.$$

Or, pour chaque $i \in \{1, \dots, m\}$, il existe $y_i \in E$ tel que

$$x_i = p_i(y_i) \Rightarrow p_j(x_i) = (p_j \circ p_i)(y_i) = 0_0$$

si $i \neq j$ (d'après la question 9). Si $i = j$ alors

$$p_i(x_i) = (p_i \circ p_i)(y_i) = (p_i)^2(y_i) = p_i(y_i) = x_i$$

(car p_i est un projecteur d'après la question 9). Par conséquent, l'égalité $(*)$ montre que $x_i = 0_E$ pour tout i donc la somme est directe, ce qui permet de conclure.

Commentaires : L'égalité $E = \text{Im}(p_1) + \cdots + \text{Im}(p_m)$ est accessible aux candidats solide en algèbre linéaire (c'est-à-dire maîtrisant la notion d'image et de somme d'espaces vectoriels). Pour la somme directe, seuls les très bonnes copies y parviennent.

Q12. Nous allons montrer que $\text{Im}(p_i) \subset N_i$ pour chaque i puis nous utiliserons un argument de dimension pour montrer l'égalité.

Dans le cas de cette question, pour chaque i , on a $P_i = X - \lambda_i$ et $N_i = \ker(P_i^{k_i}(u))$. Par définition des polynômes Q_i , pour chaque i , on a $\pi_u = P_i^{k_i}Q_i$ et par définition de p_i , on a

$$p_i = R_i(u) \circ Q_i(u) = (R_i Q_i)(u)$$

Soit $x \in N_i$, il existe $y \in E$ tel que $x = p_i(y)$ alors

$$\begin{aligned} P_i^{k_i}(u)(x) &= P_i^{k_i}(u)(p_i(y)) = (P_i^{k_i}(u) \circ p_i)(x) \\ &= \left(P_i^{k_i}(u) \circ (R_i Q_i)(u) \right)(x) = \left(P_i^{k_i} R_i Q_i \right)(u)(x) \\ &= \left(R_i P_i^{k_i} Q_i \right)(u)(x) = (R_i \pi_u)(u)(x) = R_i(u) [\pi_u(u)(x)] \end{aligned}$$

D'après le théorème de Cayley-Hamilton, π_u annule u c'est-à-dire $\pi_u(u) = 0_{\mathcal{L}(E)}$ donc $P_i^{k_i}(u)(x) = 0_E$ ce qui prouve l'inclusion

$$(*) : \forall i \in \{1, \dots, m\}, \text{Im}(p_i) \subset N_i.$$

En particulier, pour tout entier i , on a :

$$(**) : \dim(\operatorname{Im}(p_i)) \leq \dim(N_i).$$

D'après les questions précédentes, nous avons les égalités ensemblistes suivantes :

$$E = N_1 \oplus \cdots \oplus N_m = \operatorname{Im}(p_1) \oplus \cdots \oplus \operatorname{Im}(p_m).$$

En passant aux dimensions et en utilisant la formule de Grassmann, on obtient :

$$\sum_{i=1}^m \dim(N_i) = \sum_{i=1}^m \dim(\operatorname{Im}(p_i)) \Leftrightarrow \sum_{i=1}^m (\dim(N_i) - \dim(\operatorname{Im}(p_i))) = 0.$$

Tous les termes de la somme sont positifs (d'après l'inégalité (**)) et leur somme vaut 0 donc chaque terme est nul c'est-à-dire

$$\forall i \in \{1, \dots, m\}, \dim(N_i) - \dim(\operatorname{Im}(p_i)) = 0 \Leftrightarrow \dim(N_i) = \dim(\operatorname{Im}(p_i)).$$

Par conséquent, cette égalité combinée à l'inclusion (*) montre l'égalité attendue :

$$\forall i \in \{1, \dots, m\}, N_i = \operatorname{Im}(p_i).$$

Commentaires : Question difficile pour ce concours et réussie uniquement par les meilleures copies. Tout candidat mentionnant que si l'une des inclusions est vraie ($N_i \subset \operatorname{Im}(p_i)$ ou l'autre) permet d'obtenir l'égalité grâce à la dimension aura sa réponse significativement valorisée.

Q13. D'après le cours, on a : $\pi_u = (X - \lambda_1) \cdots (X - \lambda_m)$ (il est scindé à racines simples car u est diagonalisable et ses seules racines sont les valeurs propres de u).

Commentaires : Il s'agit d'une question de cours mais seuls les candidats solides en réduction fourniront la bonne réponse.

Q14. Comme le dénominateur est scindé à racines simples, il existe des nombres $\lambda_1, \dots, \lambda_m$ tel que :

$$\frac{1}{\pi_u} = \frac{1}{(X - \lambda_1) \cdots (X - \lambda_m)} = \frac{\alpha_1}{X - \lambda_1} + \cdots + \frac{\alpha_m}{X - \lambda_m}$$

En multipliant cette égalité par $X - \lambda_i$ et en simplifiant le membre de gauche, on obtient

$$\alpha_i + \sum_{j \neq i} \frac{\alpha_j (X - \lambda_i)}{X - \lambda_j} = \frac{X - \lambda_i}{\pi_u} = \frac{1}{Q_i}.$$

En faisant tendre X vers λ_i , on obtient $\alpha_i = \frac{1}{Q_i(\lambda_i)}$ donc

$$\frac{1}{\pi_u} = \frac{1}{Q_1(\lambda_1)(X - \lambda_1)} + \cdots + \frac{1}{Q_m(\lambda_m)(X - \lambda_m)}.$$

En multipliant cette égalité par π_u , on obtient l'égalité :

$$\begin{aligned} 1 &= \frac{1}{Q_1(\lambda_1)} \times \frac{\pi_u}{X - \lambda_1} + \cdots + \frac{1}{Q_m(\lambda_m)} \times \frac{\pi_u}{X - \lambda_m} \\ &= \frac{1}{Q_1(\lambda_1)} \times Q_1 + \cdots + \frac{1}{Q_m(\lambda_m)} \times Q_m. \end{aligned}$$

Ainsi, nous avons une relation de Bezout liant les polynômes Q_i et nous pouvons choisir comme polynôme R_i , $R_i = \frac{1}{Q_i(\lambda_i)}$. Par définition des p_i , on a :

$$p_i = (R_i Q_i)(u) = \left(\frac{Q_i}{Q_i(\lambda_i)} \right)(u) = \frac{Q_i(u)}{Q_i(\lambda_i)}.$$

Commentaires : La première question est de niveau standard (et une application directe du cours). Elle ne doit pas vous poser de problème. Si c'est le cas, la retravailler. La seconde est destinée aux très bonnes copies.

Q15. Si on effectue la décomposition en éléments simples de la fraction $\frac{X}{\pi_u}$ (selon le même procédé qu'à la question précédente), on obtient

$$\frac{X}{\pi_u} = \frac{\lambda_1}{Q_1(\lambda_1)(X - \lambda_1)} + \cdots + \frac{\lambda_m}{Q_m(\lambda_m)(X - \lambda_m)}.$$

En multipliant par π_u , on obtient la formule souhaitée (comme à la question précédente)

$$X = \sum_{i=1}^m \frac{\lambda_i Q_i(X)}{Q_i(\lambda_i)}.$$

En "évaluant" cette égalité en u , on obtient :

$$u = \sum_{i=1}^m \lambda_i \frac{Q_i(u)}{Q_i(\lambda_i)} = \sum_{i=1}^m \lambda_i p_i.$$

Commentaires : La première question est de difficulté standard (mais sélective). La seconde est élémentaire ... si le candidat maîtrise la notion de polynômes d'endomorphismes.

Q16.

- (a) Cette matrice est symétrique à coefficients réels donc elle est diagonalisable d'après le théorème spectral. Par calcul direct, on a $A^2 = 4I_4$.

Commentaires : Question élémentaire.

- (b) Comme A n'est pas multiple de I_4 , son polynôme minimal est de degré au moins 2 et comme $X^2 - 4$ annule A , il est de degré 2. Comme ce dernier polynôme est unitaire, on peut affirmer que

$$\pi_A = X^2 - 4 = (X - 2)(X + 2).$$

Si on pose $P_1 = X - 2$ et $P_2 = X + 2$ alors $\pi_A = P_1 P_2$. Avec les notations de la question 14, on pose

$$\begin{aligned} Q_1 &= \frac{\pi_A}{P_1} = P_2, \quad Q_2 = \frac{\pi_A}{P_2} = P_1, \\ \widetilde{Q}_1 &= \frac{Q_1}{Q_1(2)} = \frac{X+2}{4}, \quad \widetilde{Q}_2 = \frac{Q_2}{Q_2(-2)} = \frac{X-2}{-4} = \frac{2-X}{4} \end{aligned}$$

alors on peut affirmer que

$$\begin{aligned}\Pi_1 &= p_1 = \widetilde{Q}_1(A) = \frac{1}{4}(A + 2I_4) = \frac{1}{4} \begin{pmatrix} 3 & 1 & 1 & 1 \\ 1 & 3 & -1 & -1 \\ 1 & -1 & 3 & -1 \\ 1 & -1 & -1 & 3 \end{pmatrix}, \\ \Pi_2 &= p_2 = \widetilde{Q}_1(A) = \frac{1}{4}(2I_4 - A) = \frac{1}{4} \begin{pmatrix} 1 & -1 & -1 & -1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \\ -1 & 1 & 1 & 1 \end{pmatrix}.\end{aligned}$$

et la décomposition spectrale de A est :

$$A = 2\Pi_1 - 2\Pi_2.$$

Commentaires : Question difficile car elle demande au candidat d'avoir intégré les notions fournies par le sujet ainsi que les résultats établis ... qui ne sont pas standards du tout à ce niveau et à ce concours !

- (c) Comme $\Pi_1\Pi_2 = 0 = \Pi_2\Pi_1$ (cf. question 9), les matrices Π_1 et Π_2 commutent. La formule du binôme de Newton nous donne :

$$A^q = 2^q (\Pi_1 - \Pi_2)^q = 2^q \sum_{k=0}^q (-1)^{q-k} \binom{q}{k} \Pi_1^k \Pi_2^{q-k}.$$

Si $k \geq 1$ et $q-k \geq 1 \Leftrightarrow k \leq q-1$ alors le produit matriciel $\Pi_1^k \Pi_2^{q-k}$ contient le facteur $\Pi_1\Pi_2$ qui vaut 0 donc $\Pi_1^k \Pi_2^{q-k} = 0$. Dans la somme, il ne reste que les deux indices extrêmes.

$$\begin{aligned}A^q &= 2^q \left[(-1)^{q-0} \binom{q}{0} \Pi_1^0 \Pi_2^{q-0} + (-1)^{q-q} \binom{q}{q} \Pi_1^q \Pi_2^{q-q} \right] \\ &= 2^q [(-1)^q \Pi_2^q + \Pi_1^q] = 2^q [(-1)^q \Pi_1 + \Pi_2]\end{aligned}$$

(car $\Pi_1^2 = \Pi_1$ donc $\Pi_1^q = \Pi_1$, idem avec Π_2).

Commentaires : Question de niveau standard. La rigueur sera un critère important d'évaluation (parler de la commutativité, distinction des cas selon les exposants, etc).

Q17. Commençons par rappeler que

$$\mathbb{C}[v] = \{P(v), P \in \mathbb{C}[X]\} \text{ et } \mathbb{C}_d[v] = \{P(v), P \in \mathbb{C}_d[X]\}.$$

Soit $w \in \mathbb{C}[v]$, il existe $P \in \mathbb{C}[X]$ tel que $w = P(v)$. La division euclidienne de P par π_v montrer l'existence de deux polynômes Q et R dans $\mathbb{C}[X]$ tel que

$$P = Q\pi_v + R \text{ avec } \deg(R) < \deg(\pi_v).$$

Par conséquent, on a :

$$w = P(v) = Q(v)\pi_v(v) + R(v) = R(v) \in \mathbb{C}_{d-1}[v]$$

(car π_v annule v donc $\pi_v(v) = 0$) où d est le degré de π_v c'est-à-dire que nous avons l'inclusion $\mathbb{C}[v] \subset \mathbb{C}_{d-1}[v]$. L'inclusion réciproque est immédiate d'où l'égalité

$$\begin{aligned}\mathbb{C}[v] &= \mathbb{C}_d[v] = \{a_0 \text{Id} + a_1 v + \dots + a_{d-1} v^{d-1}, (a_0, \dots, a_{d-1}) \in \mathbb{C}^d\} \\ &= \text{Vect}(\text{Id}, v, \dots, v^{d-1}).\end{aligned}$$

Vérifions que la famille $(\text{Id}, v, \dots, v^{d-1})$ est libre. Dans ce cas, on pourra affirmer que

$$\dim(\mathbb{C}[v]) = d = \deg(\pi_v)$$

et nous aurons conclut. Soient $(a_0, \dots, a_{d-1}) \in \mathbb{C}^d$ tel que

$$a_0 \text{Id} + a_1 v + \dots + a_{d-1} v^{d-1} = 0.$$

Ainsi, le polynôme $P = a_0 + a_1 X + \dots + a_{d-1} X^{d-1}$ annule v et il est degré strictement inférieur au polynôme minimal de v donc il est nul c'est-à-dire $a_0 = 0, \dots, a_{d-1} = 0$ ce qui prouve la liberté de la famille et conclut la question.

Commentaires : Il s'agit d'une question de cours ... mais pour les candidats à ce concours, ce n'est probablement pas le cas (car ils ne l'auront pas intégré). Tout point clé de la preuve sera fortement valorisé (liberté, caractère générateur). Pour le caractère générateur, on peut remarquer que v^d est combinaison linéaire de $\text{Id}, v, \dots, v^{d-1}$, composée cette égalité par v pour en déduire que v^{d+1} est aussi combinaison linéaire des $(v^i)_{0 \leq i < d}$ puis itérer. Si cette question vous a posé problème, n'hésitez pas à la retravailler. Les raisonnements utilisés sont standards et accessibles à ce concours.

Q18. D'après la question précédente, $\mathbb{C}[u]$ est de dimension m qui est le cardinal de la famille $(p_1, \dots, p_m)$. Si cette famille appartient à $\mathbb{C}[u]$ et si elle est libre alors elle sera bien une base de cet espace vectoriel.

D'après la question 14, chaque p_i est un polynôme en u donc appartient à $\mathbb{C}[u]$.

Soit $(a_1, \dots, a_m) \in \mathbb{C}^m$ tel que $\sum_{i=1}^m a_i p_i = 0$. En composant cette égalité par p_j , on obtient :

$$\sum_{i=1}^m a_i p_j \circ p_i = 0 \stackrel{(*)}{\Leftrightarrow} a_j (p_j)^2 = 0 \stackrel{(**)}{\Leftrightarrow} a_j p_j = 0 \stackrel{(***)}{\Leftrightarrow} a_j = 0.$$

(*) car $p_j \circ p_i = 0$ si $i \neq j$. (**) p_j est un projecteur. (***) car p_j est non nul car son image est l'espace propre associé à λ_j qui est non nul par définition.

Commentaires : Question destinée aux meilleures copies.

Q19. Supposons que $(p_1, \dots, p_m)$ soit une base de $\mathbb{C}[u]$. Comme $u \in \mathbb{C}[u]$, il existe des nombres $a_1, \dots, a_m$ tel que $u = a_1 p_1 + \dots + a_m p_m$. Si $x \in \text{Im}(p_j)$ alors $p_j(x) = x$ et, pour tout $i \neq j$, on a

$$p_i(x) = p_i(p_j(x)) = (p_i \circ p_j)(x) = 0(x) = 0$$

donc $u(x) = \sum_{i=1}^m a_i p_i(x) = a_j x$. Si on fixe une base $\mathcal{B}_j$ de $\text{Im}(p_j)$ alors $\mathcal{B}_j$ est une base de $\text{Im}(p_j)$ formée de vecteurs propres de u . D'après la question 11, on a

$$E = \text{Im}(p_1) \oplus \dots \oplus \text{Im}(p_m)$$

donc

$$\mathcal{B} = \mathcal{B}_1 \cup \dots \cup \mathcal{B}_m$$

est une base E formée de vecteurs propres pour u . Par conséquent, u est diagonalisable, ce qui est absurde donc $(p_1, \dots, p_m)$ n'est pas une base de $\mathbb{C}[u]$.

Commentaires : Question destinée aux meilleures copies.

Q20. Pour tout polynôme P de $\mathbb{C}[X]$, il existe des nombres $a_0, \dots, a_d$ tels que $P = \sum_{q=0}^d a_q X^q$ donc

$$\begin{aligned} P(u) &= \sum_{q=0}^d a_q u^q = \sum_{q=0}^d a_q \sum_{i=1}^m \lambda_i^q f_i = \sum_{q=0}^d \sum_{i=1}^m a_q \lambda_i^q f_i \\ &= \sum_{i=1}^m \sum_{q=0}^d a_q \lambda_i^q f_i \text{ (Fubini) } = \sum_{i=1}^m f_i \left(\sum_{q=0}^d a_q \lambda_i^q \right) = \sum_{i=1}^m f_i P(\lambda_i). \end{aligned}$$

On choisit le polynôme $P = (X - \lambda_1) \cdots (X - \lambda_m)$ qui est scindé à racines simples et s'annule en chaque λ_i donc

$$P(u) = \sum_{i=1}^m f_i 0 = 0$$

c'est-à-dire que P annule u d'où la diagonalisabilité de u .

Commentaires : Question destinée aux meilleures copies.

ÉPREUVE MUTUALISÉE AVEC E3A-POLYTECH**ÉPREUVE SPÉCIFIQUE - FILIÈRE MP**

INFORMATIQUE**Durée : 4 heures**

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

RAPPEL DES CONSIGNES

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

Les calculatrices sont interdites.

Le sujet est composé de trois parties indépendantes.

Partie I - Programmation en OCaml : sélection du $(k + 1)^{\text{e}}$ plus petit élément

La sélection du $(k + 1)^{\text{e}}$ plus petit élément d'une liste d'entiers L , non nécessairement triée, consiste à trouver le $(k + 1)^{\text{e}}$ élément de la liste obtenue en triant L dans l'ordre croissant.

Par exemple, si $L = [9; 1; 2; 4; 7; 8]$ le 3^e plus petit élément de L est 4. On pourra remarquer que si la liste L est triée dans l'ordre croissant, le $(k + 1)^{\text{e}}$ plus petit élément est l'élément de rang k dans L .

On présente un algorithme permettant de résoudre ce problème de sélection avec une complexité temporelle linéaire dans le pire cas. Celui-ci est basé sur le principe de "diviser pour régner" et sur le choix d'un bon pivot pour partager la liste en deux sous-listes.

Dans cette partie, les fonctions demandées sont à écrire en OCaml et ne doivent faire intervenir aucun trait impératif du langage (références, tableaux ou autres champs mutables ou exception par exemple).

Étant donné un réel a , on note $\lfloor a \rfloor$ le plus grand entier inférieur ou égal à a .

I.1 - Fonctions utiles

Dans cette section, on écrit des fonctions auxiliaires qui sont utiles pour la fonction principale.

Q1. Écrire une fonction récursive de signature :

```
longueur : 'a list -> int
```

et telle que `longueur l` est la longueur de la liste `l`.

Q2. Écrire une fonction récursive de signature :

```
insertion : 'a list -> 'a -> 'a list
```

et telle que `insertion l a` est la liste triée dans l'ordre croissant obtenue en ajoutant l'élément a dans la liste croissante `l`.

Q3. En déduire une fonction récursive de signature :

```
tri_insertion : 'a list -> 'a list
```

et telle que `tri_insertion l` est la liste obtenue en triant `l` dans l'ordre croissant.

Q4. Écrire une fonction récursive de signature :

```
selection_n : 'a list -> int -> 'a
```

et telle que `selection_n l n` est l'élément de rang `n` de la liste `l`.

Par exemple, `selection_n [4;2;6;4;1;15] 3` est égal à 4.

Q5. Écrire une fonction récursive de signature :

`paquets_de_cinq : 'a list -> 'a list list`

et telle que `paquets_de_cinq l` est une liste de listes obtenue en regroupant les éléments de la liste `l` par paquets de cinq sauf éventuellement le dernier paquet qui est non vide et qui contient au plus cinq éléments. Par exemple :

- `paquets_de_cinq []` est égal à `[]`,
- `paquets_de_cinq [2;1;2;1;3]` est égal à `[[2;1;2;1;3]]`,
- `paquets_de_cinq [3;4;2;1;5;6;3]` est égal à `[[3;4;2;1;5]; [6;3]]`.

Q6. Écrire une fonction récursive de signature :

`medians : 'a list list -> 'a list`

et telle que `medians l` est la liste `m` obtenue en prenant dans chaque liste l_k apparaissant dans la liste de listes `l` l'élément médian de l_k . On convient que pour une liste A dont les éléments sont exactement $a_0 \leq a_1 \leq \dots \leq a_{n-1}$, l'élément médian désigne $a_{\lfloor \frac{n}{2} \rfloor}$.

Dans le cas où la liste L n'est pas triée, l'élément médian désigne l'élément médian de la liste obtenue en triant L par ordre croissant. Par exemple :

`medians [[3;1;5;3;2]; [4;3;1]; [1;3]; [5;1;2;4]]` est égal à `[3;3;1;2]`.

Q7. Écrire une fonction de signature :

`partage : 'a -> 'a list -> 'a list * 'a list * int * int`

telle que `partage p l` est un quadruplet l_1, l_2, n_1, n_2 où l_1 est la liste des éléments de `l` plus petit que `p`, l_2 est la liste des éléments de `l` strictement plus grand que `p`, n_1 et n_2 sont respectivement les longueurs de l_1 et l_2 .

I.2 - La fonction de sélection et sa complexité

On détaille la fonction de sélection :

Q8. Écrire une fonction récursive de signature :

`selection : 'a list -> int -> 'a`

telle que `selection l k` est le $(k+1)^{\text{e}}$ plus petit élément de la liste `l`. L'écriture de la fonction sera une traduction en OCaml de l'Algorithme 1 présenté en page 4.

On cherche à déterminer la complexité en nombre de comparaisons de la fonction `selection`. Pour tout $n \in \mathbb{N}$, on note $T(n)$ le nombre maximum de comparaisons entre éléments lors d'une sélection d'un élément quelconque dans des listes L **sans répétition** de taille n .

En analysant l'Algorithme 1, il est possible de démontrer que :

$$\forall n \geq 55, T(n) \leq T\left(\left\lfloor \frac{n+4}{5} \right\rfloor\right) + T\left(\left\lfloor \frac{8n}{11} \right\rfloor\right) + 4n. \quad (I)$$

Q9. En admettant la proposition (I), montrer que pour tout entier n supérieur à 1, on a :

$$T(n) \leq (200 + T(55))n.$$

Pour l'initialisation, on pourra remarquer que T est une fonction croissante.

Algorithme 1 - Sélection du $(k + 1)^{\text{e}}$ plus petit élément

```

1  SELECTION L k :
   /* L est une liste, k est un entier positif */
2  début
3    n ← LONGUEUR L
4    si n ≤ 5 alors
5      M ← (TRI_INSERTION L)
6      retourner l'élément de rang k de M
7    fin
8    sinon
9      L_Cinq ← PAQUETS_DE_CINQ L
10     M ← MEDIANS L_Cinq
11     pivot ← SELECTION M ((n + 4) // 5) // 2
        /* L'opérateur // désigne le quotient d'entiers. Le rang ((n + 4) // 5) // 2
           correspond au rang du médian de la liste M */
12     L1, L2, n1, n2 ← PARTAGE pivot L
13     si k < n1 alors
14       retourner SELECTION L1 k
15     fin
16     sinon
17       retourner SELECTION L2 (k - n1)
18     fin
19   fin
20 fin

```

Partie II - Recherche d'une clique de célébrités

II.1 - Définitions et propriétés

Définition 1 (Graphe). On appelle **graphe** un couple $G = (S, A)$ où S est un ensemble fini appelé ensemble des sommets et A est une partie de $S \times S$, appelée ensemble des arêtes.

On pourra remarquer que, dans cette définition de graphe, les éléments de la forme (s, s) où $s \in S$ sont des arêtes possibles.

Définition 2 (Clique). Soit $G = (S, A)$ un graphe. Soit S' une partie de S . On dit que S' est une **clique** si :

$$\forall (s_1, s_2) \in S' \times S', (s_1, s_2) \in A.$$

Définition 3 (Clique de célébrités, célébrité). Soient $G = (S, A)$ un graphe et C une partie de S . On dit que C est une **clique de célébrités** si C est une clique et :

$$\forall (c, s) \in C \times S, ((s, c) \in A) \wedge ((c, s) \in A \implies s \in C).$$

Un élément de l'ensemble C est alors appelé **célébrité**.

Le terme "célébrité" provient de l'interprétation suivante : l'ensemble des sommets correspond à un ensemble de personnes et une arête (s, c) représente le fait que s connaît c . Ainsi, une célébrité est connue de tous et elle connaît uniquement les autres célébrités.

Q10. Dans cette question, on pose $S = \{0, 1, 2, \dots, 6\}$. Pour chacun des graphes suivants, préciser s'ils contiennent une clique de célébrités non vide. Dans le cas où il y en a une, l'expliquer.

1. $G_1 = (S, A_1)$ avec $A_1 = \{(1, 2), (1, 3), (1, 5), (2, 6)\}$.
2. $G_2 = (S, A_2)$ avec

$$A_2 = \left\{ \begin{array}{l} (0, 3), (0, 5), (1, 2), (1, 3), (2, 2), (2, 3), (3, 3), \\ (4, 1), (4, 3), (4, 5), (5, 1), (5, 3), (6, 1), (6, 3) \end{array} \right\}.$$

Q11. Soit $G = (S, A)$ un graphe quelconque. Montrer que s'il existe une clique de célébrités non vide C dans G , alors celle-ci est unique.

Dans la suite, on note C_G l'unique clique de célébrités non vide du graphe G . Dans le cas où celle-ci n'existe pas, C_G désigne alors l'ensemble vide qui est noté $\emptyset$.

Q12. Soient $G = (S, A)$ un graphe et p un sommet de G . On note $G' = (S \setminus \{p\}, A \cap (S \setminus \{p\} \times S \setminus \{p\}))$. Montrer les propositions suivantes :

- a) Montrer que si $C_{G'}$ est égal à l'ensemble vide, alors $C_G \in \{\emptyset, \{p\}\}$.
- b) Montrer que si $C_G \setminus \{p\} \neq \emptyset$, alors $C_{G'} = C_G \setminus \{p\}$.
- c) On suppose que $C_{G'}$ n'est pas l'ensemble vide et on fixe c' un élément de $C_{G'}$.
 - i) Montrer que si (p, c') n'est pas un élément de A , alors $C_G \in \{\emptyset, \{p\}\}$.
 - ii) Montrer que si (c', p) n'est pas un élément de A , alors $C_G \in \{\emptyset, C_{G'}\}$.
 - iii) Montrer que si (p, c') et (c', p) sont des éléments de A , alors $C_G \in \{\emptyset, \{p\} \cup C_{G'}\}$.

II.2 - Algorithmique et programmation en Python (Informatique Commune)

Dans la suite, l'ensemble des sommets est de la forme $\{0, 1, \dots, n-1\}$ où n est un entier supérieur à 1 et un graphe $G = (S, A)$ est représenté en Python par sa liste d'adjacence que l'on note L_G et qui est définie par :

$$[[j \mid j \in S \text{ et } (i, j) \in A] \mid i \in S].$$

Par exemple, si $G = (\{0, 1, 2, 3\}, \{(0, 1), (3, 2), (3, 1), (1, 2)\})$, alors $L_G = [[1], [2], [], [1, 2]]$.

On pourra remarquer que si l'ensemble des sommets d'un graphe G est égal à $\{0, 1, \dots, n-1\}$, alors la longueur de la liste L_G est égale à n .

Q13. Écrire une fonction Python `est_clique(L, R)` prenant en argument une liste L qui est une liste d'adjacence d'un graphe $G = (S, A)$ et une liste R sans répétition d'éléments de S et qui renvoie `True` si l'ensemble des éléments de R constitue une clique de G et `False` sinon.

Q14. On considère le graphe G ayant comme liste d'adjacence :

$$L_G = [[1, 3, 5], [0, 2], [4, 6], [2, 4, 5, 6], [2], [2, 3, 4], [2, 4, 6]].$$

Décrire l'évolution de la variable C à chaque étape de l'Algorithme 2 décrit en page 6.

Q15. Écrire une fonction Python `Clique_possible_C(G)` prenant en argument une liste G représentant un graphe et qui renvoie la liste C construite à l'aide de l'Algorithme 2.

Q16. Montrer par récurrence sur le nombre de sommets que si G est un graphe où C_G est non vide, alors `Clique_possible_C(G)` est égale à C_G .

Algorithme 2 - Construction d'une clique de célébrités possibles

```

1 CLIQUE_POSSIBLE_C G :
2 début
3   C ← []
4   S ← [0, 1, ..., n - 1]
5   /* n est le nombre de sommet de G */
6   pour chaque s élément de S faire
7     si C est vide alors
8       Ajouter s dans C
9     fin
10    sinon
11      c ← premier élément de C
12      t ← FAUX
13      /* t permet de vérifier si on a effectué certaines instructions */
14      si (s,c) n'est pas une arête de G alors
15        C ← [s]
16        t ← VRAI
17      fin
18      si (c,s) n'est pas une arête de G alors
19        C ← C
20        t ← VRAI
21      fin
22      si t = FAUX alors
23        Ajouter s à la fin de liste C
24      fin
25 fin
26 retourner C
  
```

Partie III - Étude d'une famille d'automates

Dans cette partie, l'alphabet Σ désigne l'ensemble $\{0, 1\}$, le symbole ε désigne le mot vide et on rappelle que Σ^* désigne l'ensemble des mots sur l'alphabet Σ .

Étant donné un mot w , on rappelle que $|w|$ désigne la longueur du mot w et l'indexation des lettres de w commence par 0. La première lettre de w est donc w_0 .

La notation $\text{Card}(E)$ désigne le cardinal d'un ensemble E .

Étant donnés un entier n et un entier non nul m , la notation $n \bmod m$ désigne le reste de la division euclidienne de n par m .

III.1 - Définitions

Définition 4 (Automate déterministe). Un **Automate déterministe** est un quintuplet $A = (Q, \Sigma, \delta, q_0, F)$ avec :

- Q un ensemble fini non vide appelé ensemble des états,
- Σ est un ensemble fini appelé alphabet,
- $\delta : Q \times \Sigma \rightarrow Q$ une application appelée application de transition,
- q_0 un élément de Q appelé état initial,
- F une partie de Q appelée ensemble des états finaux.

Définition 5 (Application de transition étendue aux mots). Soit $A = (Q, \Sigma, \delta, q_0, F)$ un automate déterministe.

On définit de manière récursive $\delta^* : Q \times \Sigma^* \rightarrow Q$ par :

$$\begin{aligned} \forall q \in Q, \quad \delta^*(q, \varepsilon) &= q, \\ \forall q \in Q, \forall a \in \Sigma, \forall w \in \Sigma^*, \quad \delta^*(q, aw) &= \delta^*(\delta(q, a), w). \end{aligned}$$

Définition 6 (Reconnaissance d'un mot par un automate). Soient $w = w_0 w_1 \dots w_n$ un mot sur un alphabet Σ et $A = (Q, \Sigma, \delta, q_0, F)$. On dit que w est **reconnu** par l'automate A si $\delta^*(q_0, w) \in F$.

Définition 7 (Automate $A_{k,p}$, fonction indicatrice $L_{k,p}$). Soient p et k deux entiers vérifiant $0 \leq k \leq p-1$. L'automate $A_{k,p}$ est défini par :

- $Q = \{0, 1, \dots, p-1\} \times \{0, 1\}$,
- $\Sigma = \{0, 1\}$,
- $\forall (c, e) \in Q, \delta((c, e), 0) = ((c+1) \bmod p, e)$,
- $\forall (c, e) \in Q, \delta((c, e), 1) = \begin{cases} ((c+1) \bmod p, 1-e) & \text{si } c = k \bmod p, \\ ((c+1) \bmod p, e) & \text{sinon.} \end{cases}$
- $q_0 = (0, 0)$,
- $F = \{0, 1, \dots, p-1\} \times \{1\}$.

On note $L_{k,p}$ la fonction indicatrice de l'ensemble des mots reconnus par $A_{k,p}$. Soit autrement :

$$\forall u \in \Sigma^*, L_{k,p}(u) = \begin{cases} 1 & \text{si } A_{k,p} \text{ reconnaît } u \\ 0 & \text{sinon.} \end{cases}$$

III.2 - Exemples et propriétés élémentaires des $A_{k,p}$

Q17. Soit $w \in \Sigma^*$. Expliciter sans démonstration l'état $\delta^*(q_0, w)$, la lecture du mot étant effectuée dans l'automate $A_{1,3}$. On pourra s'aider d'une représentation graphique de l'automate.

Dans les questions **Q18 à Q22**, p et k désignent des entiers tels que $p > 2$ et $k \in \{0, 1, \dots, p-1\}$.

Q18. Soit $w \in \Sigma^*$. Expliciter l'état $\delta^*(q_0, w)$, la lecture du mot étant effectuée dans l'automate $A_{k,p}$.
On ne demande pas de démonstration.

Un corollaire direct du résultat de la question **Q18** est que l'ensemble des mots reconnu par l'automate $A_{k,p}$ est égal à :

$$\{w \in \Sigma^* \mid \text{Card}(\{m \in \mathbb{N} \mid pm + k \leq |w| - 1 \text{ et } w_{pm+k} = 1\}) \text{ est impair}\}.$$

Dans la suite du problème, on admet ce résultat.

Q19. Soit w un mot reconnu par un automate $A_{k,p}$. Montrer que w est reconnu par au moins un autre automate parmi $A_{0,2}, A_{1,2}, A_{l,p}$ avec $l \neq k$.

Définition 8 (Ou exclusif étendu aux mots binaires). On rappelle que le **Ou exclusif** qu'on note $\oplus$ est une opération définie sur $\{0, 1\}$ par :

$$0 \oplus 0 = 1 \oplus 1 = 0 \text{ et } 0 \oplus 1 = 1 \oplus 0 = 1.$$

Soient $n \in \mathbb{N}$, u et v deux éléments de $\{0, 1\}^n$. On définit le **Ou exclusif** de u et v , noté $u \oplus v$, le mot de longueur n défini par :

$$\forall i \in \{0, 1, \dots, n-1\}, (u \oplus v)_i = u_i \oplus v_i.$$

Q20. Soient u et v deux mots de Σ^* de même longueur. Montrer que :

$$L_{k,p}(u \oplus v) = L_{k,p}(u) \oplus L_{k,p}(v).$$

Q21. Soit w un mot binaire vérifiant :

$$L_{0,2}(w) = L_{1,2}(w) = 0 \text{ et } \forall k \in \{1, 2, \dots, p-1\}, L_{k,p}(w) = 0.$$

- Montrer que $L_{0,p}(w) = 0$.
- En déduire que pour tout mot $w' \in 0^* \cdot w$, on a :

$$L_{0,2}(w') = L_{1,2}(w') = 0 \text{ et } \forall k \in \{1, 2, \dots, p-1\}, L_{k,p}(w') = 0.$$

Q22. Montrer que pour tout $w \in \Sigma^*$ et $w' \in w \cdot 0^*$, on a $L_{k,p}(w) = L_{k,p}(w')$.

Remarque. Ces égalités permettent la construction d'une relation d'équivalence sur les mots qui est utilisée pour montrer que deux mots de longueur N peuvent être séparés par un automate de la forme $A_{k,p}$ ayant $O(\sqrt{N} \ln(N))$ états.

FIN

SOLUTION

Q1.

```
let rec longueur l = match l with
| [] -> 0
| a::q -> 1 + longueur q;;
```

Commentaires : Question élémentaire de cours.

Q2.

```
let rec insertion l a = match l with
| [] -> [a]
| x::q when x > a -> a::l
| x::q -> x::(insertion q a);;
```

Commentaires : Même remarque qu'à la question précédente.

Q3.

```
let rec tri_insertion l = match l with
| [] -> []
| a::q -> insertion (tri_insertion q) a;;
```

Commentaires : Même remarque qu'à la question précédente. Conclusion : le cours est à connaître (dont les programmes) même en option informatique :-). Que cela soit pour CCINP mais aussi pour CentraleSupElec et Mines-Ponts.

Q4.

```
let rec selection_n l n = match l, n with
| a::q, n when n==0 -> a
| a::q, n -> selection_n q (n-1)
| _, _ -> failwith "élément inexistant";;
```

Commentaires : Question sans difficulté particulière.

Q5.

```
let rec paquets_de_cinq l = match l with
| [] -> []
| a::b::c::d::e::q -> [a;b;c;d;e]::(paquets_de_cinq q)
| l -> [l];;
```

Commentaires : La difficulté de la question consiste à bien traiter tous les sous-cas (oui, je sais, c'est fastidieux) dont le cas des listes non vides à moins de 4 éléments (il faut renvoyer non pas la liste `l` mais la liste de listes `[l]`!).

Q6.

```
let rec medians l =
  match l with
  | [] -> [] ;
  | a::q -> (selection_n (tri_insertion a) ((longueur a)/2))::(
    medians suite);;
```

Commentaires : Question de difficulté standard.

Q7.

```
let partage p l =
  let rec aux l1 l2 n1 n2 acc = match acc with
    | [] -> l1,l2,n1,n2
    | a::q when a < p -> aux (a::l1) l2 (n1+1) n2 q
    | a::q -> aux l1 (a::l2) n1 (n2+1) q
  in aux [] [] 0 0 l;;
```

Commentaires : Il s'agit presque d'une question de cours (partition d'une liste dans le cas du tri rapide). Il faut être attentif aux différents détails.

Q8.

```
let rec selection l k =
  let n = longueur l in
  if n <= 5 then selection_n (tri_insertion l) k
  else
    let l_cinq = paquets_de_cinq l in
    let m = medians l_cinq in
    let pivot = selection m (((n+4)/5)/2) in
    let l1,l2,n1,n2 = partage pivot l in
    if k < n1 then selection l1 k
    else selection l2 (k-n1) ;;
```

Commentaires : Question de difficulté standard pour ce concours.

Q9. On procède par récurrence forte sur n . Si $n \in \{1, \dots, 55\}$, comme la suite $(T(n))_n$ est manifestement croissante (comparer n nombres demande moins de comparaison de $n+1$ nombres), on a

$$T(n) \leq T(55) \leq (200 + T(55)) \leq (200 + T(55))n$$

ce qui prouve l'initialisation $n \in \{1, \dots, 55\}$. **Hérédité.** Soit $n \geq 55$. Supposons la propriété vraie pour tout entier $k < n$ alors

$$T(n) \leq T\left(\left\lfloor \frac{n+4}{5} \right\rfloor\right) + T\left(\left\lfloor \frac{8n}{11} \right\rfloor\right) + 4n.$$

Comme

$$\left\lfloor \frac{n+4}{5} \right\rfloor \leq \frac{n+4}{5} < n \text{ car } 5n - (n+4) = 4(n-1) > 0$$

et $\left\lfloor \frac{8n}{11} \right\rfloor \leq \frac{8n}{11} < n$ car $\frac{8}{11} < 1$ et $n > 0$. Par croissance de la suite T et par hypothèse de récurrence, on obtient :

$$\begin{aligned} T(n) &\leq T\left(\frac{n+4}{5}\right) + T\left(\frac{8n}{11}\right) + 4n \\ &\leq (200 + T(55))\left(\frac{n+4}{5} + \frac{8n}{11}\right) + 4n \\ &\leq (200 + T(55))\left(\frac{11(n+4)}{5} + \frac{5 \times 8n}{11}\right) + 4n \\ &= (200 + T(55))\frac{51n+44}{55} + 4n. \end{aligned}$$

Il reste à montrer la majoration suivante

$$\begin{aligned} (200 + T(55))\frac{51n+44}{55} + 4n &\leq (200 + T(55))n \\ \Leftrightarrow (200 + T(55))51n + 44 + 4 \times 55n &\leq (200 + T(55))55n \\ \Leftrightarrow 44 + 220n &\leq 4n(200 + T(55)) \\ \Leftrightarrow 44 &\leq (800 - 220 + T(55))n \\ \Leftrightarrow 44 &\leq (580 + T(55))n. \end{aligned}$$

Comme

$$(580 + T(55))n \geq (580 + T(55)) \geq 580 \geq 44,$$

on obtient l'inégalité attendue donc l'hérédité est prouvée ainsi que la récurrence.

Commentaires : Question difficile car non ordinaire et utilisant des subtilités mathématiques auxquels ne penseront pas une large majorité des candidats à ce concours. En pratique, cette question est réussie uniquement par les meilleures copies.

Q10. G_1 ne contient pas de clique de célébrités non vide (aucun élément n'est connu de tous). G_2 admet $\{3\}$ comme clique de célébrités non vide (3 est connu de tous et c'est le seul).

Commentaires : Question permettant au candidat à se familiariser avec la notion introduite par le sujet. Être toujours attentif à ce type de questions car elles sont bien utiles pour s'appropriier ces notions, tant pour la programmation que pour les questions théoriques.

Q11. Soient C_1 et C_2 deux cliques de célébrités non vide. Si $c_1 \in C_1$ alors c_1 est connu de tous. Comme C_2 est un non vide, il contient au moins un élément c_2 . Cet élément c_2 connaît nécessairement c_1 et comme c_2 est ne connaît que des célébrités (de C_2 vu la définition des cliques de célébrité), on en déduit que c_1 appartient à C_2 donc $C_1 \subset C_2$. En échangeant les rôles de C_1 et C_2 , on obtient l'inclusion réciproque d'où l'égalité attendue.

Commentaires : Question sélective non par sa difficulté intrinsèque mais par le fait que le candidat doit seul s'approprier la notion en s'aidant de l'interprétation en français via les célébrités. Il est attendu la double implication (en ne faisant pas semblant ou en ne comprenant pas que le raisonnement se fait par implication et non équivalence!).

Q12.

- (a) On suppose que $C_{G'} = \emptyset$. Supposons que C_G est non vide alors il possède un élément c qui est connu de tous les éléments de G . Si on pose $C' = C_G \setminus \{p\}$ alors C' est une clique (tous les éléments de C_G étaient voisins donc ceux de C' aussi) de célébrités (ils sont tous connus de tous et ne connaissent que des célébrités de C_G qui sont différents de p donc appartiennent à C') non vide. D'après la question précédente, on peut affirmer que $C' = C_{G'} = \emptyset$ donc C_G est vide (absurde car C_G est présumé non vide) ou réduit à $\{p\}$. Par conséquent, C_G est vide ou réduit à $\{p\}$.
- (b) Si on pose $C' = C_G \setminus \{p\}$, comme $C_{G'}$ est non vide, d'après la question précédente, C' est une clique de célébrités non vide donc elle est égale à $C_{G'}$ (cf. question 11).
- (c)
- Si $C_G \setminus \{p\}$ est non vide, d'après la question 12.b), on a $C_G \setminus \{p\} = C_{G'}$. Comme $c' \in C_{G'} = C_G \setminus \{p\} \subset C_G$, c' est une célébrité de G donc il est connu de tous, en particulier de p . Ainsi l'arête (p, c') existe, ce qui est absurde donc $C_G \setminus \{p\}$ est vide d'où C_G est vide ou vaut $\{p\}$.
 - Comme (c', p) n'est pas une arête, p n'est pas une célébrité donc $C_G = C_G \setminus \{p\}$. Si $C_G \setminus \{p\}$ est non vide, d'après la question 12.b), on peut affirmer que $C_{G'} = C_G \setminus \{p\} = C_G$. Si $C_G \setminus \{p\}$ est vide alors $C_G = C_G \setminus \{p\}$ est aussi vide.
 - Supposons C_G non vide. Si p n'appartient pas à C_G alors $C_G \setminus \{p\} = C_G$ est non vide donc, d'après la question 12.b), $C_{G'} = C_G \setminus \{p\} = C_G$. Comme (c, p) est une arête, la célébrité de G connaît p donc p est une célébrité de G i.e. $p \in C_G$, ce qui est absurde. Ainsi, p appartient à C_G et comme $(p, c') \in A$, on peut affirmer que $c' \in C_G$ donc $C_G \setminus \{p\}$ est non vide. D'après la question 12.b), on peut affirmer que $C_{G'} = C_G \setminus \{p\}$ donc $C_G = \{p\} \cup C_{G'}$.

Commentaires : Ces différentes questions sont sélectives car elles demandent des preuves fondées sur des définitions et non sur de pseudo évidences déclarées par le candidat. En outre, elle nécessite une bonne maîtrise par le candidat des techniques de preuve (disjonction de cas, preuve par l'absurde, par contraposée). Au final, elles sont convenables traitées uniquement dans les bonnes copies (pas forcément excellentes).

Q13.

```
def est_clique(L,R):
    for x in R :
        for y in R :
            if x not in L[y] :
                return False
    return True
```

Commentaires : Question sans difficulté ... si le candidat a bien compris ce qu'est une clique (cela ne doit pas poser de problème particulier aux optionnaires info).

Q14. On a consécutivement : $[]$, $[0]$, $[0,1]$, $[2]$, $[2]$, $[2,4]$, $[2,4]$, $[2,4,6]$

Commentaires : Question sans difficulté particulière permettant de tester si le candidat est apte à faire tourner un algorithme sur un exemple simple.

Q15.

```

def Clique_possible_C(G):
    C = []
    S = [i for i in range(len(G))]
    for s in S :
        if C == [] :
            C.append(s)
        else :
            c = C[0]
            t = False
            if c not in G[s] :
                C = [s]
                t = True
            if s not in G[c] :
                t = True
            if t == False :
                C.append(s)
    return C

```

Commentaires : Question sans aucune difficulté. Elle est assez bien réussie en général.

Q16. Montrons par récurrence sur n la propriété $(\mathcal{H}_n)$: Si G admet n éléments et C_G est non vide $Clique\_Possible\_C(G)$ renvoie C_G ".

Si $n = 1$, c'est immédiat. Supposons la propriété vérifiée pour un entier $n \geq 1$. Considérons un graphe G à $n + 1$ éléments que l'on présume numéroté $\{0, \dots, n\}$ avec C_G non vide. L'exécution jusqu'à la fin de la boucle *for* pour $s = n - 1$ correspond au code pour le graphe $G' = G \setminus \{n\}$ qui possède $n - 1$ éléments. Par hypothèse de récurrence, $Clique\_Possible\_C(G')$ renvoie $C = C_{G'}$.

Premier cas. S'il est vide, alors le code correspondant à l'exécution de la boucle pour $s = n$ renvoie nécessairement $\{n\}$. Or, d'après la question 12.a), comme $C_{G'}$ est vide, C_G est vide (absurde par hypothèse de $(\mathcal{H}_n)$) donc $C_G = \{n\}$ donc la propriété $(\mathcal{H}_n)$ est vérifiée.

Deuxième cas. $C_{G'}$ est non vide. Lors de l'exécution de la boucle pour $s = n$, la condition *Si C est vide* n'est pas vérifiée (car $C_{G'} \neq \emptyset$). On note c le premier élément de C .

Premier sous-cas. Si l'arête $(s = n, c)$ n'existe pas alors $C = \{n\}$ est renvoyé par la fonction. D'après la question 12.c)i), on peut affirmer que C_G est vide (absurde ici par l'hypothèse $(\mathcal{H}_n)$) ou vaut $\{n\}$ donc la propriété $(\mathcal{H}_n)$ est vérifiée.

Deuxième sous-cas. L'arête $(s = n, c)$ existe. Si l'arête $(c, s = n)$ n'existe pas, $C = C_{G'}$ est renvoyé par le programme. D'après la question 12.c)ii), on peut affirmer que C_G est vide (absurde ici par l'hypothèse $(\mathcal{H}_n)$) ou vaut $C_{G'}$ donc la propriété $(\mathcal{H}_n)$ est vérifiée.

Troisième sous-cas. Les arêtes $(s = n, c)$ et $(c, s = n)$ existent, t reste *FAUX* donc $C = C_{G'} \cup \{n\}$ est renvoyé par le programme. D'après la question 12.c)iii), on peut affirmer que C_G est vide (absurde ici par l'hypothèse $(\mathcal{H}_n)$) ou vaut $C_{G'} \cup \{n\}$ donc la propriété $(\mathcal{H}_n)$ est vérifiée.

Par conséquent, l'hérédité est vérifiée ce qui achève la récurrence.

Commentaires : Question difficile car le candidat doit savoir interpréter soit le code proposé soit l'algorithme proposé, s'appuyer précisément sur lui pour faire une preuve étayée et fonder sur le code ou l'algorithme et non pas sur le souhait ce qu'il devrait faire. N'hésitez pas à vous entraîner sur cet exemple et d'autres pour vous familiariser avec les preuves en informatique. Elles sont toujours sélectives ET rémunératrices (en restant de difficulté raisonnable).

Q17. Pour $\delta^*((0,0), w)$, l'abscisse est augmenté de 1 mod (3) à chaque lettre lue donc elle vaut la longueur $|w|$ du mot modulo 3. Si on note r le nombre de lettres d'indice i qui valent 1 modulo 3 à chaque lecture de l'une de ces lettres, on alterne entre 1 et 0 (passage de 0 à $1 - 0 = 1$ puis à $1 - 1 = 0$) donc tous les deux fois, on aboutit à 0. Ainsi, si ce nombre r est pair, l'ordonnée vaut 0 et 1 sinon donc

$$\delta^*((0,0)) = (|w| \bmod (3), 0) \text{ si } r \text{ est pair et } (|w| \bmod (3), 1) \text{ sinon.}$$

Commentaires : Question sélective car elle demande au candidat d'avoir un peu de recul sur les observations qu'il fait en lisant différents mots (s'il songe à le faire ...).

Q18. Pour $\delta^*((0,0), w)$, l'abscisse est augmenté de 1 mod (p) à chaque lettre lue donc elle vaut la longueur $|w|$ du mot modulo p . Si on note r le nombre de lettres d'indice i qui valent 1 modulo p , à chaque lecture de l'une de ces lettres, on alterne entre 1 et 0 (passage de 0 à $1 - 0 = 1$ puis à $1 - 1 = 0$) donc tous les deux fois, on aboutit à 0. Ainsi, si ce nombre r est pair, l'ordonnée vaut 0 et 1 sinon donc

$$\delta^*((0,0)) = (|w| \bmod (p), 0) \text{ si } r \text{ est pair et } (|w| \bmod (p), 1) \text{ sinon.}$$

Commentaires : Même remarque que précédemment.

Q19. Considérons un mot non reconnu par $A_{0,2}$ et $A_{1,2}$. Si r le nombre de lettres d'indice i qui valent 0 modulo 2 (resp. 1 mod 2) i.e. i pair (resp. i impair) alors $\delta^*((0,0), w)$ n'est pas un état final donc, d'après la question précédente, ce nombre vérifie $r = 0 \bmod (2)$ donc r est pair que l'indice i soit pair ou impair. Par conséquent, le nombre de 1 dans w est un nombre pair. En partitionnant les indices i selon leurs valeurs modulo p , si on note r_ℓ le nombre d'indice i valant ℓ modulo p avec $w_i = 1$ (nombre de 1 en position $\ell \bmod (p)$ en clair) alors $\sum_{\ell=0}^p r_\ell$ est le nombre de 1 donc il est pair. Or, r_k est impair car w est reconnu par $A_{k,p}$ (les mots pour lequel r_k est impair sont exactement les mots reconnus par $A_{k;p}$ la phrase précédant cette question). Comme la somme de $r_0, \dots, r_p$ est pair, au moins un autre r_ℓ , avec $\ell \neq k$, est impair donc w est reconnu par $A_{\ell,k}$.

Commentaires : Question difficile que seules les meilleures copies ont traité correctement.

Q20. On note :

- r_k le nombre de 1 de u en position $i = k \bmod (p)$ alors $r_k = \sum_{i \text{ tel que } i=k \bmod (p)} u_k$;
- s_k le nombre de 1 de v en position $i = k \bmod (p)$ alors $s_k = \sum_{i \text{ tel que } i=k \bmod (p)} v_k$;
- t_k le nombre de 1 de $u \oplus v$ en position $i = k \bmod (p)$

Pour tout $i = k \bmod (p)$, on a $(u \oplus v)_i = 1$ si et seulement si seul l'un des u_i ou l'un des v_i vaut 1. Chaque indice $i = k \bmod (p)$ où se trouve un 1 à la fois dans u et v alors $(u \oplus v)_i = 0$. Si on note z_k le nombre de ces indices alors les 1 apparaissant dans $u \oplus v$ correspondent aux 1 apparaissant dans u sans ceux qui sont communs à v plus ceux dans v auxquels ont enlevé ceux qui sont communs à u c'est-à-dire

$$t_k = (r_k - z_k) + (s_k - z_k) = r_k + s_k - 2z_k$$

Ainsi, t_k a la même parité de $r_k + s_k$. Ainsi, t_k est impair si et seulement si seul l'un des r_k ou s_k est impair c'est-à-dire $u \oplus v$ est reconnu par $A_{k,p}$ si et seulement si seul l'un des mots u ou v est reconnu par $A_{k,p}$ donc

$$L_{k,p}(u \oplus v) = L_{k,p}(u) \oplus L_{k,p}(v).$$

Commentaires : Même remarque que précédemment.

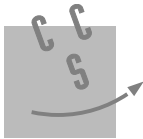
Q21.

- (a) Par hypothèse, w n'est pas reconnu par $A_{0,2}$, ni par $A_{1,2}$ ni $A_{k,p}$ ($1 \leq k \leq p-1$), il n'est pas reconnu par $A_{0,p}$ (par contraposée de la question 19) d'où le résultat.
- (b) w n'est reconnu par aucun des $A_{k,p}$ ($0 \leq k \leq p-1$) ni $A_{0,2}$ ni $A_{1,2}$ donc son nombre de 1 (en toute position $k \bmod (p)$, etc) est pair. En ajoutant autant de 0 à w au début, on ne fait que décaler tous les indices de la même façon, ce qui ne change pas le nombre de 1 en position $k \bmod (p)$ (si on ajoute r zéros, les positions $i = k \bmod (p)$ deviennent $i = k + r \bmod (p)$ donc les 1 accouplés avant le décalage le reste après). Ainsi, w' n'est reconnu par aucun de ces automates, d'où la réponse.

Commentaires : Même remarque que précédemment.

Q22. Cela résulte immédiatement que la question 21.b) (le nombre de 1 en position $k \bmod (p)$ est inchangé).

Commentaires : Même remarque que précédemment.



CONCOURS CENTRALE-SUPÉLEC

Mathématiques 1

MP, MPI

2023

4 heures

Calculatrice autorisée

Sur le calcul ombral

Objectifs

Ce problème introduit le calcul ombral et propose d'en démontrer certains résultats.

Historiquement, ce « calcul » reposait sur un ensemble de manipulations heuristiques sur les indices qui étaient traités comme des puissances. Pour justifier ces règles, une solution consiste à utiliser des endomorphismes agissant sur des polynômes. Ce problème a pour objectif de présenter ces règles et d'en déduire des identités polynomiales non triviales.

Notations

- $\mathbb{K}$ désigne $\mathbb{R}$ ou $\mathbb{C}$.
- $\mathbb{K}[X]$ désigne l'ensemble des polynômes à coefficients dans $\mathbb{K}$. Dans ce problème, on identifie polynômes formels et fonctions polynomiales de $\mathbb{K}$ dans $\mathbb{K}$ associées. On identifie de plus les éléments de $\mathbb{K}$ aux polynômes constants.
- Tout polynôme $p \in \mathbb{K}[X]$ s'écrit de manière unique

$$p = \sum_{k=0}^{+\infty} a_k X^k$$

où (a_k) est une suite à valeurs dans $\mathbb{K}$ nulle à partir d'un certain rang. Si p n'est pas le polynôme nul, son degré $\deg(p)$ est le plus grand entier k tel que $a_k \neq 0$. Par convention, le degré du polynôme nul est -1 (cette convention est inhabituelle).

- Si n est un entier naturel, $\mathbb{K}_n[X]$ désigne le sous-espace vectoriel de $\mathbb{K}[X]$ des polynômes de degré inférieur ou égal à n .
- On note $\mathcal{L}(\mathbb{K}[X])$ l'algèbre des endomorphismes de l'espace vectoriel $\mathbb{K}[X]$.
- On note I l'endomorphisme identité de $\mathbb{K}[X]$.
- Les éléments inversibles de l'algèbre $\mathcal{L}(\mathbb{K}[X])$ sont les endomorphismes bijectifs (automorphismes) de l'espace vectoriel $\mathbb{K}[X]$.
- Pour $T \in \mathcal{L}(\mathbb{K}[X])$ et $p \in \mathbb{K}[X]$, on note $Tp = T(p)$.
- D désigne l'endomorphisme de dérivation sur $\mathbb{K}[X]$: $\forall p \in \mathbb{K}[X], D(p) = Dp = p'$.
- Si T est un endomorphisme de $\mathbb{K}[X]$, on définit la suite d'endomorphismes (T^k) par récurrence : $T^0 = I$ et, pour tout $k \in \mathbb{N}$, $T^{k+1} = T \circ T^k = T^k \circ T$.

I Étude d'endomorphismes de $\mathbb{K}[X]$

I.A — Soit $a \in \mathbb{K}$. Pour tout $p \in \mathbb{K}[X]$, on pose $E_a(p) = E_a p = p(X + a)$.

Q 1. Montrer que E_a est un automorphisme de $\mathbb{K}[X]$.

I.B — À tout $p \in \mathbb{K}[X]$, on associe la fonction $J(p) = Jp$ de $\mathbb{R}$ dans $\mathbb{R}$ définie par

$$\forall x \in \mathbb{R}, \quad J(p)(x) = Jp(x) = \int_x^{x+1} p(t) dt.$$

Q 2. Montrer que J est un endomorphisme de $\mathbb{R}[X]$.

Q 3. Montrer que J conserve le degré et que J est inversible.

I.C — À tout $p \in \mathbb{K}[X]$, on associe la fonction $L(p) = Lp$ de $\mathbb{K}$ dans $\mathbb{K}$ définie par

$$\forall x \in \mathbb{K}, \quad L(p)(x) = Lp(x) = - \int_0^{+\infty} e^{-t} p'(x+t) dt.$$

Q 4. Montrer que $\int_0^{+\infty} e^{-t} t^k dt$ existe pour tout $k \in \mathbb{N}$ et calculer sa valeur.

Q 5. Montrer que L est un endomorphisme de $\mathbb{K}[X]$. Est-il inversible ?

II Formule de Taylor pour les endomorphismes shift-invariants de $\mathbb{K}[X]$

Soit T un endomorphisme de $\mathbb{K}[X]$. On dit que :

- T est *shift-invariant* si, pour tout $a \in \mathbb{K}$, $E_a \circ T = T \circ E_a$;
- T est un *endomorphisme delta* si T est shift-invariant et si l'image du polynôme X par T est une constante non nulle : $TX \in \mathbb{K}^*$.

II.A –

Q 6. Soit $a \in \mathbb{K}$. Vérifier que les endomorphismes I et D sont shift-invariants, ainsi que les endomorphismes E_a , J et L définis dans la partie I. Sont-ils des endomorphismes delta ?

Q 7. Montrer que l'ensemble des endomorphismes shift-invariants de $\mathbb{K}[X]$ est une sous-algèbre de $\mathcal{L}(\mathbb{K}[X])$. L'ensemble des endomorphismes delta de $\mathbb{K}[X]$ est-il stable par addition ? par composition ?

II.B –

Q 8. Soit $(a_k)_{k \in \mathbb{N}}$ une suite d'éléments de $\mathbb{K}$. Pour tout polynôme $p \in \mathbb{K}[X]$, montrer que l'expression

$$\sum_{k=0}^{+\infty} a_k D^k p$$

a un sens et définit un polynôme de $\mathbb{K}[X]$.

On note alors $\sum_{k=0}^{+\infty} a_k D^k$ l'application de $\mathbb{K}[X]$ qui, à un polynôme $p \in \mathbb{K}[X]$, associe le polynôme $\sum_{k=0}^{+\infty} a_k D^k p$.

Q 9. Montrer que, pour toute suite $(a_k)_{k \in \mathbb{N}}$ d'éléments de $\mathbb{K}$, $\sum_{k=0}^{+\infty} a_k D^k$ est un endomorphisme shift-invariant.

Q 10. Soit $(a_k)_{k \in \mathbb{N}}$ et $(b_k)_{k \in \mathbb{N}}$ des suites d'éléments de $\mathbb{K}$ telles que $\sum_{k=0}^{+\infty} a_k D^k = \sum_{k=0}^{+\infty} b_k D^k$.

Montrer que, pour tout $k \in \mathbb{N}$, $a_k = b_k$.

Pour tout $n \in \mathbb{N}$, on définit le polynôme $q_n = \frac{X^n}{n!}$. On se donne T un endomorphisme de $\mathbb{K}[X]$.

Q 11. Montrer que T est un endomorphisme shift-invariant si, et seulement si,

$$T = \sum_{k=0}^{+\infty} (Tq_k)(0) D^k.$$

Q 12. Montrer que deux endomorphismes shift-invariants de $\mathbb{K}[X]$ commutent.

II.C – Dans cette sous-partie, on applique le résultat de la question 11 aux endomorphismes de la partie I.

Q 13. Pour tout $p \in \mathbb{K}[X]$ non nul et $a \in \mathbb{K}$, montrer, à l'aide de la question 11, que

$$p(X+a) = \sum_{k=0}^{\deg(p)} \frac{a^k}{k!} p^{(k)}$$

où $p^{(k)}$ désigne la dérivée k -ième du polynôme p . Reconnaître cette formule.

Q 14. Pour $p \in \mathbb{K}[X]$, exprimer Jp en fonction des dérivées $p^{(k)}$ ($k \in \mathbb{N}$) de p .

Q 15. Démontrer que l'endomorphisme $D - I$ est inversible et exprimer L en fonction de $(D - I)^{-1}$.

II.D – Dans cette sous-partie, T est un endomorphisme non nul shift-invariant de $\mathbb{K}[X]$.

On rappelle que le degré du polynôme nul est par convention égal à -1 .

Q 16. Montrer qu'il existe un entier naturel $n(T)$ tel que, pour tout polynôme $p \in \mathbb{K}[X]$,

$$\deg(Tp) = \max\{-1, \deg(p) - n(T)\}.$$

Q 17. En déduire $\ker(T)$ en fonction de $n(T)$.

Q 18. Montrer que les trois assertions suivantes sont équivalentes :

- (1) T est inversible ;
- (2) $T1 \neq 0$;
- (3) $\forall p \in \mathbb{K}[X], \deg(Tp) = \deg(p)$.

Q 19. Si ces conditions sont vérifiées, montrer que T^{-1} est encore un endomorphisme shift-invariant.

II.E – Dans cette sous-partie, T est un endomorphisme delta de $\mathbb{K}[X]$.

Q 20. Montrer qu'il existe une suite de scalaires $(\alpha_k)_{k \in \mathbb{N}}$ vérifiant $\alpha_0 = 0, \alpha_1 \neq 0$ et $T = \sum_{k=1}^{+\infty} \alpha_k D^k$.

Q 21. Montrer qu'il existe un unique endomorphisme U shift-invariant et inversible tel que $T = D \circ U$. Préciser U dans le cas $T = D$, puis dans le cas $T = L$.

Q 22. Pour tout polynôme $p \in \mathbb{K}[X]$ non nul, vérifier que $\deg(Tp) = \deg(p) - 1$. En déduire $\ker(T)$ et le spectre de T .

Q 23. Pour $n \in \mathbb{N}$, on note T_n la restriction de T à $\mathbb{K}_n[X]$. Montrer que T_n est un endomorphisme de $\mathbb{K}_n[X]$. Est-il diagonalisable ?

Q 24. Déterminer $\text{Im}(T_n)$ en fonction de $n \in \mathbb{N}$ et en déduire que T est surjectif.

III Suite de polynômes associée à un endomorphisme delta

On souhaite montrer que, pour tout endomorphisme delta Q , il existe une unique suite de polynômes $(q_n)_{n \in \mathbb{N}}$ de $\mathbb{K}[X]$ telle que

- $q_0 = 1$;
- $\forall n \in \mathbb{N}, \deg(q_n) = n$;
- $\forall n \in \mathbb{N}^*, q_n(0) = 0$;
- $\forall n \in \mathbb{N}^*, Qq_n = q_{n-1}$.

Cette suite sera appelée *suite de polynômes associée* à l'endomorphisme delta Q .

III.A – Soit Q un endomorphisme delta.

Q 25. Montrer l'existence et l'unicité de la suite $(q_n)_{n \in \mathbb{N}}$ de polynômes associée à Q .

Q 26. Montrer que, pour tout entier naturel n ,

$$\forall (x, y) \in \mathbb{K}^2, \quad q_n(x+y) = \sum_{k=0}^n q_k(x)q_{n-k}(y).$$

III.B – Réciproquement, soit $(q_n)_{n \in \mathbb{N}}$ une suite de polynômes de $\mathbb{K}[X]$ telle que $\forall n \in \mathbb{N}, \deg(q_n) = n$ et

$$\forall (x, y) \in \mathbb{K}^2, \quad q_n(x+y) = \sum_{k=0}^n q_k(x)q_{n-k}(y).$$

Q 27. Montrer qu'il existe un unique endomorphisme delta Q dont $(q_n)_{n \in \mathbb{N}}$ est la suite de polynômes associée.

III.C – Soit Q un endomorphisme delta, soit $(q_n)_{n \in \mathbb{N}}$ la suite de polynômes associée à Q et soit n un entier naturel.

Q 28. Montrer que la famille $(q_0, q_1, \dots, q_n)$ est une base de $\mathbb{K}_n[X]$.

Q 29. D'après la question 23, Q induit un endomorphisme de $\mathbb{K}_n[X]$ noté Q_n . Donner sa matrice dans la base précédente. En déduire sa trace, son déterminant et son polynôme caractéristique.

III.D – Dans cette sous-partie, on détermine la suite $(q_n)_{n \in \mathbb{N}}$ de polynômes associée à certains endomorphismes.

Q 30. Pour $Q = D$, vérifier que

$$\forall n \in \mathbb{N}, \quad q_n = \frac{X^n}{n!}.$$

Q 31. Pour $Q = E_1 - I$, vérifier que

$$\forall n \in \mathbb{N}^*, \quad q_n = \frac{X(X-1)\cdots(X-n+1)}{n!}.$$

III.E – Cette sous-partie propose de généraliser la formule de Taylor démontrée dans la partie II. On se donne Q un endomorphisme delta et on note $(q_n)_{n \in \mathbb{N}}$ la suite de polynômes associée à Q .

Q 32. Démontrer que, pour tout $p \in \mathbb{K}[X]$, l'expression $\sum_{k=0}^{+\infty} (Q^k p)(0) q_k$ a un sens et définit un polynôme de $\mathbb{K}[X]$, puis que

$$p = \sum_{k=0}^{+\infty} (Q^k p)(0) q_k.$$

Q 33. En déduire que, pour tout endomorphisme shift-invariant T , on a

$$T = \sum_{k=0}^{+\infty} (T q_k)(0) Q^k.$$

III.F –

Q 34. En choisissant $Q = E_1 - I$, démontrer que, si p est un polynôme non constant, alors

$$p'(X) = \sum_{k=1}^{\deg(p)} \frac{1}{k} \left(\sum_{j=0}^k (-1)^{j+1} \binom{k}{j} p(X+j) \right).$$

C'est la *formule de dérivation numérique des polynômes*.

IV Un peu de calcul ombral

Si T est un endomorphisme de $\mathbb{K}[X]$ on définit sa dérivée de Pincherle, notée T' , comme l'endomorphisme de $\mathbb{K}[X]$ tel que,

$$\forall p \in \mathbb{K}[X], \quad T'(p) = T(Xp) - XT(p).$$

IV.A – Soient S et T deux endomorphismes de $\mathbb{K}[X]$.

Q 35. Montrer que, s'il existe $(a_n)_{n \in \mathbb{N}}$ suite de scalaires telle que $T = \sum_{k=0}^{+\infty} a_k D^k$, alors $T' = \sum_{k=1}^{+\infty} k a_k D^{k-1}$.

Q 36. Si T est un endomorphisme shift-invariant, montrer que T' est encore un endomorphisme shift-invariant.

Q 37. Si T est un endomorphisme delta, montrer que T' est un endomorphisme shift-invariant et inversible.

Q 38. Vérifier que $(S \circ T)' = S' \circ T + S \circ T'$.

IV.B – Soit Q un endomorphisme delta. On rappelle que d'après la partie II, il existe un unique endomorphisme U shift-invariant et inversible tel que $Q = D \circ U$. On note $(q_n)_{n \in \mathbb{N}}$ la suite de polynômes associée à Q au sens de la partie III.

Q 39. Démontrer que, pour tout $n \in \mathbb{N}^*$, on a

$$(Q' \circ U^{-n-1})(X^n) = X U^{-n}(X^{n-1}).$$

Q 40. En déduire que, pour tout $n \in \mathbb{N}^*$,

$$n! q_n(X) = X U^{-n}(X^{n-1})$$

puis que

$$n q_n(X) = X(Q')^{-1}(q_{n-1}).$$

IV.C – Dans cette sous-partie, on applique les résultats de la question 40 à l'endomorphisme L étudié dans les parties I et II. On note $(\ell_n)_{n \in \mathbb{N}}$ sa suite de polynômes associée au sens de la partie III.

Q 41. Vérifier que, pour $n \in \mathbb{N}^*$,

$$\ell'_n = \ell'_{n-1} - \ell_{n-1}$$

et

$$X \ell''_n - X \ell'_n + n \ell_n = 0$$

et

$$\ell_n(X) = \sum_{k=1}^n (-1)^k \binom{n-1}{k-1} \frac{X^k}{k!}.$$

IV.D – Soient Q un endomorphisme delta de suite de polynômes associée notée $(q_n)_{n \in \mathbb{N}}$.

Q 42. Montrer qu'il existe un unique endomorphisme inversible T tel que

$$\forall n \in \mathbb{N}, \quad Tq_n = \frac{X^n}{n!}.$$

Q 43. Montrer aussi que $D = T \circ Q \circ T^{-1}$.

IV.E – On fixe $\alpha > 0$ et on définit la fonction W de $\mathbb{K}[X]$ par

$$W : \begin{cases} \mathbb{K}[X] & \rightarrow \mathbb{K}[X] \\ p & \mapsto p(\alpha X) \end{cases}$$

Q 44. Montrer que W est un automorphisme de $\mathbb{K}[X]$.

On pose $P = W \circ L \circ W^{-1}$ où L est l'endomorphisme étudié dans les parties I et II.

Q 45. Montrer que

$$P = \frac{1}{\alpha} D \circ \left(\frac{1}{\alpha} D - I \right)^{-1}.$$

Q 46. Montrer ensuite que P est un endomorphisme delta dont la suite de polynômes associée $(p_n)_{n \in \mathbb{N}}$ vérifie

$$\forall n \in \mathbb{N}, \quad p_n = \ell_n(\alpha X).$$

Q 47. Vérifier que $D = L \circ (L - I)^{-1}$ puis que $P = L \circ (\alpha I + (1 - \alpha)L)^{-1}$.

On note T l'unique automorphisme vérifiant, pour tout $n \in \mathbb{N}$, $T\ell_n = \frac{X^n}{n!}$ et on pose $Q = T \circ P \circ T^{-1}$.

Q 48. Montrer que $Q = D \circ (\alpha I + (1 - \alpha)D)^{-1}$. En déduire que Q est un endomorphisme delta dont la suite de polynômes associée $(r_n)_{n \in \mathbb{N}}$ vérifie

$$\forall n \in \mathbb{N}^*, \quad r_n = \sum_{k=1}^n \binom{n-1}{k-1} \alpha^k (1-\alpha)^{n-k} \frac{X^k}{k!}.$$

Q 49. Conclure que

$$\forall n \in \mathbb{N}^*, \quad \ell_n(\alpha X) = \sum_{k=1}^n \binom{n-1}{k-1} \alpha^k (1-\alpha)^{n-k} \ell_k(X).$$

Les endomorphismes W et T étudiés dans la sous-partie IV.E sont appelés opérateurs ombraux. Les polynômes (ℓ_n) associés à l'endomorphisme L sont connus sous le nom de polynômes de Laguerre (de paramètre -1). La dernière formule démontrée grâce aux opérateurs ombraux est leur formule de duplication.

• • • FIN • • •

SOLUTION

Etude d'endomorphismes de $\mathbb{K}[X]$

1. On a bien sûr

$$\forall \lambda \in \mathbb{K}, \forall p, q \in \mathbb{K}[X], E_a(p + \lambda q) = E_a(p) + \lambda E_a(q)$$

c'est à dire la linéarité de E_a . On vérifie que

$$E_a \circ E_{-a} = E_{-a} \circ E_a = I$$

ce qui montre la bijectivité de E_a de $\mathbb{K}[X]$ dans lui même : E_a est un automorphisme de $\mathbb{K}[X]$.

On est en première question et il faut éviter de dire sans rien de plus que E_a est linéaire. Cependant, la linéarité est vraiment immédiate. Ecrire ce que cela signifie permet de contourner le problème. Pour la bijectivité, il faut prendre garde au fait que $\mathbb{K}[X]$ n'est pas de dimension finie. L'injectivité ou la surjectivité ne suffisent pas à conclure. Et si on choisit, comme ici, d'exhiber l'inverse, il faut bien considérer la composition dans les deux sens.

2. Soit
- $p \in \mathbb{R}[X]$
- .
- p
- possède une primitive
- p_1
- qui est encore un polynôme. Par théorème fondamental,

$$\forall x \in \mathbb{R}, J(p)(x) = p_1(x+1) - p_1(x)$$

et $J(p)$ est un polynôme ($J(p) \in \mathbb{R}[X]$).

Soient $p, q \in \mathbb{R}[X]$ et $\lambda \in \mathbb{R}$. Par linéarité du passage à l'intégrale,

$$\forall x \in \mathbb{R}, J(p + \lambda q)(x) = J(p)(x) + \lambda J(q)(x)$$

et par identification polynôme-fonction polynomiale, $J(p + \lambda q) = J(p) + \lambda J(q)$: $J \in \mathcal{L}(\mathbb{R}[X])$.

Il y a bien deux points à noter : linéarité et stabilité de $\mathbb{R}[X]$. Nous insistons, ici, sur l'identification car c'est la première fois qu'elle sert. L'argument n'est pas indispensable au vu de ce qu'annonce le préambule.

3. On a

$$\forall n \in \mathbb{N}, J(X^n) = \frac{1}{n+1}((X+1)^{n+1} - X^{n+1}) = \frac{1}{n+1} \sum_{k=0}^n \binom{n+1}{k} X^k$$

On a donc

$$\forall n \in \mathbb{N}, \deg(J(X^n)) = n = \deg(X^n)$$

Une somme de polynômes de degrés distincts étant de degré égal au maximum des degrés, on en déduit par linéarité que p et $\deg(J(p))$ ont même degré quand $p \neq 0$. Le résultat reste vrai si $p = 0$:

$$\forall p \in \mathbb{R}[X], \deg(J(p)) = \deg(p)$$

J est linéaire et transforme la base canonique de $\mathbb{R}[X]$ en une base de $\mathbb{R}[X]$ (puisque l'image est une famille échelonnée en degré). On a donc

$$J \text{ est inversible}$$

On pourrait, bien sûr, considérer un polynôme p quelconque sous forme de somme et calculer les coefficients de Jp en fonction de ceux de p . Il est cependant bien plus élégant de tirer partie de la linéarité en utilisant une base et l'action de Jp sur celle-ci. De même, l'inversibilité peut être découpée en injectivité et surjectivité mais la façon dont l'énoncé est posé mène assez naturellement à étudier l'action sur $\mathbb{R}_n[X]$ et à utiliser l'échelonnement en degré.

4. Soit $k \in \mathbb{N}$. $f_k : t \mapsto e^{-t}t^k$ est continue sur $\mathbb{R}^+$ et $o(1/t^2)$ au voisinage de $+\infty$ (croissances comparées). C'est donc une fonction intégrable sur $\mathbb{R}^+$ et a fortiori I_k existe pour tout $k \in \mathbb{N}$. Sous réserve de validité, une intégration par parties donne

$$\forall k \in \mathbb{N}, \int_0^{+\infty} t^{k+1} e^{-t} dt = [-t^{k+1} e^{-t}]_0^{+\infty} + (k+1) \int_0^{+\infty} t^k e^{-t} dt$$

Le terme dans le crochet admet une limite finie en $+\infty$, ce qui légitime le calcul et donne

$$\forall k \in \mathbb{N}, \int_0^{+\infty} t^{k+1} e^{-t} dt = (k+1) \int_0^{+\infty} t^k e^{-t} dt$$

Comme $(k+1) \int_0^{+\infty} e^{-t} dt = 1$, une récurrence donne immédiatement

$$\forall k \in \mathbb{N}, \int_0^{+\infty} t^k e^{-t} dt = k!$$

On effectue ici une intégration par parties dans une intégrale généralisée et il est indispensable de prendre des précautions car cette opération peut faire apparaître deux termes qui n'existent pas mais dont la somme existe. Vous pouvez faire la vérification de la validité après l'intégration par parties mais il est bon, surtout en début de problème, de l'annoncer avant. Vous reconnaîtrez dans l'intégrale calculée la valeur de $\Gamma(n+1)$ où $\Gamma : x \mapsto \int_0^{+\infty} e^{-t} t^{x-1} dt$ est la célèbre fonction d'Euler qui fait l'objet de bien des problèmes de concours.

5. Pour $p \in \mathbb{K}[X]$, $t \mapsto e^{-t}p'(x+t)$ est continue sur $\mathbb{R}^+$ et $o(1/t^2)$ au voisinage de $+\infty$ (croissances comparées). $L(p)(x)$ est donc bien défini.

La linéarité de p découle de la linéarité de la dérivation et de celle du passage à l'intégrale. La question précédente donne

$$\forall n \in \mathbb{N}, \forall x \in \mathbb{R}, L(X^n)(x) = -n \int_0^{+\infty} (x+t)^{n-1} e^{-t} dt = -n \sum_{k=0}^{n-1} \binom{n-1}{k} (n-1-k)! x^k$$

$L(X^n) \in \mathbb{K}[X]$ et par linéarité, L est à valeurs dans $\mathbb{K}[X]$.

Enfin, $L(1) = 0$ et 1 est élément non nul du noyau. Ainsi

L est un endomorphisme non inversible de $\mathbb{K}[X]$

Pour montrer qu'une propriété n'a pas lieu, un "contre-exemple" suffit. Ici, pas besoin d'exhiber le noyau en entier.

Formule de Taylor pour les endomorphismes shift-invariants de $\mathbb{K}[X]$

6. Un endomorphisme T est shift-invariant si

$$\forall b \in \mathbb{K}, \forall p \in \mathbb{K}[X], (T(p))(X+b) = T(p(X+b))$$

- $(I(p))(X+b) = p(X+b) = I(p(X+b))$: I est shift-invariant.
- $(D(p))(X+b) = p'(X+b) = (p(X+b))' = D(p(X+b))$: D est shift-invariant.
- $E_a(p)(X+b) = p(X+a+b) = E_a(p(X+b))$: E_a est shift-invariant.
- Le changement de variable $u = t - b$ donne

$$(J(p))(x+b) = \int_{x+b}^{x+b+1} p(t) dt = \int_x^{x+1} p(u+b) du = J(p(X+b))(x)$$

et ainsi J est shift-invariant.

- On a aussi

$$(L(p))(x+b) = - \int_0^{+\infty} e^{-t} p'(x+b+t) dt$$

Comme $(p(X+b+t))' = p'(X+b+t)$, ceci est aussi égal à $L(p(X+b))(x)$ et L est shift-invariant.

$IX = X$, $DX = 1$, $E_a X = X + a$, JX est de degré 1, $L(X) = -1$ et donc

D et L sont delta mais pas I , E_a et J

Cette question peut sembler un peu fastidieuse. Elle n'en est pas moins importante car d'une part tout le sujet va tourner autour de ces endomorphismes et d'autre part elle permet de se familiariser avec les notations. Certains points ne sont pas "si triviaux". Ainsi, quand on écrit $(p(X+b))' = p'(X+b)$ on montre que l'on comprend qu'il y a une dérivation composée.

7. Soient T_1, T_2 des endomorphismes shift-invariants. Soit $\lambda \in \mathbb{K}$. Pour tout $p \in \mathbb{K}[X]$, on a

$$E_a \circ (T_1 + \lambda T_2) = E_a \circ T_1 + \lambda E_a \circ T_2 = T_1 \circ E_a + T_2 \circ E_a = (T_1 + \lambda T_2) \circ E_a$$

(la première égalité provient de la linéarité de E_a , la seconde des shift-invariances, la troisième de la définition de la composition)

$$E_a \circ (T_1 \circ T_2) = (E_a \circ T_1) \circ T_2 = (T_1 \circ E_a) \circ T_2 = T_1 \circ (E_a \circ T_2) = T_1 \circ (T_2 \circ E_a) = (T_1 \circ T_2) \circ E_a$$

(cette fois, on utilise la distributivité de la loi $\circ$ et les shift-invariances).

L'ensemble des endomorphismes shift-invariants est inclus dans $\mathcal{L}(\mathbb{R}[X])$, est stable par combinaisons linéaires, par composition et contient le neutre I et donc

L'ensemble des endomorphismes shift-invariants est une sous-algèbre de $\mathcal{L}(\mathbb{R}[X])$

D et $-D$ sont delta mais $D + (-D) = 0$ ne l'est pas. $D \circ D(X) = D(1) = 0 \notin \mathbb{K}^*$ ainsi

L'ensemble des endomorphismes delta n'est stable ni par addition ni par composition

Pour vérifier qu'on a une sous-algèbre, on doit vérifier qu'on a sous-espace vectoriel (non vide, stable par combinaisons linéaires) et sous-anneau (stabilité par addition, passage à l'opposé, composition et l'ensemble contient le neutre multiplicatif). Notez bien que $(f+g) \circ h = (f \circ h) + (g \circ h)$ est toujours vrai si les compositions existent. En revanche $f \circ (g+h)$ ne l'est en général pas. On a donc distributivité d'un côté mais pas de l'autre.

8. Soit $p \in \mathbb{K}[X]$. Si $p = 0$ alors $\forall k \in \mathbb{N}$, $D^k p = 0$. Sinon, $\forall k \geq \deg(p) + 1$, $D^k(p) = 0$. La somme $\sum_{k=0}^{+\infty} a_k D^k(p)$ est donc finie (termes nuls à partir d'un certain rang). Comme elle est composée de polynômes, c'est un polynôme.

$$\sum_{k=0}^{\infty} a_k D^k(p) \in \mathbb{K}[X]$$

Avec la convention choisie pour le degré du polynôme nul, on pourrait ne pas à distinguer les cas. En général, il est conseillé de particulariser le cas du polynôme nul dans un raisonnement qui utilise le degré.

9. Fixons $(a_k)_{k \in \mathbb{N}} \in \mathbb{K}^{\mathbb{N}}$ et notons $T = \sum_{k=0}^{\infty} a_k D^k$. On a

$$\forall n \in \mathbb{N}, \forall p \in \mathbb{K}_n[X], T(p) = \sum_{k=0}^n a_k D^k(p)$$

$T_n = \sum_{k=0}^n a_k D^k$ est un endomorphisme shift-invariant comme combinaison linéaires de tels endomorphismes. Si $p \in \mathbb{K}[X]$, on peut considérer un majorant n du degré de p et on a

$$T(p)(X+a) = T_n(p)(X+a) = T_n(p(X+a)) = T(p(X+a))$$

la dernière égalité étant vraie car $p(X+a)$ est aussi de degré $\leq n$.

Et de façon similaire, en considérant $p, q \in \mathbb{K}[X]$, $\lambda \in \mathbb{K}$ et un majorant des degrés de p et q , on montre que $T(p + \lambda q) = T(p) + \lambda T(q)$. Ainsi

$$\sum_{k=0}^{\infty} a_k D^k \text{ est un endomorphisme shift-invariant}$$

Attention : l'endomorphisme T ci-dessus est défini comme une somme infinie. Quand on écrit Tp , on obtient une somme finie mais dont le nombre de termes dépend de p . Il n'est donc pas possible de conclure directement en parlant de somme d'endomorphismes, erreur que l'on a du retrouver dans de nombreuses copies.

10. $D^k(X^n)(0) = \delta_{n,k} n!$ car $D^k(X^n) = 0$ si $k \geq n+1$ et vaut $\frac{n!}{(n-k)!} X^{n-k}$ sinon. Ainsi

$$\left(\sum_{k=0}^{\infty} a_k D^k(X^n) \right) (0) = n! a_n$$

On en déduit que

$$\sum_{k=0}^{\infty} a_k D^k = \sum_{k=0}^{\infty} b_k D^k \implies (\forall k \in \mathbb{N}, a_k = b_k)$$

11. Le calcul mené à la question précédente montre que

$$D^k(q_n)(0) = \delta_{k,n}$$

Considérons un endomorphisme T et posons

$$T_1 = \sum_{k=0}^{\infty} (Tq_k)(0) D^k$$

On a en particulier (en appliquant l'égalité en q_n ce qui donne une égalité entre polynôme et en évaluant cette égalité en 0)

$$\forall n \in \mathbb{N}, (T_1 q_n)(0) = (T q_n)(0)$$

Comme T_1 et T sont linéaires et comme $(q_n)_{n \in \mathbb{N}}$ est une base de $\mathbb{K}[X]$, on en déduit par combinaisons linéaires que

$$\forall p \in \mathbb{K}[X], (T_1 p)(0) = (T p)(0)$$

Cette propriété est vraie sans la shift-invariance. C'est elle qui va nous permettre de prouver le résultat demandé.

Supposons que T est shift-invariant. Pour tout $a \in \mathbb{K}$, $(T p)(X + a) = T(p(X + a))$ et en appliquant cette égalité avec q_n ;

$$(T q_n)(X + a) = T(q_n(X + a)) = \frac{1}{n!} \sum_{k=0}^n \binom{n}{k} a^{n-k} T(X^k)$$

Evaluons cette identité polynomiale en 0 :

$$(T q_n)(a) = \frac{1}{n!} \sum_{k=0}^n \binom{n}{k} a^{n-k} T(X^k)(0) = \frac{1}{n!} \sum_{k=0}^n \binom{n}{k} a^{n-k} T_1(X^k)(0)$$

Comme T_1 est shift-invariant (question 9), le même calcul avec T_1 donne

$$(T q_n)(a) = (T_1 q_n)(a)$$

Ceci étant vrai pour tout a , les polynômes $T q_n$ et $T_1 q_n$ sont égaux. Par linéarité et comme (q_n) est une base de $\mathbb{K}[X]$, les endomorphismes T et T_1 sont égaux.

Réciproquement, si $T = T_1$ alors T est shift-invariant puisque T_1 l'est.

L'endomorphisme T est shift-invariant ssi $T = \sum_{k=0}^{\infty} (T q_k)(0) D^k$

12. Commençons par montrer que si V est shift-invariant, alors V et D commutent. On se donne un tel V . Soit $p \in \mathbb{K}[X]$ et n un majorant du degré de p . On a

$$V(p) = \sum_{k=0}^{\infty} (V q_k)(0) D^k p = \sum_{k=0}^n (V q_k)(0) D^k p$$

$$D \circ V(p) = D(V(p)) = \sum_{k=0}^n (V q_k)(0) D^{k+1} p = V(D(p))$$

puisque $D(p)$ est aussi de degré $\leq n$. On a donc $D \circ V = V \circ D$.

Une itération donne facilement que D^k et V commutent pour tout $k \in \mathbb{N}$.

Soient alors U, V deux endomorphismes shift-invariant. Soit $p \in \mathbb{K}[X]$. Il existe un entier n tel que p et $V(p)$ sont tous deux de degré $\leq n$. On a alors

$$V \circ U(p) = V \left(\sum_{k=0}^n (U q_k)(0) D^k p \right) = \sum_{k=0}^n (U q_k)(0) V(D^k p) = \sum_{k=0}^n (U q_k)(0) D^k (V(p)) = U(V(p))$$

(on peut limiter les sommes à n avec le choix que nous avons fait de n car $D^k q = 0$ si $k > \deg(q)$).

Ceci étant vrai pour tout p , U et V commutent.

Deux endomorphismes shift-invariants commutent

Comme en question 9, il faut prendre garde au fait que les expressions sont définies avec des sommes infinies et qu'il n'est a priori pas possible de manipuler comme si elles étaient finies (sauf à ne faire qu'un calcul formel). L'avantage de la tactique ci-dessus est qu'elle évite d'avoir à travailler avec des sommes doubles et de trop grosses expressions. Elle me semble naturelle au sens où la question 11 indique, moralement, que si on sait ce qui se passe pour D et ses itérées, on sait ce qui se passe pour tous les endomorphismes shift-invariants.

13. E_a étant un endomorphisme shift-invariant, la question 11 donne

$$E_a = \sum_{k=0}^{\infty} (E_a q_k)(0) D^k$$

c'est à dire

$$\forall p \in \mathbb{K}[X], p(X+a) = \sum_{k=0}^{\infty} (q_k(X+a))(0) p^{(k)}$$

L'évaluation en 0 de $q_k(X+a) = \frac{(X+a)^k}{k!}$ vaut $\frac{a^k}{k!}$ et on a donc

$$\forall p \in \mathbb{K}[X], p(X+a) = \sum_{k=0}^{\infty} \frac{a^k}{k!} p^{(k)}$$

Comme $p^{(k)} = 0$ si $k > \deg(p)$, on a même

$$\forall p \in \mathbb{K}[X], p(X+a) = \sum_{k=0}^{\deg(p)} \frac{a^k}{k!} p^{(k)}$$

La formule est vraie pour $p = 0$ en convenant qu'une somme vide est nulle. On reconnaît la formule de Taylor

Dans cette question, tout est affaire de notations. Il faut ainsi juste bien comprendre ce que signifie $(E_a q)(0)$. La formule de Taylor pour un polynôme s'écrit usuellement

$$p(b) = \sum_{k=0}^{\deg(p)} \frac{(b-a)^k}{k!} p^{(k)}(a)$$

Ici, on l'utilise en remplaçant b par $y+a$ et n interprétant l'égalité fonctionnelle comme une égalité polynomiale de la variable a . Peut-être aurait-il fallu développer ce point.

14. J étant un endomorphisme shift-invariant, on peut lui appliquer la question 11 :

$$\forall p \in \mathbb{K}[X], Jp = \sum_{k=0}^{\infty} (Jq_k)(0) p^{(k)}$$

q_{k+1} est une primitive de q_k et donc

$$Jq_k(0) = q_{k+1}(1) - q_{k+1}(0) = \frac{1}{(k+1)!}$$

On a ainsi

$$\forall p \in \mathbb{K}[X], \quad Jp = \sum_{k=0}^{\infty} \frac{1}{(k+1)!} p^{(k)}$$

On pourrait bien sûr sommer seulement jusqu'au rang $\deg(p)$.

15. $(D - I)(X^n) = nX^{n-1} - X^n$ et la famille $((D - I)(X^n))$ est une base de $\mathbb{K}[X]$ (échélonnée en degré). L'endomorphisme $D - I$ transforme une base en une base et donc $D - I$ est inversible.

Soit $p \in \mathbb{K}[X]$. Une intégration par parties donne, sous réserve de validité

$$\begin{aligned} (L \circ D)(p)(x) &= - \int_0^{\infty} e^{-t} p''(x+t) dt \\ &= [e^{-t} p'(x+t)]_{t=0}^{t=+\infty} - \int_0^{+\infty} e^{-t} p'(x+t) dt \\ &= p'(x) + L(p)(x) \end{aligned}$$

l'intégration par partie étant validée car les différentes quantités existent. On a montré que $L \circ D = D + L$ ou encore $L \circ (D - I) = D$ (par linéarité de L). $D - I$ étant inversible, on conclut que

$$L = D \circ (D - I)^{-1}$$

Comme en question 1, il faut noter que l'on ne travaille pas en dimension finie. L'inversibilité de l'endomorphisme $D - I$ n'équivaut donc pas à sa bijectivité. Cette question nécessite un peu d'initiative pour "mélanger" L et D et on aurait aussi pu considérer $(D \circ L)(p)$ ce qui nous aurait amené à $L = D \circ (D - I)^{-1}$. Le fait que D et $(D - I)^{-1}$ commutent est immédiat car c'est le cas de D et $D - I$.

16. T étant shift-invariant, on peut lui appliquer la question 11 :

$$T = \sum_{k=0}^{\infty} (Tq_k)(0) D^k$$

T étant supposé non nul, l'un des $(Tq_k)(0)$ est non nul. On peut donc poser

$$n(T) = \min\{k \in \mathbb{N}, (Tq_k)(0) \neq 0\}$$

puisque une partie non vide de $\mathbb{N}$ possède un minimum.

Considérons alors $p \in \mathbb{K}[X]$. On a

$$Tp = \sum_{k=n(T)}^{\infty} (Tq_k)(0) p^{(k)}$$

Si $\deg(p) \leq n(T) - 1$ alors tous les termes dans la somme sont nuls ($p^{(k)} = 0$ si $k \geq n(T)$) et $\deg(Tp) = -1$.

Sinon, le terme d'indice $k = n(T)$ est de degré $\deg(p) - n(T)$ et les autres sont de degré strictement inférieur (le terme d'indice k est nul ou de degré $\deg(p) - k$). La somme est donc de degré $\deg(p) - n(T)$. Ainsi (formule valable dans les deux cas)

$$\forall p \in \mathbb{K}[X], \quad \deg(Tp) = \max\{-1, \deg(p) - n(T)\}$$

Il faut bien comprendre que l'on veut un entier $n(T)$ qui ne dépend que de T et donc pas de p . Il convient donc de définir $n(T)$ PUIS d'introduire un p quelconque pour vérifier les relations.

17. Tp est nul quand son degré vaut -1 , c'est à dire quand $\deg(p) - n(T) \leq -1$ et

$$\ker(T) = \mathbb{K}_{n(T)-1}[X]$$

18. Procédons de manière circulaire.

Supposons T inversible. Son noyau est réduit au polynôme nul et ainsi $T1 \neq 0$.

Supposons $T1 \neq 0$. On a donc $n(T) = 0$ (puisque $q_0 = 1$ et avec l'expression trouvée de $n(T)$) et pour tout p , $\deg(Tp) = \max\{-1, \deg(p)\} = \deg(p)$.

Supposons que pour tout p , $\deg(Tp) = \deg(p)$. Alors, T est un endomorphisme qui transforme la base canonique en une base (échelonnement en degré) et T est inversible.

Les propriétés (1), (2) et (3) sont équivalentes

19. T étant shift-invariant, pour tout a on a $E_a \circ T = T \circ E_a$. Il suffit alors de composer par T^{-1} à gauche et droite pour en déduire que

T^{-1} est aussi shift-invariant

20. T étant shift-invariant, on peut lui appliquer la question 11 et

$$T = \sum_{k=0}^{\infty} (Tq_k)(0) D^k$$

$Tq_1 = TX$ est un scalaire α_1 non nul et donc $\alpha_1 = (Tq_1)(0) \neq 0$.

Si, par l'absurde, $T1$ est non nul, la question 18 indique que T est inversible et donc que $\deg(TX) = 1$ ce qui contredit $TX \in \mathbb{K}^*$. On a donc $T1 = 0$ et donc $(Tq_1)(0) = T1(0) = 0$.

Il existe une suite $(\alpha_k)_{k \geq 1}$ avec $\alpha_1 \neq 0$ telle que $T = \sum_{k=1}^{\infty} \alpha_k D^k$

21. Supposons que U convienne. Par la question 12, U et D commutent et donc $U \circ D = T$. En particulier,

$$\forall n \in \mathbb{N}, U(q_n) = U(D(q_{n+1})) = T(q_{n+1})$$

L'action de U est ainsi connue sur une base et, par linéarité, U est parfaitement connu.

Réciproquement, notons U l'unique endomorphisme de $\mathbb{K}[X]$ tel que $\forall n, U(q_n) = T(q_{n+1})$. Comme D et T commutent (question 12), on a

$$\forall n, D \circ U(q_n) = T(D(q_{n+1})) = T(q_n)$$

et les endomorphismes $D \circ U$ et T sont égaux (puisque'ils coïncident sur une base).

Il existe un unique U shift-invariant tel que $T = D \circ U$

Une question où il s'agit de montrer une existence est toujours un peu délicate. Deux options s'offrent à nous : exhiber l'objet ou raisonner par l'absurde. Ici, on veut même un

peu plus puisque l'on demande aussi l'unicité. La méthode naturelle est le raisonnement par analyse et synthèse. A la fin de l'étape d'analyse, on doit disposer du seul objet qui peut convenir et la synthèse est alors une simple vérification.

La difficulté est donc, dans l'étape d'analyse, de trouver des choses qui caractérisent U . Dans le cas d'une application linéaire, essayer d'obtenir l'image d'une base doit vous être naturel.

Si $T = D$, $U = I$ convient et si $T = L$, $U = (D - I)^{-1}$ convient en utilisant les questions 15 ($L = D \circ (D - I)^{-1}$) et 19 (la stabilité des shift-invariant par combinaisons linéaire et passage à l'inverse donne le caractère shift-invariant de $(D - I)^{-1}$).

22. Avec les notations de la partie II.D, on a cette fois $n(T) = 1$. Si $\deg(p) - 1 \geq -1$, c'est à dire si p est non nul, on a donc $\max\{-1, \deg(p) - 1\} = \deg(p) - 1$ et

$$\forall p \in \mathbb{K}[X] \setminus \mathbb{K}, \deg(Tp) = \deg(p) - 1$$

En particulier, aucun polynôme de degré ≥ 1 n'est dans le noyau de T . De plus, $T1 = 0$ (car $D^k 1 = 0$ pour $k \geq 1$) et ains

$$\ker(T) = \mathbb{K}$$

Si $T(p) = \lambda p$ avec λ scalaire non nul alors $\deg(T(p)) = \deg(p)$ ce qui entraîne $p = 0$. Un scalaire non nul n'est donc pas valeur propre et

$$\text{Sp}(T) = \{0\}$$

Attention, nous ne sommes pas en dimension finie et même si $\mathbb{K} = \mathbb{C}$, ceci n'entraîne pas la nilpotence de T . Pensez à l'endomorphisme D dont la seule valeur propre est 0 mais tel que $\forall n, D^n X^n = n! \neq 0$ et qui n'est donc pas nilpotent. La nilpotence viendra en se limitant à un espace de dimension finie, ce qui vient ensuite.

23. La question 22 montre que $T(\mathbb{K}_n[X]) \subset \mathbb{K}_{n-1}[X] \subset \mathbb{K}_n[X]$. Ainsi $\mathbb{K}_n[X]$ est stable par T et

$$T \text{ induit un endomorphisme } T_n \text{ sur } \mathbb{K}_n[X]$$

$\text{Sp}(T_n) \subset \text{Sp}(T)$ et comme $1 \in \mathbb{K}_n[X]$, 1 est vecteur propre de T_n associé à 0. 0 est donc l'unique valeur propre de T_n et T_n n'est diagonalisable que s'il est nul ce qui n'a lieu que si $n = 0$.

$$T_n \text{ est non diagonalisable sauf si } n = 0$$

Ici, T_n est un endomorphisme de dimension finie dont 0 est la seule valeur propre. Si le corps de base est $\mathbb{C}$, le cours nous indique que T_n est nilpotent. En effet, si $u \in \mathcal{L}(E)$ avec E qui est un $\mathbb{C}$ -espace de dimension d alors χ_u est le produit des $(X - \lambda)^{m_\lambda}$ où les λ sont les valeurs propres et les m_λ les multiplicités. Si 0 est la seule valeur propre alors $\chi_u = X^d$ et par théorème de Cayley-Hamilton, $u^d = 0$.

24. La question 22 montre que l'image par T de la famille $(X^k)_{0 \leq k \leq n}$ est une base de $\mathbb{K}_{n-1}[X]$.

On a donc $\text{Im}(T_n) = \mathbb{K}_{n-1}[X]$. Si $p \in \mathbb{K}[X]$, on peut trouver un entier n tel que $p \in \mathbb{K}_n[X]$ et p possède un antécédent par T dans $\mathbb{K}_{n+1}[X]$. T est surjectif.

Suite de polynômes associée à un endomorphisme delta

25. D'après II.E, Q induit sur $\mathbb{K}_n[X]$ un endomorphisme de noyau $\mathbb{K}$ et d'image $\mathbb{K}_{n-1}[X]$. Comme $X\mathbb{K}_{n-1}[X]$ est un supplémentaire de $\mathbb{K}$ dans $\mathbb{K}_n[X]$ (par théorème de division euclidienne par exemple), le théorème du rang indique que Q_n induit un isomorphisme $\widetilde{Q}_n$ de $X\mathbb{K}_{n-1}[X]$ dans $\mathbb{K}_{n-1}[X]$. Si on pose

$$q_0 = 1 \text{ et } \forall n \in \mathbb{N}^*, q_n = \widetilde{Q}_n^{-1}(q_{n-1})$$

on dispose alors d'une suite convenable ($q_k(0) = 0$ pour $k \geq 1$ car l'image de $\widetilde{Q}_k^{-1}$ est $X\mathbb{K}_{k-1}[X]$, $Qq_k = q_{k-1}$ par construction et la question 22 donne $\deg(q_k) = k$ par récurrence à partir de $\deg(q_0) = 1$).

J'ai utilisé ici la version générale du théorème du rang : si $u \in \mathcal{L}(E, F)$ et si $G \oplus \ker(u) = E$ alors u induit un isomorphisme de G dans l'image de u .

Supposons que (r_n) soit une suite convenable. Montrons par récurrence que pour tout n , $r_n = q_n$.

- Initialisation : $q_0 = r_0 = 1$ montre le résultat au rang 0.
- Hérédité : supposons le résultat vrai au rang $n-1 \geq 0$. On a alors $Qq_n = q_{n-1} = r_{n-1} = Qr_n$ et donc $q_n - r_n \in \ker(Q) = \mathbb{K}$. $q_n - r_n$ est une constante mais comme $q_n(0) = r_n(0)$, cette constante est nulle.

On a montré que

Il existe une unique suite (q_n) convenable

Il est tout à fait possible de raisonner par analyse et synthèse pour obtenir l'existence et l'unicité. Il est même sans doute possible de mener un raisonnement par équivalence. Si vous choisissez cette option, il faut cependant être très attentif à l'argumentation et à la structure logique. Nous déconseillons ce type de raisonnement par équivalence.

26. Prouvons par récurrence sur n que

$$\forall x, y \in \mathbb{K}, q_n(x+y) = \sum_{k=0}^n q_k(x)q_{n-k}(y)$$

- Initialisation : le résultat au rang 0 se lit $q_0(x+y) = q_0(x)q_0(y)$ et est vérifié car $q_0 = 1$.
- Hérédité : supposons le résultat vrai au rang n . Pour tous $x, y \in \mathbb{K}$, on a donc

$$Qq_{n+1}(x+y) = \sum_{k=0}^n q_{n-k}(y)Qq_{k+1}(x)$$

On fixe y et on obtient une identité entre polynôme que l'on peut écrire

$$E_y(Qq_{n+1}) = \sum_{k=0}^n q_{n-k}(y)Qq_{k+1}$$

Q étant shift-invariant le membre de gauche est aussi égal à $Q(q_{n+1}(X+y))$ et ainsi

$$q_{n+1}(X+y) - \sum_{k=0}^n q_{n-k}(y)q_{k+1}(X) \in \ker(Q) = \mathbb{K}$$

Cette différence est constante et la valeur en 0 est $q_{n+1}(y)$. Ainsi

$$q_{n+1}(X+y) = q_{n+1}(y) + \sum_{k=0}^n q_{n-k}(y)q_{k+1}(X) = \sum_{j=0}^{n+1} q_{n+1-j}(y)q_j(X)$$

On obtient ainsi le résultat au rang $n+1$.

La suite (q_n) étant définie de manière récurrente et non explicite, le recours à un raisonnement par récurrence pour obtenir des renseignements sur les q_n (ici est ailleurs) doit venir immédiatement à l'esprit.

27. Supposons que Q soit un endomorphisme δ dont (q_n) est la suite associée. On a alors

$$Qq_0 = 0 \text{ et } \forall n \in \mathbb{N}^*, Qq_n = q_{n-1}$$

par échelonnement, la famille $(q_n)_{n \in \mathbb{N}}$ est une base de $\mathbb{K}[X]$ et ces relations caractérisent Q qui est donc unique sous réserve qu'il existe.

Réciproquement, on considère l'unique endomorphisme Q agissant comme ci-dessus sur la base $(q_n)_{n \in \mathbb{N}}$.

Les relations vérifiées par les q_n donnent, pour $n=0$, que $q_0 = \alpha \in \mathbb{K}^*$ et $\alpha = \alpha^2$ ce qui entraîne $q_0 = \alpha = 1$.

On a par hypothèse $\deg(q_n) = n$ et par construction $Qq_n = q_{n-1}$ pour $n \in \mathbb{N}^*$.

On a la relation polynomiale

$$\forall n \in \mathbb{N}^*, q_n(X) = \sum_{k=0}^n q_{n-k}(0)q_k(X)$$

et par identification (les q_k formant une famille libre), $q_n(0) = 0$ (identification des coordonnées de q_0). Les quatre relations que doit vérifier la suite (q_n) pour être associée à Q sont donc obtenues. Il reste à voir que Q est un endomorphisme delta.

Par construction, Q envoie un polynôme de degré n sur un polynôme de degré $n-1$ et

$QX \in \mathbb{K}^*$. Enfin, pour tout y , et tout $n \geq 0$,

$$\begin{aligned}
 E_y \circ Q(q_{n+1}) &= E_y(q_n) \\
 &= q_n(X + y) \\
 &= \sum_{k=0}^n q_{n-k}(y)q_k(X) \\
 &= \sum_{k=0}^n q_{n-k}(y)Qq_{k+1}(X) \\
 &= Q\left(\sum_{k=0}^n q_{n-k}(y)q_{k+1}\right) \\
 &= Q\left(\sum_{j=1}^n q_{n+1-j}(y)q_j\right) \\
 &= Q(q_{n+1}(X + y) - q_{n+1}(y)q_0) \\
 &= Q \circ E_y(q_{n+1}) - q_{n+1}(y)Qq_0 \\
 &= Q \circ E_y(q_{n+1}) \quad \text{car } Qq_0 = 0
 \end{aligned}$$

L'égalité $E_y \circ (q_0) = Q \circ e_y(q_0)$ est aussi vraie ($0 = 0$) et Q est shift-invariant. Elle est finalement delta. On a montré que

Q est l'unique endomorphisme delta auquel (q_n) est associée

Obtenir le seul endomorphisme Q possible (le début du raisonnement) est aisé. Il faut, avant de se lancer, bien comprendre ce qui est attendu. On ne peut, ainsi, se contenter de vérifier les quatre propriétés et la vérification du caractère delta est indispensable. Vous serez récompensés si vous ne faites qu'une partie du travail et surtout si vous indiquez au correcteur que vous comprenez qu'il vous manque un morceau (par exemple si vous n'arrivez pas à montrer le caractère delta. Il est sans doute intéressant, dans la réciproque, de commencer par lister les points qui sont à prouver (puis de prouver ceux que l'on réussit à aborder).

28. $(q_0, \dots, q_n)$ est une famille échelonnée en degré de 0 à n et donc

$(q_0, \dots, q_n)$ est une base de $\mathbb{K}_n[X]$

29. q_0 est envoyé sur 0 et q_k sur q_{k-1} . Ainsi

$$\text{Mat}(Q_n, (q_0, \dots, q_n)) = \begin{pmatrix} 0 & 1 & 0 & \dots & 0 \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & 0 \\ \vdots & & \ddots & \ddots & 1 \\ 0 & \dots & \dots & \dots & 0 \end{pmatrix}, \text{Tr}(Q_n) = 0, \det(Q_n) = 0, \chi_{Q_n} = X^n$$

30. Notons tout d'abord que D est un endomorphisme delta et qu'il lui est bien associée une suite.

posons $q_n = \frac{X^n}{n!}$. On a $q_0 = 1$, $\deg(q_n) = n$, $q_n(0) = 0$ et $Dq_n = q_{n-1}$ quand $n \geq 1$.

La suite associée à D est $\left(\frac{X^n}{n!}\right)_{n \geq 0}$

31. $Q = E_1 - I$ est shift-invariant et $QX = (X + 1) - X = 1 \in \mathbb{K}^*$. Q est donc un endomorphisme delta.

On a (avec les notations de la question), $q_0 = 1$, $\deg(q_n) = n$ et si $n \in \mathbb{N}^*$, $q_n(0) = 1$ et

$$Qq_n = q_n(X + 1) - q_n(X) = \frac{X \dots (X - n + 2)}{n!} ((X + 1) - (X - n + 1)) = q_{n-1}$$

La suite associée à $E_1 - I$ est $\left(\frac{X(X-1)\dots(X-n+1)}{n!} \right)_{n \geq 0}$

32. On a vu que $Q(\mathbb{K}_n[X]) = Q(\mathbb{K}_{n-1}[X])$. Appliquer Q à un polynôme non nul diminue donc le degré de 1. Une récurrence montre donc que

$$\forall k \in \llbracket 0, n \rrbracket, Q^k(\mathbb{K}_n[X]) = \mathbb{K}_{n-k}[X]$$

On en déduit que si $n = \deg(p)$ (avec $p \neq 0$), alors $Q^n(p)$ est une constante et $Q^{n+1}(p) = 0$ puis $Q^k(p) = 0$ pour $k \geq n + 1$. Ceci reste vrai si $p = 0$ (on a alors $n = -1$). Ainsi

$$\forall p \in \mathbb{K}[X], \sum_{k=0}^{\infty} (Q^k p)(0) q_k \in \mathbb{K}[X]$$

puisque la somme est en fait finie et constituée d'éléments de $\mathbb{K}[X]$.

Comme en question 9, $Q_1 : p \mapsto \sum_{k=0}^{\infty} (Q^k p)(0) q_k$ est un endomorphisme. Par définition $Qq_n = q_{n-1}$ pour $n \geq 1$ et (récurrence aisée)

$$\forall k \in \llbracket 0, n \rrbracket, Q^k q_n = q_{n-k}$$

Comme $Q^k q_n$ si $k \geq n + 1$ (début de la question), on a

$$Q_1 q_n = \sum_{k=0}^n (Q^k q_n)(0) q_k = \sum_{k=0}^n q_{n-k}(0) q_k = q_n$$

Q_1 est un endomorphisme qui agit comme I sur les q_n qui forment une base de $\mathbb{K}[X]$ et donc $Q_1 = I$:

$$\forall p \in \mathbb{K}[X], p = \sum_{k=0}^{\infty} (Q^k p)(0) q_k$$

Cette sous-partie est annoncée comme une généralisation de la formule de Taylor prouvée plus tôt et il s'agit donc juste d'adapter la preuve faite alors. C'est le cas aussi pour la question qui suit.

33. On reprend la preuve de la question 11 en l'adaptant-généralisant (mais c'est essentiellement la même chose).

Considérons un endomorphisme T et posons

$$T_1 = \sum_{k=0}^{\infty} (Tq_k)(0) Q^k$$

On a en particulier (en appliquant l'égalité en q_n ce qui donne une égalité entre polynôme et en évaluant cette égalité en 0)

$$\forall n \in \mathbb{N}, (T_1 q_n)(0) = (Tq_n)(0)$$

puisque $Q^k(q_n) = 0$ si $k \geq n+1$ et si $k \in \llbracket 0, n \rrbracket$, $Q^k(q_n)(0) = q_{n-k}(0) = \delta_{n,k}$.
Comme T_1 et T sont linéaires et comme $(q_n)_{n \in \mathbb{N}}$ est une base de $\mathbb{K}[X]$, on en déduit par combinaisons linéaires que

$$\forall p \in \mathbb{K}[X], (T_1 p)(0) = (T p)(0)$$

Supposons que T est shift-invariant. Pour tout $a \in \mathbb{K}$, $(T p)(X+a) = T(p(X+a))$ et en appliquant cette égalité avec q_n ;

$$(T q_n)(X+a) = T(q_n(X+a)) = T\left(\sum_{k=0}^n q_k(X) q_{n-k}(y)\right) = \sum_{k=0}^n q_{n-k}(y) (T q_k)$$

Evaluons cette identité polynomiale en 0 :

$$(T q_n)(a) = \sum_{k=0}^n q_{n-k}(y) T q_k(0)$$

Comme T_1 est shift-invariant (comme en question 9), le même calcul avec T_1 donne

$$(T q_n)(a) = (T_1 q_n)(a)$$

Ceci étant vrai pour tout a , les polynôme $T q_n$ et $T_1 q_n$ sont égaux. Par linéarité et comme (q_n) est une base de $\mathbb{K}[X]$, les endomorphismes T et T_1 sont égaux.

$$T = \sum_{k=0}^{\infty} (T q_k)(0) Q^k \text{ si } T \text{ est shift-invariant}$$

34. On choisit $Q = E_1 - I$ qui est bien delta et on a donc $q_n = \frac{X(X-1)\dots(X-n+1)}{n!}$ pour $n \geq 1$.
Pour un tel n ,

$$q'_n = \frac{1}{n!} \sum_{j=0}^{n-1} \prod_{0 \leq k \leq n-1, k \neq j} (X-k)$$

et donc

$$q'_n(0) = \frac{(-1)^{n-1}}{n}$$

Montrons par ailleurs par récurrence que pour tout entier k ,

$$\forall p \in \mathbb{K}[X], Q^k p = \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j)$$

- Initialisation : la formule est vraie pour $k = 0$ (elle se lit $\forall p, p = p$).

- Hérédité : on suppose le résultat vrai à un rang $k \geq 0$. On a alors pour tout p ,

$$\begin{aligned}
 Q^{k+1}p &= Q(Q^k p) \\
 &= \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} (p(X+j+1) - p(X+j)) \\
 &= \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j+1) - \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j) \\
 &= \sum_{j=1}^{k+1} (-1)^{k+1-j} \binom{k}{j-1} p(X+j) - \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j) \\
 &= p(X+k+1) + \sum_{j=1}^k (-1)^{k+1-j} \left(\binom{k}{j-1} + \binom{k}{j} \right) p(X+j) + (-1)^{k+1} p(X) \\
 &= p(X+k+1) + \sum_{j=1}^k (-1)^{k+1-j} \binom{k+1}{j} p(X+j) + (-1)^{k+1} p(X)
 \end{aligned}$$

ce qui donne, en regroupant les termes, le résultat au rang $k+1$.

On utilise la formule de la question 33 avec l'opérateur shift-invariant D . Comme on l'a vu plus haut, on peut dans la somme se limiter à des indices $k \leq \deg(p)$ et ainsi

$$p' = Dp = \sum_{k=0}^{\deg(p)} q'_k(0) Q_k(p) = \sum_{k=0}^{\deg(p)} \left(q'_k(0) \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j) \right)$$

$q'_0(0) = 0$ et en remplaçant $q'_k(0)$ par sa valeur

$$p' = \sum_{k=0}^{\deg(p)} \left(\frac{(-1)^{k-1}}{k} \sum_{j=0}^k (-1)^{k-j} \binom{k}{j} p(X+j) \right)$$

ce qui décrit aussi

$$p' = \sum_{k=0}^{\deg(p)} \left(\frac{1}{k} \sum_{j=0}^k (-1)^{j+1} \binom{k}{j} p(X+j) \right)$$

C'est une question de synthèse qu'il est bien sûr intéressant d'aborder. Cependant, comme l'énoncé donne le résultat le correcteur sera attentif à l'argumentation donnée.

Un peu de calcul ombral

35. Par formule de Leibniz, on a

$$\forall k \geq 1, D^k(Xp) = \sum_{j=0}^k \binom{k}{j} D^j X D^{k-j} p = X D^k p + k D^{k-1} p$$

par ailleurs, si $T = \sum_{k=0}^{\infty} a_k D^k$ (quand on applique avec un polynôme p , la somme est en fait finie)

$$T'(p) = \sum_{k=0}^{\infty} a_k D^k(Xp) - X \sum_{k=0}^{\infty} a_k D^k p = a_0 Xp + \sum_{k=1}^{\infty} a_k (X D^k p + k D^{k-1} p) - \sum_{k=0}^{\infty} a_k X D^k p$$

Il y a simplification et

$$\left(\sum_{k=0}^{\infty} a_k D^k \right)' = \sum_{k=1}^{\infty} k a_k D^{k-1}$$

Dans la formule de Leibniz, on a supposé $k \geq 1$ pour être sûr que la somme contient deux termes. Pour $k = 0$, on peut considérer que le terme est nul (on multiplie par k) mais D^{-1} n'a pas de sens.

36. Si T est shift-invariant, la question 11 montre qu'il a une écriture comme en question précédente. On a ainsi une écriture de T' qui montre avec la question 9 la shift-invariance.

La shift-invariance est conservée par dérivation de Pincherle

37. Si T est shift-invariant, T' l'est aussi (question précédente). De plus, si on suppose que T est delta, il existe (question 20) des scalaires α_k avec $\alpha_1 \neq 0$ tels que

$$T = \sum_{k=1}^{\infty} \alpha_k D^k$$

et on a donc

$$T' = \sum_{k=1}^{\infty} k \alpha_k D^{k-1} = \sum_{k=0}^{\infty} k \alpha_{k+1} D^k$$

$T'1 = \alpha_1 \neq 0$ et par question 18, T' est inversible.

Si T est delta, T' est shift-invariant et inversible

38. Soient S et T deux endomorphismes de $\mathbb{K}[X]$. Pour tout p , on a

$$\begin{aligned} (S' \circ T + S \circ T')(p) &= S'(T(p)) + S(T(Xp) - XT(p)) \\ &= S(XT(p)) - XS(T(p)) + S(T(Xp)) - S(XT(p)) \\ &= S(T(Xp)) - XS(T(p)) \\ &= (S \circ T)'(p) \end{aligned}$$

où on a utilisé la définition de la dérivée de Pincherle et la linéarité de S . Ainsi

$$(S \circ T)' = S' \circ T + S \circ T'$$

On a donc formellement la même formule qu'avec la dérivée usuelle et on peut ainsi espérer généraliser d'autre formule comme la dérivée de T^n (où il s'agit bien sûr d'une itérée pour la composition) ou de T^{-1} quand T est inversible. Gardons cela en tête !

39. Partant de $Q = D \circ U$ (et comme $D' = I$), on a $Q' = U + D \circ U'$. En composant, on a donc

$$Q'^{-n-1} = U^{-n} + D \circ U'^{-n-1}$$

On montre par récurrence que pour tout endomorphisme T shift-invariant,

$$\forall k \in \mathbb{N}, (T^k)'^{k-1} \circ T'$$

- C'est vrai si $k = 0$ car $I' = 0$.
- Si le résultat est vrai au rang k , alors (rappelons que tous les endomorphismes shift-invariants commutent)

$$(T^{k+1})'^k \circ T)^{k})'^k \circ T'^{k-1} \circ T'^k \circ T'^k \circ T'$$

En appliquant ceci avec U^{-1} , on a $(U^{-n})'^{-n-1} \circ U' = -nU'^{-n-1}$ et donc

$$Q'^{-n-1} = U^{-n} - \frac{1}{n}D \circ (U^{-n})'$$

Or, $(D \circ U^{-n})'^{-n} + D \circ (U^{-n})'$ et ainsi

$$Q'^{-n-1} = U^{-n} - \frac{1}{n}((D \circ U^{-n})'^{-n}) = \frac{n+1}{n}U^{-n} - \frac{1}{n}(D \circ U^{-n})'$$

et en particulier

$$Q'^{-n-1}(X^n) = \frac{n+1}{n}U^{-n}(X^n) - \frac{1}{n}(D \circ U^{-n})'^n$$

Mais par définition et par commutation

$$\begin{aligned} (D \circ U^{-n})'^n &= (D \circ U^{-n})(X^{n+1}) - X(D \circ U^{-n})(X^n) \\ &= (n+1)U^{-n}(X^n) - nXU^{-n}(X^{n-1}) \end{aligned}$$

Il reste à injecter plus haut pour obtenir

$$\boxed{Q'^{-n-1}(X^n) = nXU^{-n}(X^{n-1})}$$

Cette question est sans doute la plus délicate du sujet. Il faut trouver les bonnes quantités auxquelles appliquer les bonnes formules. Vous avez peut-être fait des tentatives donnant des choses s'approchant du résultat et, par exemple, faisant apparaître U^{-n} lors du calcul de $(Q'^{-n-1})(X^n)$. Il peut être intéressant de proposer une de ces tentatives en commentant rapidement.

40. D'après la question précédente, il suffit de montrer que

$$\forall n \in \mathbb{N}^*, Q' \left(U^{-n-1} \left(\frac{X^n}{n!} \right) \right) = q_n$$

Notons r_n le polynôme du membre de gauche et convenons que $r_0 = 1$. On a $\deg(r_n) = n$ (question 18) et pour $n \geq 1$, $r_n(0) = 0$ (question 39) et (on rappelle que tous les endomorphismes shift-invariants commutent)

$$\begin{aligned} Qr_n &= Q' \left(Q \circ U^{-n-1} \left(\frac{X^n}{n!} \right) \right) \\ &= Q' \left(D \circ U^{-n} \left(\frac{X^n}{n!} \right) \right) \\ &= Q' \left(U^{-n} \left(\frac{X^{n-1}}{(n-1)!} \right) \right) \end{aligned}$$

On obtient $Qr_n = r_{n-1}$ si $n \geq 2$ et on vérifie que l'égalité reste vraie pour $n = 1$. L'unicité de la question 25 donne l'égalité voulue et ainsi

$$\boxed{\forall n \in \mathbb{N}^*, n!q_n = XU^{-n}(X^{n-1})}$$

Pour vérifier une expression de q_n , il suffit comme en questions 30 ou 31 de s'assurer que les quatre propriétés caractéristiques de la suite (q_n) ont bien lieu.

Ceci et la question 39 donne

$$(Q'^{-n-1})(X^n) = n!q_n$$

et en composant par $(Q'^{-1}$,

$$U^{-n-1}(X^n) = n!(Q'^{-1}(q_n))$$

On vérifie que cette égalité est encore vraie pour $n = 0$: de $Q = U \circ D$, on trouve $Q' = U + U' \circ D$ et donc $Q'(1) = U(1)$ puis $Q'^{-1}(1) = U^{-1} \circ Q'(1) = 1$.

Pour $n \geq 1$, on peut utiliser cette égalité avec $n - 1$:

$$U^{-n}(X^{n-1}) = (n-1)!(Q'^{-1}(q_{n-1}))$$

On multiplie par X et on utilise la relation prouvée ci-dessus

$$X(Q'^{-1}(q_{n-1})) = \frac{1}{(n-1)!} XU^{-n}(X^{n-1}) = \frac{n!q_n}{(n-1)!} = nq_n$$

$$\boxed{\forall n \in \mathbb{N}^*, nq_n = X(Q'^{-1}(q_{n-1}))}$$

41. Dans le cas $Q = L$, on a vu en question 21 que $U = (D - I)^{-1}$ et donc $(D - I)L = D$. On en déduit que

$$(D - I)\ell_{n-1} = (D - I)(L\ell_n) = D\ell_n$$

c'est à dire

$$\boxed{\ell'_n = \ell'_{n-1} - \ell_{n-1}}$$

Remarquons que, par le calcul fait en question 39, $U'^{-1})'^{-2}D'^{-2}$. Ainsi

$$L' = U + U'^{-2}(D - I - D) = -(D - I)^{-2}$$

et $(L'^{-1} = -(D - I)^2$. La question 40 donne ainsi

$$nq_n = -X(D - I)^{-2}q_{n-1} = -X(D - I)(q'_{n-1} - q_{n-1}) = -X(D - I)(q'_n) = -X(q''_n - q'_n)$$

On a montré que

$$\boxed{X\ell''_n - X\ell'_n + n\ell_n = 0}$$

On montre par récurrence que

$$\boxed{\ell_n = \sum_{k=1}^n (-1)^k \binom{n-1}{k-1} \frac{X^k}{k!}}$$

- On a $\ell_0 = 1$ et donc $\ell'_1 = -1$. Comme $\ell_1(0) = 0$, on a donc $\ell_1 = -X$ ce qui correspond à la formule au rang 1.

- Supposons la formule vraie à un rang $n \geq 1$. On a alors

$$\begin{aligned}
 \ell'_{n+1} &= \sum_{k=1}^n (-1)^k \binom{n-1}{k-1} \frac{X^{k-1}}{(k-1)!} - \sum_{k=1}^n (-1)^k \binom{n-1}{k-1} \frac{X^k}{k!} \\
 &= \sum_{k=0}^{n-1} (-1)^{k+1} \binom{n-1}{k} \frac{X^k}{k!} - \sum_{k=1}^n (-1)^k \binom{n-1}{k-1} \frac{X^k}{k!} \\
 &= -1 + \sum_{k=1}^{n-1} (-1)^{k+1} \left(\binom{n-1}{k} + \binom{n-1}{k-1} \right) \frac{X^k}{k!} + (-1)^{n+1} \frac{X^n}{n!} \\
 &= \sum_{k=0}^n (-1)^{k+1} \binom{n}{k} \frac{X^k}{k!}
 \end{aligned}$$

On primitive en tenant compte de $\ell_{n+1}(0) = 0$:

$$\ell_{n+1} = \sum_{k=0}^n (-1)^{k+1} \binom{n}{k} \frac{X^{k+1}}{(k+1)!}$$

et un changement d'indice donne la relation au rang $n+1$.

42. $(q_n)_{n \geq 0}$ étant une base de $\mathbb{K}[X]$, il existe un unique endomorphisme T de $\mathbb{K}[X]$ tel que pour tout n , $Tq_n = \frac{X^n}{n!}$. T est linéaire et transforme une base en une base : c'est un isomorphisme.

Il existe un unique endomorphisme inversible T tel que $\forall n$, $Tq_n = \frac{X^n}{n!}$

43. Pour montrer que deux endomorphismes commutent, il suffit de montrer qu'ils coïncident sur une base. Or, on a

$$\forall n \geq 1, T \circ Q \circ T^{-1} \left(\frac{X^n}{n!} \right) = T \circ Q(q_n) = T(q_{n-1}) = \frac{X^{n-1}}{(n-1)!} = D \frac{X^n}{n!}$$

et $T \circ Q \circ T^{-1}(1) = T \circ Q(1) = T(0) = 0 = D(1)$. On a donc

$$D = T \circ Q \circ T^{-1}$$

Il faut retenir de cette égalité, que " Q et D , c'est la même chose à isomorphisme près". C'est un peu comme si on opérât un "changement de base".

44. W est immédiatement linéaire. Elle est bijective de bijection réciproque $p \mapsto p(X/\alpha)$. Ainsi,

W est un automorphisme de $\mathbb{K}[X]$

45. Remarquons que $D \circ W^{-1}(p)$ est le polynôme dérivé de $W^{-1}(p)$ c'est à dire de $P(X/\alpha)$. Il vaut $\frac{1}{\alpha}$ fois le polynôme dérivé de P en X/α c'est à dire $\frac{1}{\alpha} W^{-1}(D(p))$. On a montré que

$$D \circ W^{-1} = \frac{1}{\alpha} W^{-1} \circ D$$

D'après la question 21, $L = (D - I)^{-1} \circ D = D \circ (D - I)^{-1}$ (ici et ailleurs, deux endomorphismes shift-invariants commutent). Ainsi,

$$P = W \circ (D - I)^{-1} \circ D \circ W^{-1} = \frac{1}{\alpha} W \circ (D - I)^{-1} \circ W^{-1} \circ D$$

J'écris alors que

$$(D - I) \circ W^{-1} = D \circ W^{-1} - W^{-1} = \frac{1}{\alpha} W^{-1} \circ D - W^{-1} = W^{-1} \left(\frac{D}{\alpha} - I \right)$$

En passant à l'inverse,

$$W \circ (D - I)^{-1} = \left(\frac{D}{\alpha} - I \right)^{-1} \circ W$$

En injectant dans l'expression de P :

$$P = \frac{1}{\alpha} \left(\frac{D}{\alpha} - I \right)^{-1} \circ D$$

et avec la commutation notée plus haut

$$P = \frac{1}{\alpha} D \circ \left(\frac{D}{\alpha} - I \right)^{-1}$$

46. Avec cette expression, P est shift-invariant comme composée d'endomorphismes shift-invariants.

$\left(\frac{D}{\alpha} - I \right)^{-1}$ étant inversible et shift-invariant, il envoie X sur un polynôme de degré 1. L'image par D de ce polynôme est donc dans $\mathbb{K}^*$ et $PX \in \mathbb{K}^*$. Ainsi,

P est un endomorphisme delta.

Posons $p_n = \ell_n(\alpha X) = W(\ell_n)$. $p_0 = 1$, $\deg(p_n) = \deg(\ell_n) = n$, $p_n(0) = \ell_n(0) = 0$ quand $n \geq 1$ et

$$\forall n \in \mathbb{N}^*, Pp_n = W \circ L \circ W^{-1}(W(\ell_n)) = W(L\ell_n) = W\ell_{n-1} = p_{n-1}$$

Ainsi,

La suite de polynômes associée à P est $(p_n) = (W(\ell_n))$

47. Comme $L = D \circ (D - I)^{-1}$, on a aussi $L \circ (D - I) = D$ i.e. $L \circ D - L = D$ ou encore $(L - I) \circ D = D$. Comme L est un endomorphisme delta, 0 est sa seule valeur propre et $L - I$ est inversible. Ainsi (des endomorphismes shift-invariants commutent)

$$D = (L - I)^{-1} \circ D = D \circ (L - I)^{-1}$$

Avec la question 45,

$$P = \frac{1}{\alpha} L \circ (L - I)^{-1} \circ \left(\frac{D}{\alpha} - I \right)^{-1} = \frac{1}{\alpha} L \circ \left(\left(\frac{D}{\alpha} - I \right) \circ (L - I) \right)^{-1}$$

Or,

$$\left(\frac{D}{\alpha} - I \right) \circ (L - I) = \frac{1}{\alpha} (D \circ (L \circ I)) - (L - I) = \frac{L}{\alpha} - (L - I)$$

Avec tout ceci et la linéarité de L ($\alpha L \circ Q = L \circ (\alpha Q)$)

$$P = L \circ (L - \alpha(L - I))^{-1}$$

ou encore

$$P = L \circ ((1 - \alpha)L + \alpha I)^{-1}$$

48. Avec la partie IV.D, on a $D = T \circ L \circ T^{-1}$ (en utilisant l'endomorphisme delta égal à L).
On a alors

$$\begin{aligned}
 Q &= T \circ L \circ ((1 - \alpha)L + \alpha I)^{-1} \circ T^{-1} \\
 &= T \circ L \circ T^{-1} \circ T \circ ((1 - \alpha)L + \alpha I)^{-1} \circ T^{-1} \\
 &= D \circ (T \circ ((1 - \alpha)L + \alpha I) \circ T^{-1})^{-1} \\
 &= D \circ ((1 - \alpha)T \circ L \circ T^{-1} + \alpha I)^{-1}
 \end{aligned}$$

Finalement,

$$\boxed{Q = D \circ ((1 - \alpha)D + \alpha I)^{-1}}$$

Q est un endomorphisme shift-invariant comme composée de tels endomorphismes. On conclut comme en question 46 que $\boxed{Q \text{ est un endomorphisme delta.}}$

On utilise la notation r_n de l'énoncé. Pour $n \geq 1$, on a

$$\begin{aligned}
 (\alpha I + (1 - \alpha)D)r_{n-1} &= \sum_{k=1}^{n-1} \binom{n-2}{k-1} \alpha^k (1 - \alpha)^{n-1-k} \left(\alpha \frac{X^k}{k!} + (1 - \alpha) \frac{X^{k-1}}{(k-1)!} \right) \\
 &= \sum_{k=1}^{n-1} \binom{n-2}{k-1} \alpha^{k+1} (1 - \alpha)^{n-1-k} \frac{X^k}{k!} \\
 &\quad + \sum_{k=1}^{n-1} \binom{n-2}{k-1} \alpha^k (1 - \alpha)^{n-k} \frac{X^{k-1}}{(k-1)!} \\
 &= \sum_{k=1}^{n-1} \binom{n-2}{k-1} \alpha^{k+1} (1 - \alpha)^{n-1-k} \frac{X^k}{k!} \\
 &\quad + \sum_{k=0}^{n-2} \binom{n-2}{k} \alpha^{k+1} (1 - \alpha)^{n-1-k} \frac{X^k}{k!} \\
 &= \sum_{k=1}^{n-2} \left(\binom{n-2}{k-1} + \binom{n-2}{k} \right) \alpha^{k+1} (1 - \alpha)^{n-1-k} \frac{X^k}{k!} \\
 &\quad + \alpha (1 - \alpha)^{n-1} + \alpha^n \frac{X^{n-1}}{(n-1)!} \\
 &= \sum_{k=0}^{n-1} \binom{n-1}{k} \alpha^{k+1} (1 - \alpha)^{n-1-k} \frac{X^k}{k!} \\
 &= Dr_n
 \end{aligned}$$

On en déduit (toujours avec la commutation) que

$$D \circ ((1 - \alpha)D + \alpha I)^{-1} r_n = r_{n-1}$$

C'est à dire $Qr_n = r_{n-1}$. Comme $\deg(r_n) = n$ et $r_n(0) = 0$ quand $n \geq 1$, on a (avec $r_0 = 1$)

$$\boxed{\text{La suite de polynômes associée à } Q \text{ est } (r_n)}$$

49. On a

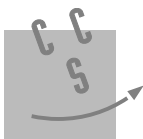
$$r_{n-1} = Qr_n = T \circ P \circ T^{-1} r_n$$

et donc $(T^{-1}r_n)$ est la suite de polynômes associée à P . Par unicité, on a

$$\ell_n(\alpha X) = T^{-1}r_n = \sum_{k=1}^n \binom{n-1}{k-1} \alpha^k (1-\alpha)^{n-k} T^{-1} \frac{X^k}{k!}$$

c'est à dire

$$\ell_n(\alpha X) = \sum_{k=1}^n \binom{n-1}{k-1} \alpha^k (1-\alpha)^{n-k} \ell_k$$



Mathématiques 2

MP, MPI

2023

CONCOURS CENTRALE-SUPÉLEC

4 heures

Calculatrice autorisée

Notations

- Dans tout le sujet, n désigne un entier naturel non nul.
- Étant donnés deux entiers naturels a et b , on note $\llbracket a, b \rrbracket$ l'ensemble des entiers naturels k tels que $a \leq k \leq b$.
- Pour deux suites de nombres réels $(u_m)_{m \in \mathbb{N}}$ et $(v_m)_{m \in \mathbb{N}}$, la notation $u_m = O(v_m)$ signifie qu'il existe une suite bornée $(M_m)_{m \in \mathbb{N}}$ telle que l'on ait

$$\exists m_0 \in \mathbb{N} \quad | \quad \forall m \geq m_0, \quad u_m = M_m v_m.$$

- On pourra utiliser sans démonstration la formule suivante, qui précise la formule de Stirling lorsque n tend vers $+\infty$:

$$n! = \left(\frac{n}{e}\right)^n \sqrt{2\pi n} \left(1 + O\left(\frac{1}{n}\right)\right).$$

Toutes les variables aléatoires considérées sont discrètes.

I Résultats préliminaires

I.A – Calcul d'une intégrale classique

Rappelons que n désigne un entier naturel non nul. On note

$$I_n = \int_0^1 \frac{1}{(1+t^2)^n} dt \quad \text{et} \quad K_n = \int_0^{+\infty} \frac{1}{(1+t^2)^n} dt.$$

I.A.1)

Q 1. Montrer que

$$I_n \geq \frac{1}{2^n}.$$

Q 2. Justifier l'existence de K_n et donner la valeur exacte de K_1 .

Q 3. Montrer que

$$\int_1^{+\infty} \frac{1}{(1+t^2)^n} dt = O\left(\frac{1}{n2^n}\right).$$

On pourra minorer $1+t^2$ par un polynôme de degré 1.

Q 4. En déduire que, lorsque n tend vers $+\infty$,

$$I_n \sim K_n.$$

Q 5. Établir la relation de récurrence $K_n = K_{n+1} + \frac{1}{2n} K_n$.

Q 6. En déduire un équivalent simple de I_n lorsque n tend vers $+\infty$.

I.A.2)

Q 7. Justifier que

$$\sqrt{n} I_n = \int_0^{\sqrt{n}} \frac{1}{(1+u^2/n)^n} du.$$

Q 8. Montrer que

$$\lim_{n \rightarrow \infty} \sqrt{n} I_n = \int_0^{+\infty} e^{-u^2} du.$$

Q 9. En déduire les valeurs de

$$\int_0^{+\infty} e^{-u^2} du \quad \text{puis de} \quad \int_{-\infty}^{+\infty} e^{-u^2/2} du.$$

Dans toute la suite, on posera pour tout x réel

$$\varphi(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad \text{et} \quad \Phi(x) = \int_{-\infty}^x \varphi(t) dt.$$

I.B – Comportement asymptotique de $1 - \Phi$ Soit $x > 0$.Q 10. En écrivant que $\varphi(t) \leq \frac{t}{x} \varphi(t)$ pour tout $t \geq x$, montrer que

$$\int_x^{+\infty} \varphi(t) dt \leq \frac{\varphi(x)}{x}.$$

Q 11. À l'aide de l'étude d'une fonction bien choisie, montrer que

$$\frac{x}{x^2 + 1} \varphi(x) \leq \int_x^{+\infty} \varphi(t) dt.$$

Q 12. En déduire un équivalent simple de $1 - \Phi(x)$ lorsque x tend vers $+\infty$.

I.C – Une inégalité maximale

Dans cette sous-partie, n est un entier naturel non nul et $Z_1, \dots, Z_n$ sont des variables aléatoires discrètes indépendantes sur un espace probabilisé $(\Omega, \mathcal{A}, \mathbb{P})$.

Pour tout $p \in \llbracket 1, n \rrbracket$, on note $R_p = \sum_{i=1}^p Z_i$.

On va montrer la propriété

$$\forall x > 0, \quad \mathbb{P}(\{\max_{1 \leq p \leq n} |R_p| \geq 3x\}) \leq 3 \max_{1 \leq p \leq n} \mathbb{P}(\{|R_p| \geq x\}).$$

On admet que les différentes fonctions intervenant dans cette inégalité sont bien des variables aléatoires discrètes.

Pour simplifier, notons A l'événement $\{\max_{1 \leq p \leq n} |R_p| \geq 3x\}$. Ainsi,

$$A = \{\omega \in \Omega \mid \max_{1 \leq p \leq n} |R_p(\omega)| \geq 3x\}.$$

Dans le cas où $n \geq 2$, définissons de plus les événements

$$A_1 = \{|R_1| \geq 3x\} \quad \text{et} \quad A_p = \left\{ \max_{1 \leq i \leq p-1} |R_i| < 3x \right\} \cap \{|R_p| \geq 3x\}$$

pour $p \in \llbracket 2, n \rrbracket$.

Q 13. Exprimer l'événement A à l'aide des événements $A_1, A_2, \dots, A_n$.

Q 14. Montrer que l'on a

$$\mathbb{P}(A) \leq \mathbb{P}(\{|R_n| \geq x\}) + \sum_{p=1}^n \mathbb{P}(A_p \cap \{|R_n| < x\}).$$

Q 15. Justifier que pour tout $p \in \llbracket 1, n \rrbracket$, on a l'inclusion

$$A_p \cap \{|R_n| < x\} \subset A_p \cap \{|R_n - R_p| > 2x\}.$$

Q 16. En déduire que

$$\mathbb{P}(A) \leq \mathbb{P}(\{|R_n| \geq x\}) + \max_{1 \leq p \leq n} \mathbb{P}(\{|R_n - R_p| > 2x\}).$$

Q 17. Conclure.

II Étude d'une suite de fonctions

Pour tout $n \in \mathbb{N}^*$ et tout $k \in \llbracket 0, n \rrbracket$, on pose

$$x_{n,k} = -\sqrt{n} + \frac{2k}{\sqrt{n}}.$$

De plus, on définit la fonction $B_n : \mathbb{R} \rightarrow \mathbb{R}$ par les conditions

$$\begin{aligned} \forall x \in \left] -\infty, -\sqrt{n} - \frac{1}{\sqrt{n}} \right[, \quad & B_n(x) = 0 \\ \forall k \in \llbracket 0, n \rrbracket, \forall x \in \left[x_{n,k} - \frac{1}{\sqrt{n}}, x_{n,k} + \frac{1}{\sqrt{n}} \right[, \quad & B_n(x) = \frac{\sqrt{n}}{2} \binom{n}{k} \frac{1}{2^n} \\ \forall x \in \left[\sqrt{n} + \frac{1}{\sqrt{n}}, +\infty \right[, \quad & B_n(x) = 0 \end{aligned}$$

L'objectif de cette partie est de montrer que la suite de fonctions $(B_n)_{n \in \mathbb{N}^*}$ converge uniformément sur $\mathbb{R}$ vers la fonction φ , définie dans la partie I. Autrement dit, on souhaite montrer

$$\lim_{n \rightarrow +\infty} \Delta_n = 0 \quad \text{avec} \quad \Delta_n = \sup_{x \in \mathbb{R}} |B_n(x) - \varphi(x)|.$$

L'usage d'une figure pour appréhender la problématique de cette partie sera vivement apprécié.

II.A –

Q 18. Comparer les réels $-x_{n,k}$ et $x_{n,n-k}$.

Q 19. Justifier l'existence du réel Δ_n pour tout $n \in \mathbb{N}^*$.

Q 20. Montrer que, pour tout $n \in \mathbb{N}^*$, on a l'égalité

$$\Delta_n = \sup_{x \geq 0} |B_n(x) - \varphi(x)|.$$

Q 21. Pour tout $n \in \mathbb{N}^*$, montrer que B_n est une application décroissante sur $\mathbb{R}^+$.

On pourra distinguer selon que n est pair ou impair.

Dans la suite de cette partie, on fixe $\varepsilon > 0$. La limite $\lim_{x \rightarrow +\infty} \varphi(x) = 0$ assure de l'existence d'un nombre $\ell \in \mathbb{R}^+$ tel que $\varphi(\ell) \leq \frac{\varepsilon}{2}$.

II.B – Dans cette sous-partie, on va montrer

$$\lim_{n \rightarrow +\infty} \sup_{x \in [0, \ell]} |B_n(x) - \varphi(x)| = 0.$$

On introduit pour cela l'ensemble

$$I_n = \{k \in \llbracket 0, n \rrbracket \mid x_{n,k} \in [0, \ell + 1]\}$$

dont on peut vérifier que c'est un intervalle d'entiers.

Dans la suite de cette sous-partie, on suppose que n et k varient de sorte que $k \in I_n$.

Q 22. Montrer que l'on a

$$k!(n-k)! = 2\pi e^{-n} k^{k+1/2} (n-k)^{n-k+1/2} \left(1 + O\left(\frac{1}{n}\right)\right)$$

pour n tendant vers l'infini.

On pourra utiliser la formule de Stirling rappelée en début d'énoncé.

Q 23. En déduire que, pour n tendant vers $+\infty$, on a

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi}} \frac{1 + O\left(\frac{1}{n}\right)}{\left(\frac{2k}{n}\right)^{k+1/2} \left(2 - \frac{2k}{n}\right)^{n-k+1/2}}.$$

Q 24. En déduire que

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi}} \frac{1 + O\left(\frac{1}{n}\right)}{\left(1 - \frac{x_{n,k}^2}{n}\right)^{\frac{n+1}{2}} \left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{x_{n,k}}{2}\sqrt{n}} \left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{-\frac{x_{n,k}}{2}\sqrt{n}}}$$

puis que

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right).$$

Q 25. Montrer qu'il existe un entier naturel n_1 tel que, pour tout entier $n \geq n_1$,

$$\sup_{x \in [0, \ell]} |B_n(x) - \varphi(x)| \leq \frac{\varepsilon}{2}.$$

II.C –

Q 26. Pour tout $\ell > 0$, montrer qu'il existe un entier naturel n_2 , tel que, pour tout $n \geq n_2$,

$$B_n(\ell) \leq 2\varphi(\ell).$$

Q 27. Conclure que la suite $(\Delta_n)_{n \in \mathbb{N}^*}$ converge vers 0.

III Applications

Soit $(\Omega, \mathcal{A}, \mathbb{P})$ un espace probabilisé et X une variable aléatoire discrète sur $(\Omega, \mathcal{A}, \mathbb{P})$ telle que $\mathbb{P}(X = -1) = 1/2$ et $\mathbb{P}(X = 1) = 1/2$. On considère une suite $(X_i)_{i \in \mathbb{N}^*}$ de variables aléatoires discrètes sur $(\Omega, \mathcal{A}, \mathbb{P})$, mutuellement indépendantes et de même loi que X . On définit alors

$$S_0 = 0 \quad \text{et} \quad \forall n \in \mathbb{N}^*, \quad S_n = \sum_{i=1}^n X_i.$$

On dit que $(S_n)_{n \in \mathbb{N}}$ est une marche aléatoire symétrique sur $\mathbb{Z}$. On admettra que pour tout $n \geq 1$, S_n est une variable aléatoire discrète sur $(\Omega, \mathcal{A}, \mathbb{P})$.

III.A – Théorème central limite

Soit I un intervalle de $\mathbb{R}$ et $(f_n)_{n \in \mathbb{N}^*}$ une suite de fonctions continues par morceaux sur I qui converge uniformément sur I vers une fonction f également continue par morceaux sur I .

Q 28. Si $(u_n)_{n \in \mathbb{N}^*}$ (respectivement $(v_n)_{n \in \mathbb{N}^*}$) est une suite de nombres réels appartenant à I qui converge vers $u \in I$ (respectivement $v \in I$), montrer que

$$\lim_{n \rightarrow +\infty} \left(\int_{u_n}^{v_n} f_n(x) \, dx \right) = \int_u^v f(x) \, dx.$$

On pose, pour tout $i \in \mathbb{N}^*$, $Y_i = \frac{X_i+1}{2}$ et $T_n = \sum_{i=1}^n Y_i$.

Q 29. Montrer que, pour tout $j \in \llbracket 0, n \rrbracket$,

$$\mathbb{P}(\{T_n = j\}) = \int_{x_{n,j}-1/\sqrt{n}}^{x_{n,j}+1/\sqrt{n}} B_n(x) \, dx,$$

où $x_{n,j}$ a été défini dans la partie II.

Considérons un couple (u, v) de réels tel que $u < v$, et notons

$$J_n = \left\{ j \in \llbracket 0, n \rrbracket \mid \frac{n + u\sqrt{n}}{2} \leq j \leq \frac{n + v\sqrt{n}}{2} \right\}.$$

Q 30. Justifier que

$$\mathbb{P} \left(\left\{ u \leq \frac{S_n}{\sqrt{n}} \leq v \right\} \right) = \sum_{j \in J_n} \mathbb{P}(\{T_n = j\}).$$

Q 31. En déduire que l'on a

$$\lim_{n \rightarrow +\infty} \mathbb{P} \left(\left\{ u \leq \frac{S_n}{\sqrt{n}} \leq v \right\} \right) = \int_u^v \varphi(x) \, dx$$

puis que

$$\lim_{n \rightarrow +\infty} \mathbb{P} \left(\left\{ u \leq \frac{S_n}{\sqrt{n}} \right\} \right) = 1 - \Phi(u)$$

où les applications φ et Φ ont été définies dans la partie I.

III.B – Critère de tension

Dans cette dernière sous-partie, on fixe $\varepsilon \in]0, 1[$.

Q 32. Montrer qu'il existe $x_0 \geq 1$ tel que l'on ait

$$\forall x \geq x_0, \quad \exists n_x \in \mathbb{N}, \quad \forall n \geq n_x, \quad x^2 \mathbb{P}(\{|S_n| \geq x\sqrt{n}\}) \leq \varepsilon.$$

Q 33. Pour x_0 et x comme à la question précédente, on fixe $N \geq \frac{n_x}{\varepsilon}$ et on choisit $n \geq N$. Montrer qu'alors

$$x^2 \mathbb{P}(\{\max_{1 \leq p \leq n} |S_p| \geq 3x\sqrt{n}\}) \leq 3\varepsilon.$$

• • • FIN • • •

SOLUTION

Résultats préliminaires

I.A - Calcul d'une intégrale classique

1. Par décroissance sur $[0, 1]$ de $t \mapsto \frac{1}{(1+t^2)^n}$, on a

$$\forall t \in [0, 1], \frac{1}{2^n} \leq \frac{1}{(1+t^2)^n}$$

On intègre entre 0 et 1 (bornes dans le bon sens) pour conclure que

$$\boxed{\frac{1}{2^n} \leq I_n}$$

2. $f_n : t \mapsto \frac{1}{(1+t^2)^n}$ est continue sur $\mathbb{R}^+$. En $+\infty$, elle équivaut à $\frac{1}{t^{2n}}$ qui est intégrable au voisinage de $+\infty$ car $2n > 1$. La fonction est intégrable sur $\mathbb{R}^+$ et $\boxed{K_n \text{ existe pour } n \geq 1}$. Une primitive de f_n sur $\mathbb{R}^+$ est arctan qui est nulle en 0 et de limite $\pi/2$ en $+\infty$. Ainsi, $\boxed{K_1 = \pi/2}$.

En début de problème, il ne faut vraiment pas oublier les arguments même s'ils vous semblent évidents. Ici (et en question suivante), on doit justifier l'existence d'une intégrale. Il faut commencer par voir où sont les problèmes : les infinis et les points d'éventuelle discontinuité. Notez que si on suit la lettre du programme, les soucis ne devraient intervenir qu'aux bornes de l'intégrale. Mieux vaut cependant bien vérifier qu'il n'y a pas un problème "caché". Ici, il n'y a aucun souci avec la borne 0 et il serait mal venu de faire une étude locale en 0. Il faut aussi, avec les arguments donnés, conclure en termes d'intégrabilité (ou donner explicitement un argument de signe constant).

3. Quand $t \geq 1$, $1+t^2 \geq 1+t \geq 0$ et donc $(1+t^2)^n \geq (1+t)^n$. Quand $n \geq 2$, le minorant est intégrable sur $[1, +\infty[$ (continu et équivalent à $1/t^n$ au voisinage de $+\infty$). Ainsi

$$\forall n \geq 2, 0 \leq \int_1^\infty \frac{dt}{(1+t^2)^n} \leq \int_1^{+\infty} \frac{dt}{(1+t)^n} = \left[-\frac{1}{(n-1)(1+t)^{n-1}} \right]_1^\infty = \frac{1}{(n-1)2^{n-1}}$$

On a donc

$$\forall n \geq 2, 0 \leq n2^n \int_1^\infty \frac{dt}{(1+t^2)^n} \leq 2 \frac{n}{n-1}$$

Le majorant est borné et ainsi

$$\boxed{\int_1^\infty \frac{dt}{(1+t^2)^n} = O\left(\frac{1}{n2^n}\right)}$$

Attention, un simple majoration de l'intégrale ne suffit pas. Il faut un encadrement ou, mieux, une majoration en valeur absolue.

4. On a donc

$$K_n = I_n + \int_1^\infty \frac{dt}{(1+t^2)^n} = I_n + O\left(\frac{1}{n2^n}\right)$$

Avec la question 1, $\frac{1}{n2^n} = o(I_n)$ et donc $K_n = I_n + o(I_n)$ c'est à dire

$$K_n \sim I_n$$

Rappelons qu'il y a essentiellement deux façons de prouver que $u_n \sim v_n$. On peut (si c'est licite) former le quotient et montrer qu'il est de limite 1. Ou former $u_n - v_n$ et montrer que ce terme est négligeable devant u_n (ou devant v_n). Il faut se demander quelle est la plus simple entre l'expression avec une différence et celle avec un quotient.

5. Soit $n \geq 1$. On forme

$$K_n - K_{n+1} = \int_0^\infty \frac{t^2}{(1+t^2)^{n+1}} dt$$

Une primitive de $\frac{t}{(1+t^2)^{n+1}}$ est $-\frac{1}{2n(1+t^2)^n}$. On dérive t en 1. Comme $-\frac{t}{2n(1+t^2)^n}$ est de limite finie (nulle) en $+\infty$ et que les fonctions sont de classe C^1 sur $\mathbb{R}^+$, une intégration par parties est licite et donne

$$K_n - K_{n+1} = \left[-\frac{t}{2n(1+t^2)^n} \right]_0^\infty + \int_0^\infty \frac{1}{2n(1+t^2)^n} dt$$

Le crochet est nul et on reconnaît le second terme qui vaut $\frac{1}{2n} K_n$. Ains

$$K_n = K_{n+1} + \frac{1}{2n} K_n$$

Cette relation de récurrence évoque bien sûr celle vérifiée par les intégrales de Wallis. Rappelons que si $W_n = \int_0^{\pi/2} \cos(x)^n dx$, on a $W_{n+2} = \frac{n+1}{n+2} W_n$. Ici, le changement de variable $t = \tan(x)$, montre que $K_n = W_{2n-2}$ et c'est donc bien la même relation. Rappelons aussi qu'une manière classique de prouver l'équivalent de Stirling est de faire l'étude de la suite (W_n) . Ici, l'énoncé fait le chemin inverse.

6. La relation précédente s'écrit aussi $K_{n+1} = \frac{2n-1}{2n} K_n$. Montrons par récurrence que

$$K_n = \frac{(2(n-1))!}{(2^{n-1}(n-1)!)^2} \frac{\pi}{2}$$

- Initialisation : le résultat est vrai au rang 1 puisque $K_1 = \frac{\pi}{2}$.
- Hérédité : supposons le résultat vrai jusqu'à un rang $n \geq 1$. On a alors

$$K_{n+1} = \frac{2n-1}{2n} \frac{(2(n-1))!}{(2^{n-1}(n-1)!)^2} \frac{\pi}{2}$$

En écrivant que $\frac{2n-1}{2n} = \frac{2n(2n-1)}{(2n)^2}$, on obtient le résultat au rang n .

L'énoncé ne donnant pas le résultat, il est envisageable de procéder en itérant la relation sous la forme

$$K_n = \frac{2n-3}{2n-2} \frac{2n-5}{2n-4} \cdots \frac{1}{2} K_1$$

On multiplie ensuite haut et bas par $(2n-2)(2n-4) \dots 2 = 2^{n-1}(n-1)!$ et on retrouve l'expression ci-dessus. Ici, on évite les points de suspension avec la récurrence.

Pour plus de commodité, je remarque que la relation de récurrence donne $K_{n+1} \sim K_n$ et je cherche un équivalent de K_{n+1} . Avec l'équivalent de Stirling,

$$K_{n+1} \sim \frac{\sqrt{4\pi n} \left(\frac{2n}{e}\right)^{2n}}{4^n 2\pi n \left(\frac{n}{e}\right)^{2n}} \frac{\pi}{2} = \frac{\sqrt{\pi}}{2\sqrt{n}}$$

On en déduit que $K_n \sim \frac{\sqrt{\pi}}{2\sqrt{n}}$ et avec la question 4,

$$I_n \sim \frac{\sqrt{\pi}}{2\sqrt{n}}$$

7. Dans I_n , on effectue le changement de variable linéaire $u = \sqrt{n}t$:

$$I_n = \frac{1}{\sqrt{n}} \int_0^{\sqrt{n}} \frac{1}{(1 + u^2/n)^n} du$$

ce que l'on peut écrire

$$\sqrt{n}I_n = \int_0^{\sqrt{n}} \frac{1}{(1 + u^2/n)^n} du$$

Ici, on fait un changement de variable dans une intégrale de fonction continue sur un segment et le programme ne demande pas que l'on vérifie les hypothèses de régularité. Même dans le cas d'une intégrale généralisée, les changements usuels peuvent être menés sans justification.

8. Notons g_n la fonction définie par

$$\forall u \in [0, \sqrt{n}], g_n(u) = \frac{1}{(1 + u^2/n)^n} \text{ et } \forall u > \sqrt{n}, g_n(u) = 0$$

en sorte que

$$\sqrt{n}I_n = \int_0^{\infty} g_n$$

On va utiliser le théorème de convergence dominée pour étudier cette suite.

- (g_n) est une suite de fonctions continues. Si $u \geq 0$ est fixé, pour n grand ($n \geq \lceil u^2 \rceil$ par exemple), $g_n(u) = \frac{1}{(1+u^2/n)^n} \rightarrow e^{-u^2}$ (car $(1 + u^2/n)^n = \exp(n \ln(1 + u^2/n))$ et le terme dans l'exponentielle est de limite u^2 et l'exponentielle est continue).
- La limite simple de (g_n) sur $\mathbb{R}^+$, la fonction $u \mapsto e^{-u^2}$, est continue sur $\mathbb{R}^+$.
- Soit $u \geq 0$. En utilisant la formule du binôme et en ne gardant que deux termes (les autres sont positifs), on a

$$(1 + u^2/n)^n \geq 1 + \binom{n}{1} \frac{u^2}{n} = 1 + u^2$$

On en déduit que $\forall u \in [0, \sqrt{n}], 0 \leq g_n(u) \leq \frac{1}{1+u^2}$, majoration qui reste vraie au delà de $\sqrt{n}$. Le majorant est intégrable sur $\mathbb{R}^+$ (voir question 2).

On peut ainsi appliquer le théorème et affirmer que

$$\lim_{n \rightarrow +\infty} \sqrt{n} I_n = \int_0^{+\infty} e^{-u^2} du$$

$\sqrt{n} I_n$ est le terme général d'une suite définie par une intégrale. L'utilisation d'un théorème d'interversion limite-intégrale est donc assez naturelle. Il y a cependant un problème ici car l'indice n apparaît simultanément dans les bornes et dans l'intégrande. Pour se ramener au cas classique d'interversion, il faut donc se débarrasser du n présent dans les bornes. Le changement de variable naturel $t = u/\sqrt{n}$ est à exclure puisqu'il nous ramènerait à l'expression initiale. La tactique utilisée consistant à forcer les bornes en multipliant par la fonction indicatrice de $[0, \sqrt{n}]$ est à connaître.

Pour l'étape de domination dans le théorème, nous avons utilisé une petite astuce. On aurait aussi pu étudier $h : t \mapsto t \ln(1 + u^2/t)$. Cette fonction est deux fois dérivable sur $[1, +\infty[$ et on peut vérifier que sa dérivée seconde est négative et donc sa dérivée décroissante. En $+\infty$, cette dérivée est de limite nulle. On a donc h' qui est positive sur $[1, +\infty[$ et h qui croît sur cet ensemble. On retrouve alors la fonction dominatrice $\frac{1}{1+u^2}$.

9. L'équivalent de la question 6 donne aussi $\sqrt{n} I_n \rightarrow \sqrt{\pi}/2$ et par unicité de la limite

$$\int_0^{+\infty} e^{-u^2} du = \frac{\sqrt{\pi}}{2}$$

Dans l'intégrale ci-dessus, on effectue le changement de variable linéaire $v = \sqrt{2}u$:

$$\int_0^{+\infty} e^{-u^2} du = \frac{1}{\sqrt{2}} \int_0^{+\infty} e^{-v^2/2} dv$$

On a donc $\int_0^{+\infty} e^{-u^2/2} du = \sqrt{\pi}/2$. Par parité de la fonction intégrée, l'intégrale est double sur $\mathbb{R}$ et donc

$$\int_{-\infty}^{+\infty} e^{-u^2/2} du = \sqrt{2\pi}$$

I.B - Comportement asymptotique de $1 - \Phi$

10. Si $t \geq x$ alors $\frac{t}{x} \geq 1$ et l'inégalité ne change pas en multipliant par $\varphi(t) \geq 0$. Comme on travaille avec des quantités positives, on peut alors écrire dans $\mathbb{R} \cup \{+\infty\}$ (en intégrant, les bornes sont dans le bon sens)

$$\int_x^{+\infty} \varphi(t) dt \leq \frac{1}{x} \int_x^{+\infty} t \varphi(t) dt$$

La dérivée de φ est $t \mapsto -t\varphi(t)$. L'intégrale du membre de droite vaut donc $[-\varphi]_x^{+\infty} = \varphi(x)$ et est donc fini. Cela indique que les intégrales existent. De plus, on trouve que

$$\int_x^{+\infty} \varphi(t) dt \leq \frac{\varphi(x)}{x}$$

Rappelons qu'il faut a priori toujours justifier l'existence d'une quantité avant de l'écrire (qu'il s'agisse d'une limite, d'une intégrale généralisée, d'une somme infinie). Le programme nous permet cependant dans le cas d'une fonction f positive sur I ou d'une famille dénombrable $(u_k)_{k \in A}$ d'écrire $\int_I f$ ou $\sum_{k \in A} u_k$ dans $\mathbb{R} \cup \{+\infty\}$ et de justifier les existences en majorant par une quantité finie.

11. Posons $\psi(x) = (x^2 + 1) \int_x^{+\infty} \varphi - x\varphi(x)$, qui est bien définie sur $\mathbb{R}^{+*}$ (et même sur $\mathbb{R}$) d'après la question précédente (où on a justifié l'existence d'intégrale).

Comme φ est continue et d'intégrale convergente en $+\infty$, une généralisation du théorème fondamental indique que $x \mapsto \int_x^{+\infty} \varphi$ est l'opposée d'une primitive de φ . ψ est donc dérivable et

$$\psi'(x) = -(x^2 + 1)\varphi(x) + 2x \int_x^{+\infty} \varphi - \varphi(x) - x\varphi'(x)$$

D'une part, $x\varphi'(x) = -x^2\varphi(x)$ et d'autre part on a l'inégalité $x \int_x^{+\infty} \varphi \leq \varphi(x)$ (pour $x > 0$). On en déduit que

$$\psi'(x) \leq 0$$

ψ est décroissante et on a donc (la limite existe par théorème de limite monotone)

$$\forall x > 0, \psi(x) \geq \lim_{t \rightarrow +\infty} \psi(t)$$

Par croissances comparées, $x\varphi(x)$ est de limite nulle en $+\infty$. Avec la question précédente,

$$\forall x > 0, 0 \leq (x^2 + 1) \int_x^{+\infty} \varphi \leq \frac{x^2 + 1}{x} \varphi(x) \underset{x \rightarrow +\infty}{\sim} x\varphi(x)$$

est aussi de limite nulle en $+\infty$. ψ est donc négative sur $\mathbb{R}^{+*}$ et ainsi

$$\boxed{\frac{x}{x^2 + 1} \varphi(x) \leq \int_x^{+\infty} \varphi}$$

On a utilisé ici une version du théorème fondamental avec une borne généralisée. C'est licite dès que l'on sait que l'intégrale existe (et que la fonction est continue bien sûr).

12. On remarque tout d'abord que (question 9)

$$\int_{-\infty}^{+\infty} \varphi = 1$$

On a donc

$$1 - \Phi(x) = \int_{-\infty}^{+\infty} \varphi - \int_{-\infty}^x \varphi = \int_x^{+\infty} \varphi$$

Les questions 10 et 11 montrent ainsi que

$$\frac{x}{x^2 + 1} \varphi(x) \leq 1 - \Phi(x) \leq \frac{\varphi(x)}{x}$$

Majorant et minorant équivalent en $+\infty$ à $\frac{\varphi(x)}{x}$ et ainsi (en divisant par $\varphi(x)/x$ et par théorème d'encadrement)

$$\boxed{1 - \Phi(x) \underset{x \rightarrow +\infty}{\sim} \frac{\varphi(x)}{x} = \frac{1}{\sqrt{2\pi x}} e^{-x^2/2}}$$

I.C - Une inégalité maximale

13. Si $\omega \in A$ alors l'un des $|R_k(\omega)|$ pour $k \in \llbracket 1, n \rrbracket$ est $\geq 3x$ (le maximum est atteint). On peut alors considérer le plus petit des tels k et on a $\omega \in A_k$.
Réciproquement, si ω est dans A_k pour $k \in \llbracket 1, n \rrbracket$, alors $|R_k(\omega)| \geq 3x$ et donc ω est dans A (le maximum des $|R_i(\omega)|$ est plus grand que $3x$).
On a donc

$$A = \bigcup_{k=1}^n A_k$$

Un événement est un ensemble (une partie de Ω). Une égalité entre événements doit donc être traitée comme une égalité entre ensembles !

14. En conditionnant selon le système complet $((|R_n| < x), (|R_n| \geq x))$, on a

$$\mathbb{P}(A) = \mathbb{P}(A \cap (|R_n| \geq x)) + \mathbb{P}(A \cap (|R_n| < x))$$

Par croissance de $\mathbb{P}$, $\mathbb{P}(A \cap (|R_n| \geq x)) \leq \mathbb{P}(|R_n| \geq x)$.

Par distributivité (et la question 13)

$$A \cap (|R_n| < x) = \bigcup_{p=1}^n (A_p \cap (|R_n| < x))$$

Les A_p étant disjoints, on a ainsi

$$\mathbb{P}(A \cap (|R_n| < x)) = \sum_{p=1}^n \mathbb{P}(A_p \cap (|R_n| < x))$$

On en déduit finalement que

$$\mathbb{P}(A) \leq \mathbb{P}(|R_n| \geq x) + \sum_{p=1}^n \mathbb{P}(A_p \cap (|R_n| < x))$$

Ne vous contentez pas d'aligner des égalités ou inégalités séparées par des "donc". Prenez la peine à chaque étape de préciser l'argument clé (caractère disjoint, monotonie, distributivité) même d'un simple mot.

15. Soit $\omega \in A_p \cap (|R_n| < x)$. On a donc

$$\forall i \in \llbracket 1, p-1 \rrbracket, |R_i(\omega)| < 3x, |R_p(\omega)| \geq 3x \text{ et } |R_n(\omega)| < x$$

En particulier la seconde forme de l'inégalité triangulaire donne

$$|R_n(\omega) - R_p(\omega)| \geq |R_p(\omega)| - |R_n(\omega)| > 3x - x = 2x$$

et $\omega \in (|R_n - R_p| > 2x)$. Comme $\omega \in A_p$, il est dans l'intersection et

$$A_p \cap (|R_n| < x) \subset A_p \cap (|R_n - R_p| > 2x)$$

Soyez toujours attentif lors de l'utilisation de l'inégalité triangulaire sous une forme ou l'autre. C'est une source importante d'erreur si on veut aller trop vite.

16. En passant aux probabilités (opération croissante) et en sommant dans les inclusions de la question 15, on trouve que

$$\sum_{p=1}^n \mathbb{P}(A_p \cap (|R_n| < x)) \leq \sum_{p=1}^n \mathbb{P}(A_p \cap (|R_n - R_p| > 2x))$$

A_p est un événement qui ne dépend que des variables $Z_1, \dots, Z_p$ et la variable $R_n - R_p$ ne dépend que de $Z_{p+1}, \dots, Z_n$. Par indépendance des Z_i et lemme des coalitions, les événements A_p et $(|R_n - R_p| > 2x)$ sont indépendants et

$$\mathbb{P}(A_p \cap (|R_n - R_p| > 2x)) = \mathbb{P}(A_p) \mathbb{P}(|R_n - R_p| > 2x)$$

On a ainsi

$$\sum_{p=1}^n \mathbb{P}(A_p \cap (|R_n| < x)) \leq \sum_{p=1}^n \mathbb{P}(A_p) \mathbb{P}(|R_n - R_p| > 2x)$$

et avec la question 14,

$$\mathbb{P}(A) - \mathbb{P}(|R_n| \geq x) \leq \sum_{p=1}^n \mathbb{P}(A_p) \mathbb{P}(|R_n - R_p| > 2x)$$

En majorant $\mathbb{P}(|R_n - R_p| > 2x)$ par le maximum sur p (multiplier par $\mathbb{P}(A_p) \geq 0$ ne change pas le sens des inégalités), on obtient

$$\mathbb{P}(A) - \mathbb{P}(|R_n| \geq x) \leq \max_{1 \leq p \leq n} \mathbb{P}(|R_n - R_p| > 2x) \sum_{p=1}^n \mathbb{P}(A_p)$$

Les A_p étant disjoints, la somme de droite est plus petite que 1 et ainsi

$$\boxed{\mathbb{P}(A) \leq \mathbb{P}(|R_n| \geq x) + \max_{1 \leq p \leq n} \mathbb{P}(|R_n - R_p| > 2x)}$$

Il faut bien comprendre l'utilisation du lemme des coalitions qui est faite ici. On retrouve ce type d'argument dans les sujets sur les marches aléatoire. Reportez vous, par exemple, au très bon sujet des Mines de 2020 (épreuve 2).

17. Si $|a| \leq x$ et $|b| \leq x$ alors $|a - b| \leq 2x$. En contraposant, si $|a - b| > 2x$ alors $|a| > x$ ou $|b| > x$. Ainsi

$$(|R_n - R_p| > 2x) \subset (|R_n| > x) \cup (|R_p| > x)$$

Par inégalité de Boole

$$\mathbb{P}(|R_n - R_p| > 2x) \leq \mathbb{P}(|R_n| > x) + \mathbb{P}(|R_p| > x)$$

et a fortiori

$$\mathbb{P}(|R_n - R_p| > 2x) \leq 2 \max_{1 \leq k \leq n} \mathbb{P}(|R_k| > x) \leq 2 \max_{1 \leq k \leq n} \mathbb{P}(|R_k| \geq x)$$

Le membre de droite ne dépend pas de p et donc

$$\max_{1 \leq p \leq n} \mathbb{P}(|R_n - R_p| > 2x) \leq 2 \max_{1 \leq k \leq n} \mathbb{P}(|R_k| \geq x)$$

Comme $\mathbb{P}(|R_n| \geq x) \leq \max_{1 \leq k \leq n} \mathbb{P}(|R_k| \geq x)$, la question 16 donne

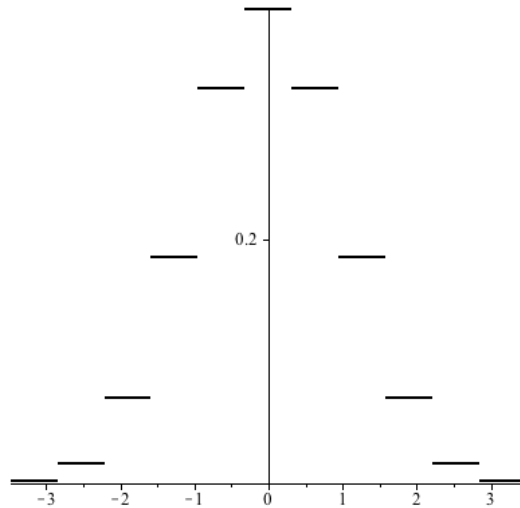
$$\mathbb{P}(A) \leq 3 \max_{1 \leq k \leq n} \mathbb{P}(|R_k| \geq x)$$

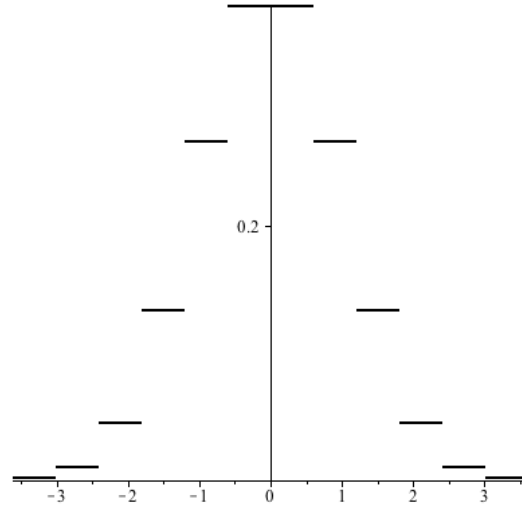
c'est à dire

$$\mathbb{P}\left(\max_{1 \leq p \leq n} |R_p| \geq 3x\right) \leq 3 \max_{1 \leq k \leq n} \mathbb{P}(|R_k| \geq x)$$

Etude d'une suite de fonctions

Comme proposé, on s'intéresse au graphe de B_n . On constate d'une part une petite différence autour de l'abscisse 0 entre les cas n pair (à gauche) et n impair (à droite). Les $x_{n,k}$ sont les abscisses des milieux des segments (attention, deux segments réunis autour de 0 dans le cas n impair).





Il y a à l'évidence une symétrie dans le dessin (pas une parité car les intervalles définissant B_n sont semi-ouverts).

18. Un simple calcul donne

$$x_{n,n-k} = -\sqrt{n} + \frac{2(n-k)}{\sqrt{n}} = \sqrt{n} - \frac{2k}{\sqrt{n}} = -\left(-\sqrt{n} + \frac{2k}{\sqrt{n}}\right) = -x_{n,k}$$

$$\boxed{x_{n,n-k} = -x_{n,k}}$$

19. B_n est une fonction qui ne prend qu'un nombre fini de valeurs (puisque'elle est constante par morceaux et qu'il n'y a qu'un nombre fini de morceaux). C'est donc une fonction bornée sur $\mathbb{R}$.

Par positivité et monotonie de l'exponentielle, $\forall x, 0 \leq \varphi(x) \leq \varphi(0)$ et φ est bornée sur $\mathbb{R}$.

Finalement $|B_n - \varphi| \leq |B_n| + |\varphi|$ est elle aussi bornée sur $\mathbb{R}$ et $\boxed{\Delta_n \text{ existe}}$.

20. Le but est de montrer que toutes les valeurs prises par $|B_n - \varphi|$ sur $\mathbb{R}^-$ le sont aussi sur $\mathbb{R}^+$. B_n n'est que "presque" paire comme indiqué en commentaire du dessin plus haut alors que φ , elle, est paire. Comme on l'a constaté sur le dessin, le problème vient des bornes des intervalles (l'une fermée et l'autre ouverte). Il va donc y avoir "naturellement" une disjonction de cas, le cas le plus embêtant étant celui d'une borne de l'un des intervalles.

Pour $k \in \llbracket 0, n \rrbracket$, je note $I_{n,k} = \left] x_{n,k} - \frac{1}{\sqrt{n}}, x_{n,k} + \frac{1}{\sqrt{n}} \right]$. D'après la question 18,

$$I_{n,n-k} = \left] -x_{n,k} - \frac{1}{\sqrt{n}}, -x_{n,k} + \frac{1}{\sqrt{n}} \right]$$

et donc, $y \in I_{n,n-k} \implies -y \in I_{n,k}$.

Par ailleurs, l'identité $\binom{n}{k} = \binom{n}{n-k}$ indique que B_n prend les mêmes valeurs sur $I_{n,k}$ et $I_{n,n-k}$.

Soit $y \in \mathbb{R}$. Distinguons les cas.

- Si y est dans l'un des $I_{n,k}$ alors avec ce qui précède et la parité de φ , $|B_n - \varphi|$ prend les mêmes valeurs en y et $-y$ et par définition de la borne supérieure on a donc (comme y ou $-y$ est positif) $|B_n(y) - \varphi(y)| \leq \Delta'_n = \sup_{x \in \mathbb{R}^+} |B_n(x) - \varphi(x)|$.
- Si $y < x_{n,0} - \frac{1}{\sqrt{n}}$, $|B_n - \varphi|$ prend encore la même valeur ($\varphi(y)$) en y et $-y$ et on a encore $|B_n(y) - \varphi(y)| \leq \Delta'_n$.
- Si $y \geq x_{n,n} + \frac{1}{\sqrt{n}}$ alors $y \geq 0$ et $|B_n(y) - \varphi(y)| \leq \Delta'_n$.
- Sinon, il existe k tel que $y = x_{n,k} - \frac{1}{\sqrt{n}}$ avec $k < n$. Il existe une suite $(y_p)_{p \in \mathbb{N}}$ d'éléments de $I_{n,k}$ qui converge vers y . Avec le premier cas, $|B_n(y_p) - \varphi(y_p)| \leq \Delta'_n$. Comme B_n est continue à droite en tout point et φ est continue, on obtient $|B_n(y) - \varphi(y)| \leq \Delta'_n$ en faisant tendre p vers l'infini.

La majoration obtenue étant vraie pour tout y , $\Delta_n \leq \Delta'_n$. L'inégalité inverse est immédiate (la borne supérieure sur une partie plus grande est plus grande), on a

$$\Delta_n = \sup_{x \geq 0} |B_n(x) - \varphi(x)|$$

21. Supposons n pair et notons le $n = 2p$. On a alors $x_{2p,p} = 0$. Sur $\mathbb{R}^+$, B_{2p} prend alors successivement les valeurs $\frac{\sqrt{2p}}{2^{2p+1}} \binom{2p}{k} = \alpha_k$ pour $k = p, p+1, \dots, 2p$ puis la valeur 0. Il s'agit donc de montrer que

$$\alpha_p \geq \alpha_{p+1} \geq \dots \geq \alpha_{2p} \geq 0$$

La dernière inégalité est immédiate et il reste à prouver que

$$\forall k \in \llbracket p, 2p-1 \rrbracket, \binom{2p}{k} \geq \binom{2p}{k+1}$$

On forme donc

$$\binom{2p}{k} - \binom{2p}{k+1} = \frac{(2p)!}{(k+1)!(2p-k)!} (2k-2p)$$

quantité qui est bien positive pour $k \in \llbracket p, 2p-1 \rrbracket$.

On peut, bien sûr, tout aussi bien former le quotient et montrer qu'il est ≤ 1 . C'est même sans doute plus naturel étant donnée la forme multiplicative des coefficients du binôme.

Si n est impair, on le note $2p+1$. Cette fois $x_{2p+1,p} = -\frac{1}{\sqrt{2p+1}}$ et B_{2p+1} prend alors successivement les valeurs $\frac{\sqrt{2p+1}}{2^{2p+2}} \binom{2p+1}{k} = \alpha_k$ pour $k = p+1, \dots, 2p+1$ puis la valeur 0. Comme ci-dessus, il s'agit de montrer que

$$\forall k \in \llbracket p+1, 2p \rrbracket, \binom{2p+1}{k} \geq \binom{2p+1}{k+1}$$

ce qui se fait de manière similaire.

$$B_n \text{ est décroissante sur } \mathbb{R}^+$$

Il n'est sans doute pas nécessaire de répéter le calcul ici et l'argument de similarité me semble suffisant.

22. Il nous est dit que “ n et k varient de sorte que $k \in I_n$ ”. Dans cette question et les suivantes, il semble que la variable utilisée soit n et k n’est pas une constante mais “dépend de n ”. Il s’agit donc a priori de vérifier une propriété valable pour toute suite (k_n) telle que $k_n \in I_n$. Comme le “grand O ” porte sur la variable n et que l’on veut ultimement une convergence uniforme, il faut que nos majorations soient uniformes vis à vis du choix de la suite (k_n) .

La condition $k \in I_n$ s’écrit

$$\frac{n}{2} \leq k \leq \frac{n}{2} + \frac{\ell+1}{2}\sqrt{n}$$

En particulier, I_n est non vide dès que n est assez grand (peut-être même pour tout $n \geq 1$ mais on ne s’intéresse qu’au comportement asymptotique).

De plus $0 \leq \frac{1}{k} \leq \frac{2}{n}$ et on pourra remplacer $O\left(\frac{1}{k}\right)$ par $O\left(\frac{1}{n}\right)$ de manière “uniforme” (la constante de majoration ne dépend pas de $k \in I_n$).

De même, on a un encadrement de $n - k$ qui donne $O\left(\frac{1}{n-k}\right) = O\left(\frac{1}{n}\right)$.

Compte-tenu de ces remarques, la formule de Stirling donne

$$(n-k)! = \frac{(n-k)^{n-k}}{e^{n-k}} \sqrt{2\pi(n-k)} \left(1 + O\left(\frac{1}{n}\right)\right)$$

$$k! = \frac{k^k}{e^k} \sqrt{2\pi k} \left(1 + O\left(\frac{1}{n}\right)\right)$$

Comme $\left(1 + O\left(\frac{1}{n}\right)\right)^2 = 1 + O\left(\frac{1}{n}\right)$, on a donc

$$k!(n-k)! = 2\pi e^{-n} k^{k+1/2} (n-k)^{n-k+1/2} \left(1 + O\left(\frac{1}{n}\right)\right)$$

23. On a $\left(1 + O\left(\frac{1}{n}\right)\right)^{-1} = 1 + O\left(\frac{1}{n}\right)$ et donc

$$\frac{1}{k!(n-k)!} = \frac{1}{2\pi e^{-n} k^{k+1/2} (n-k)^{n-k+1/2}} \left(1 + O\left(\frac{1}{n}\right)\right)$$

On utilise la formule de Stirling pour exprimer $n!$ et on multiplie. Comme $\left(1 + O\left(\frac{1}{n}\right)\right)^2 = 1 + O\left(\frac{1}{n}\right)$, on obtient

$$\begin{aligned} \binom{n}{k} &= \frac{n^{n+1/2}}{\sqrt{2\pi k^{k+1/2} (n-k)^{n-k+1/2}}} \left(1 + O\left(\frac{1}{n}\right)\right) \\ &= \frac{1}{\sqrt{n}} \frac{1}{\sqrt{2\pi} (k/n)^{k+1/2} (1-k/n)^{n-k+1/2}} \left(1 + O\left(\frac{1}{n}\right)\right) \end{aligned}$$

En multipliant par $\frac{\sqrt{n}}{2^{n+1}}$, on trouve $B_n(x_{n,k})$. On répartit le terme 2^{n+1} et on obtient

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi} \left(\frac{2k}{n}\right)^{k+1/2} \left(2 - \frac{2k}{n}\right)^{n-k+1/2}} \left(1 + O\left(\frac{1}{n}\right)\right)$$

24. On a $\frac{2k}{n} = 1 + \frac{x_{n,k}}{\sqrt{n}}$ et $2 - \frac{2k}{n} = 1 - \frac{x_{n,k}}{\sqrt{n}}$ et aussi $k = \frac{\sqrt{n}x_{n,k} + n}{2}$. Ainsi

$$\begin{aligned} \frac{1}{\left(\frac{2k}{n}\right)^{k+1/2} \left(2 - \frac{2k}{n}\right)^{n-k+1/2}} &= \frac{1}{\left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{\sqrt{n}x_{n,k} + n + 1}{2}} \left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{n - \sqrt{n}x_{n,k} + 1}{2}}} \\ &= \frac{1}{\left(1 - \frac{x_{n,k}^2}{n}\right)^{\frac{n+1}{2}} \left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{\sqrt{n}x_{n,k}}{2}} \left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{-\sqrt{n}x_{n,k}}{2}}} \end{aligned}$$

et avec la question précédente,

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi}} \frac{1 + O\left(\frac{1}{n}\right)}{\left(1 - \frac{x_{n,k}^2}{n}\right)^{\frac{n+1}{2}} \left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{\sqrt{n}x_{n,k}}{2}} \left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{-\sqrt{n}x_{n,k}}{2}}}$$

Remarquons que puisque $k \in I_n$, $x_{n,k} \in [0, \ell]$ est borné (indépendamment de n et de $k \in I_n$). Ainsi (on utilise $\ln(1+u) = u + O(u^2)$ et $O(x_{n,k}^2/n) = O(1/n)$)

$$\begin{aligned} \left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{\sqrt{n}x_{n,k}}{2}} &= \exp\left(\frac{\sqrt{n}x_{n,k}}{2} \ln\left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)\right) \\ &= \exp\left(\frac{\sqrt{n}x_{n,k}}{2} \left(\frac{x_{n,k}}{\sqrt{n}} + O\left(\frac{1}{n}\right)\right)\right) \\ &= \exp\left(\frac{x_{n,k}^2}{2}\right) \exp\left(O\left(\frac{1}{\sqrt{n}}\right)\right) \\ &= \exp\left(\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right) \end{aligned}$$

et de même

$$\left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{\frac{-\sqrt{n}x_{n,k}}{2}} = \exp\left(\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$$

Le dernier terme donne

$$\begin{aligned} \left(1 - \frac{x_{n,k}^2}{n}\right)^{\frac{n+1}{2}} &= \exp\left(\frac{n+1}{2} \ln\left(1 - \frac{x_{n,k}^2}{n}\right)\right) \\ &= \exp\left(\frac{n+1}{2} \left(-\frac{x_{n,k}^2}{n} + O\left(\frac{1}{n^2}\right)\right)\right) \\ &= \exp\left(-\frac{x_{n,k}^2}{2} - \frac{x_{n,k}^2}{2n} + O\left(\frac{1}{n}\right)\right) \\ &= \exp\left(-\frac{x_{n,k}^2}{2} + O\left(\frac{1}{n}\right)\right) \\ &= \exp\left(-\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{n}\right)\right) \end{aligned}$$

Comme $\left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)^2 \left(1 + O\left(\frac{1}{n}\right)\right) = \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$ et que l'inverse de ce terme est aussi $\left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$, on a donc

$$\frac{1}{\left(1 - \frac{x_{n,k}^2}{n}\right)^{\frac{n+1}{2}} \left(1 + \frac{x_{n,k}}{\sqrt{n}}\right)^{\sqrt{n}\frac{x_{n,k}}{2}} \left(1 - \frac{x_{n,k}}{\sqrt{n}}\right)^{-\sqrt{n}\frac{x_{n,k}}{2}}} = \exp\left(-\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$$

et on conclut que

$$B_n(x_{n,k}) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{x_{n,k}^2}{2}\right) \left(1 + O\left(\frac{1}{\sqrt{n}}\right)\right)$$

Dans tous ces calculs, il faut prendre garde à l'ordre que l'on peut atteindre et bien préciser les développements usuels que l'on utilise. Le résultat étant fourni par l'énoncé, le correcteur va guetter les petites tricheries et ne les pardonnera pas.

25. φ est continue sur $[0, \ell]$ et y est donc uniformément continue par théorème de Heine. Ainsi, il existe $r > 0$ tel que

$$\forall x, y \in [0, \ell], |x - y| \leq r \implies |\varphi(x) - \varphi(y)| \leq \frac{\varepsilon}{4}$$

On peut ensuite choisir un rang n_0 tel que $\frac{1}{\sqrt{n_0}} \leq r$ (puisque $(1/n)$ est de limite nulle).

Comme $(x_{n,k})$ est bornée, $\left(\exp\left(-\frac{x_{n,k}^2}{2}\right)\right)$ l'est aussi et la question précédente donne

$$B_n(x_{n,k}) - \varphi(x_{n,k}) = O\left(\frac{1}{\sqrt{n}}\right)$$

Il existe donc une constante M et un rang n'_0 tel que

$$\forall n \geq n'_0, \forall k \in I_n, |B_n(x_{n,k}) - \varphi(x_{n,k})| \leq \frac{M}{\sqrt{n}}$$

Le majorant est de limite nulle et quitte à augmenter n'_0 , on peut supposer que

$$\forall n \geq n'_0, \forall k \in I_n, |B_n(x_{n,k}) - \varphi(x_{n,k})| \leq \frac{\varepsilon}{4}$$

On choisit alors $n_1 = \max(n_0, n'_0)$ et on se donne $n \geq n_1$.

Il est important de pouvoir définir un rang n_1 qui ne dépend que de ℓ . Il faut maintenant majorer une borne supérieure et c'est seulement à ce niveau que l'on va se donner un x .

Soit $x \in [0, \ell]$. La famille $(x_{n,p})_{0 \leq p \leq n}$ est une subdivision d'un segment qui contient $[0, \ell]$. Il existe donc un unique k tel que $x \in [x_{n,k} - \frac{1}{\sqrt{n}}, x_{n,k} + \frac{1}{\sqrt{n}}]$. On écrit alors que

$$|B_n(x) - \varphi(x)| \leq |B_n(x_{n,k}) - \varphi(x_{n,k})| + |\varphi(x_{n,k}) - \varphi(x)|$$

Comme $n \geq n_0$ (et donc $|x_{n,k} - x| \leq \frac{1}{\sqrt{n}} \leq r$), le second terme du membre de droite est plus petit que $\varepsilon/4$.

Le premier terme du membre de droite pose un petit problème. En effet, $x_{n,k} \in \left[x - \frac{1}{\sqrt{n}}, x + \frac{1}{\sqrt{n}}\right]$ donne bien $x_{n,k} \leq \ell + 1$ mais a priori pas $x_{n,k} \geq 0$ et donc pas $k \in I_n$.

- Si $x_{n,k} \geq 0$, on a bien $k \in I_n$ et $|B_n(x_{n,k}) - \varphi(x_{n,k})| \leq \varepsilon/4$ car $n \geq n'_0$. On a alors $|B_n(x) - \varphi(x)| \leq \frac{\varepsilon}{2}$.
- Sinon, $x_{n,k} < 0 \leq x < x_{n,k+1}$ et on est donc dans le cas où n est impair avec $2k+1 = n$ et $x_{n,k+1} = -x_{n,k}$ (question 18).
 $x_{n,k+1} \in [0, \ell+1]$ et donc $k+1 \in I_n$. De plus $B_n(x_{n,k}) = B_n(x_{n,k+1})$ et $\varphi(x_{n,k}) = \varphi(x_{n,k+1})$. On peut donc reprendre le raisonnement avec $x_{n,k+1}$ et conclure de même.

On a montré que

$$\boxed{\exists n_1, \forall n \geq n_1, \sup_{x \in [0, \ell]} |B_n(x) - \varphi(x)| \leq \frac{\varepsilon}{2}}$$

Remarque : on a

$$x_{n,k} \in \left[x - \frac{1}{\sqrt{n}}, x + \frac{1}{\sqrt{n}} \right] \subset [-1, \ell+1]$$

k n'est à priori pas dans I_n mais en changeant la définition de I_n en $\{k \in [0, n], x_{n,k} \in [-1, \ell]\}$, on ne change rien au raisonnement des questions qui précèdent. On aurait donc aussi pu conclure en supposant que n_1 a été choisi avec cette définition de I_n .

26. Soit $\ell > 0$. On n'a encore utilisé nulle part notre ε et le fait que $\varphi(\ell) \leq \varepsilon/2$. On peut donc reprendre la sous-partie II.B avec le segment $[0, \ell]$ et $\varepsilon = 2\varphi(\ell) > 0$. Pour n assez grand

$$|B_n(\ell) - \varphi(\ell)| \leq \varphi(\ell)$$

$$\boxed{\exists n_2, \forall n \geq n_2, \sup_{x \in [0, \ell]} |B_n(x) - \varphi(x)| \leq \varphi(\ell)}$$

27. Soit $n \geq \max(n_1, n_2)$. Par choix de n_1 , on a

$$\forall x \in [0, \ell], |B_n(x) - \varphi(x)| \leq \frac{\varepsilon}{2}$$

Comme B_n et φ sont décroissantes sur $\mathbb{R}^+$ (et positives)

$$\forall x \geq \ell, |B_n(x) - \varphi(x)| \leq |B_n(x)| + |\varphi(x)| \leq B_n(\ell) + \varphi(\ell)$$

Par choix de n_2 et de ℓ , cette quantité est plus petite que $3\varphi(\ell)$ et donc que $\frac{3\varepsilon}{2}$.

On a finalement

$$\forall n \geq \max(n_1, n_2), \sup_{x \geq 0} |B_n(x) - \varphi(x)| \leq \frac{3\varepsilon}{2}$$

Avec la question 20, on en déduit que $\Delta_n \rightarrow 0$ (définition des limites) et

$$\boxed{(B_n) \text{ converge uniformément vers } \varphi \text{ sur } \mathbb{R}}$$

Tomber sur un majorant égal à $\frac{3\varepsilon}{2}$ et pas ε n'est pas gênant puisque $\varepsilon > 0$ peut être choisi quelconque. On peut quand même s'interroger sur le choix étrange des majorants $\varepsilon/2$ en II.B par l'énoncé puisqu'en fin de compte on n'obtient pas un beau ε tout rond. Peut-être le concepteur a-t-il en tête une autre tactique. Cela n'a pas d'importance.

Applications

III.A - Théorème central limite

28. On forme la différence et on utilise la relation de Chasles :

$$\int_u^v f - \int_{u_n}^{v_n} f_n = \int_u^v (f - f_n) - \int_u^{u_n} f_n - \int_{v_n}^v f_n$$

Une fonction continue par morceaux sur un segment y est bornée. L'inégalité triangulaire et une majoration grossière donnent

$$\left| \int_u^v f - \int_{u_n}^{v_n} f_n \right| \leq |u - v| \|f - f_n\|_{\infty, [u, v]} + |u - u_n| \|f_n\|_{\infty, [u, u_n]} + |v - v_n| \|f_n\|_{\infty, [v, v_n]}$$

La convergence uniforme sur I entraîne celle sur $[u, v]$ et $\|f - f_n\|_{\infty, [u, v]} \rightarrow 0$.

Supposons que u et v soient intérieurs à I . Il existe alors $a, b \in I$ tels que u et v sont dans $]a, b[$. Pour n assez grand ($n \geq n_0$), on a encore $u_n \in [a, b]$ et $v_n \in [a, b]$ et

$$\forall n \geq n_0, |u - u_n| \|f_n\|_{\infty, [u, u_n]} \leq |u - u_n| \|f_n\|_{\infty, [a, b]}$$

Comme $\|f_n\|_{\infty, [a, b]} \rightarrow \|f\|_{\infty, [a, b]}$ avec la convergence uniforme sur $[a, b]$ (puisque sur I), le majorant est de limite nulle.

De même, $|v - v_n| \|f_n\|_{\infty, [v, v_n]} \rightarrow 0$.

Dans ce cas, $\left| \int_u^v f - \int_{u_n}^{v_n} f_n \right|$ est majoré par un terme de limite nulle.

Si u ou v est au bord de I , il suffit d'adapter en considérant un voisinage à droite ou à gauche selon le cas pour remplacer $]a, b[$.

$$\lim_{n \rightarrow +\infty} \int_{u_n}^{v_n} f_n = \int_u^v f$$

Comme en question 8, le problème est ici que l'indice n apparaît dans les bornes et dans l'intégrande. On pourrait (comme ci-dessus) justifier qu'il existe un segment $[a, b] \subset I$ tel que pour tout n , $u_n, v_n \in [a, b]$ (et donc aussi $u, v \in [a, b]$). Puis introduire les fonctions $g_n = f_n \times \mathbf{1}_{[a, b]}$. On doit alors étudier la suite de terme général $\int_a^b g_n$, ce que l'on peut faire à l'aide d'un théorème d'interversion.

29. Chaque Y_i suit la loi de Bernoulli de paramètre $1/2$ (loi uniforme sur $\{0, 1\}$). La mutuelle indépendance des X_i et le lemme des coalitions donne la mutuelle indépendance des Y_i . Le cours nous indique que T_n suit la loi binomiale de paramètres n et $1/2$ et

$$\forall j \in \llbracket 0, n \rrbracket, \mathbb{P}(T_n = j) = \binom{n}{j} \frac{1}{2^n}$$

Sur l'intervalle $[x_{n,j} - 1/\sqrt{n}, x_{n,j} + 1/\sqrt{n}[$, B_n est constante égale à $\frac{\sqrt{n}}{2^{n+1}} \binom{n}{j}$. Son intégrale sur le segment est donc égale à $2/\sqrt{n}$ (longueur du segment) fois cette valeur et ainsi

$$\forall j \in \llbracket 0, n \rrbracket, \mathbb{P}(T_n = j) = \int_{x_{n,j} - 1/\sqrt{n}}^{x_{n,j} + 1/\sqrt{n}} B_n$$

Le programme donnant des résultats sur les sommes de variables de Bernoulli indépendantes et de même paramètre, il est agréable de se ramener à cette situation quand on nous fait travailler avec des variables de Rademacher (à valeurs dans $\{-1, 1\}$ et non $\{0, 1\}$).

30. On a $S_n = 2T_n - n$ et donc

$$u \leq \frac{S_n(\omega)}{\sqrt{n}} \leq v \iff \frac{n + u\sqrt{n}}{2} \leq T_n(\omega) \leq \frac{n + v\sqrt{n}}{2}$$

On a donc l'égalité ensembliste (T_n étant à valeurs dans $\mathbb{Z}$)

$$\left(u \leq \frac{S_n}{\sqrt{n}} \leq v\right) = (T_n \in J_n) = \bigcup_{j \in J_n} (T_n(\omega) = j)$$

Comme la réunion est disjointe, le passage aux probabilités donne

$$\mathbb{P}\left(u \leq \frac{S_n}{\sqrt{n}} \leq v\right) = \sum_{j \in J_n} \mathbb{P}(T_n(\omega) = j)$$

31. Les questions 30 et 29 ainsi que la relation de Chasles donnent

$$\mathbb{P}\left(u \leq \frac{S_n}{\sqrt{n}} \leq v\right) = \int_{x_{n,i_n}-1/\sqrt{n}}^{x_{n,j_n}+1/\sqrt{n}} B_n$$

où i_n est le minimum de J_n et j_n son maximum (il est assez immédiat que J_n est un segment d'entiers). On a

$$i_n = \left\lceil \frac{n + u\sqrt{n}}{2} \right\rceil = \frac{n}{2} + \frac{u\sqrt{n}}{2} + O(1) \quad \text{et} \quad j_n = \left\lfloor \frac{n + v\sqrt{n}}{2} \right\rfloor = \frac{n}{2} + \frac{v\sqrt{n}}{2} + O(1)$$

On en déduit que

$$u_n = x_{n,i_n} - 1/\sqrt{n} = -\sqrt{n} + \frac{2i_n}{\sqrt{n}} = u + O\left(\frac{1}{\sqrt{n}}\right) \rightarrow u$$

et de même

$$v_n = x_{n,j_n} + 1/\sqrt{n} \rightarrow v$$

La partie II et la question 28 donnent alors

$$\lim_{n \rightarrow +\infty} \mathbb{P}\left(u \leq \frac{S_n}{\sqrt{n}} \leq v\right) = \int_u^v \varphi(x) dx$$

Pour tous les entiers $n, p \geq 1$, on considère l'événement $E_{n,p} = \left(u \leq \frac{S_n}{\sqrt{n}} \leq u + p\right)$. $(E_{n,p})_{p \geq 1}$ est une suite croissante d'événement dont la réunion est l'événement $(u \leq S_n/\sqrt{n})$. Par continuité croissante, on a donc

$$\lim_{p \rightarrow +\infty} \mathbb{P}(E_{n,p}) = \mathbb{P}\left(u \leq \frac{S_n}{\sqrt{n}}\right)$$

Sous réserve d'existence des limites,

$$\lim_{n \rightarrow +\infty} \mathbb{P} \left(u \leq \frac{S_n}{\sqrt{n}} \right) = \lim_{n \rightarrow +\infty} \lim_{p \rightarrow +\infty} \mathbb{P}(E_{n,p})$$

En admettant que l'interversion des limites est licite, on a alors

$$\lim_{n \rightarrow +\infty} \mathbb{P} \left(u \leq \frac{S_n}{\sqrt{n}} \right) = \lim_{p \rightarrow +\infty} \lim_{n \rightarrow +\infty} \mathbb{P}(E_{n,p}) = \lim_{p \rightarrow +\infty} \int_u^{u+p} \varphi = \int_u^\infty \varphi = 1 - \Phi(u)$$

Pour conclure, il nous suffit de montrer que l'on peut intervertir les limites (ce qui entraînera l'existence des limites).

On pose $F_p : n \mapsto \mathbb{P}(E_{n,p})$ et on cherche à utiliser le théorème de double limite pour (F_p) au voisinage de l'infini.

- Pour tout p , on a $F_p(n) \rightarrow \int_u^{u+p} \varphi$ quand $n \rightarrow +\infty$.
- Pour tout n , $F_p(n) \rightarrow \mathbb{P}(u \leq S_n/\sqrt{n}) = F(n)$ quand $p \rightarrow +\infty$. (F_p) converge donc simplement sur $\mathbb{N}^*$ vers F . On veut une convergence uniforme au voisinage de l'infini. On forme

$$|F(n) - F_p(n)| = \mathbb{P}(u \leq S_n/\sqrt{n}) - \mathbb{P}(u \leq S_n/\sqrt{n} \leq u+p) = \mathbb{P}(u+p < S_n/\sqrt{n})$$

Comme $S_n = 2T_n - n$, on a donc

$$|F(n) - F_p(n)| = \mathbb{P} \left(T_n > \frac{\sqrt{n}(u+p) + n}{2} \right)$$

T_n est une variable positive admettant un moment d'ordre 1 égal à $n/2$ (loi binomiale de paramètres n et $1/2$). On a ainsi

$$|F(n) - F_p(n)| = \mathbb{P}(T_n - \mathbb{E}(T_n) > \frac{\sqrt{n}(u+p)}{2}) \leq \mathbb{P}(|T_n - \mathbb{E}(T_n)| > \frac{\sqrt{n}(u+p)}{2})$$

Comme T_n admet un moment d'ordre 2 et $\mathbb{V}(T_n) = \frac{n}{4}$, l'inégalité de Tchebychev donne

$$|F(n) - F_p(n)| \leq \frac{n}{4} \frac{4}{n(u+p)^2} = \frac{1}{(u+p)^2}$$

Le majorant est indépendant de n et de limite nulle et (F_p) converge uniformément vers F sur $\mathbb{N}^*$.

On a ainsi justifié les calculs formels et prouvé que

$$\boxed{\lim_{n \rightarrow +\infty} \mathbb{P} \left(u \leq \frac{S_n}{\sqrt{n}} \right) = 1 - \Phi(u)}$$

III.B - Critère de tension

32. La question 12 donne $x^2(1 - \Phi(x)) \sim \frac{x}{\sqrt{2\pi}} e^{-x^2/2}$ au voisinage de $+\infty$. C'est donc (par croissances comparées) une fonction de limite nulle en $+\infty$. Il existe donc un réel $x_0 \geq 1$ tel que

$$\forall x \geq x_0, \quad x^2(1 - \Phi(x)) \leq \frac{\varepsilon}{4}$$

Soit $x \geq x_0$. Avec la question 31, il existe un rang n_x tel que

$$\forall n \geq n_x, \mathbb{P}\left(x \leq \frac{S_n}{\sqrt{n}}\right) \leq 1 - \Phi(x) + \frac{\varepsilon}{4x^2}$$

On a ainsi

$$\forall n \geq n_x, x^2 \mathbb{P}\left(x \leq \frac{S_n}{\sqrt{n}}\right) \leq \frac{\varepsilon}{2}$$

On peut appliquer ceci avec la suite de variables aléatoires $(-X_i)$ qui vérifie les mêmes hypothèses et obtenir

$$\forall n \geq n_x, x^2 \mathbb{P}\left(x \leq -\frac{S_n}{\sqrt{n}}\right) \leq \frac{\varepsilon}{2}$$

La combinaison des deux résultats donne

$$\boxed{\forall x \geq x_0, \exists n_x, \forall n \geq n_x, x^2 \mathbb{P}(|S_n| \geq x\sqrt{n}) \leq \varepsilon}$$

33. Appliquons la partie I.C avec $Z_i = X_i$. L'inégalité obtenue avec x remplacé par $x\sqrt{n}$ se lit

$$\mathbb{P}\left(\max_{1 \leq p \leq n} |S_p| \geq 3x\sqrt{n}\right) \leq 3 \max_{1 \leq p \leq n} \mathbb{P}(|S_p| \geq x\sqrt{n})$$

On veut majorer cette quantité par 3ε et le but est donc de montrer que

$$\forall p \in \llbracket 1, n \rrbracket, \mathbb{P}(|S_p| \geq x\sqrt{n}) \leq \frac{\varepsilon}{x^2}$$

(on aura alors un majorant indépendant de p et le maximum sera plus petit que ε/x^2).

- Supposons que $n_x \leq p \leq n$. On a alors $(|S_p| \geq x\sqrt{n}) \subset (|S_p| \geq x\sqrt{p})$ et donc

$$x^2 \mathbb{P}(|S_p| \geq x\sqrt{n}) \leq x^2 \mathbb{P}(|S_p| \geq x\sqrt{p}) \leq \varepsilon$$

- Si $p < n_x$, on va appliquer une inégalité de Tchebychev. Les X_i possédant un moment d'ordre 2, il en est de même de S_p . Par linéarité (pour l'espérance) et par indépendance des X_i (pour la variance), on a

$$\mathbb{E}(S_p) = \sum_{i=1}^p \mathbb{E}(X_i) = 0 \quad \text{et} \quad \mathbb{V}(S_p) = \sum_{i=1}^p \mathbb{V}(X_i) = p$$

Comme $x\sqrt{n} > 0$, on a donc

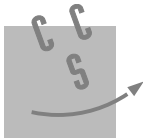
$$\mathbb{P}(|S_p| \geq x\sqrt{n}) = \mathbb{P}(|S_p - \mathbb{E}(S_p)| \geq x\sqrt{n}) \leq \frac{\mathbb{V}(S_p)}{(x\sqrt{n})^2} = \frac{p}{nx^2}$$

Et comme $1 \leq p < n_x$, si on choisit $n \geq \frac{n_x}{\varepsilon}$, on a $p \leq n\varepsilon$ et

$$\mathbb{P}(|S_p| \geq x\sqrt{n}) \leq \frac{n\varepsilon}{nx^2} = \frac{\varepsilon}{x^2}$$

On a ainsi montré que

$$\boxed{\forall n \geq \frac{n_x}{\varepsilon}, x^2 \mathbb{P}\left(\max_{1 \leq p \leq n} |S_p| \geq 3x\sqrt{n}\right) \leq 3\varepsilon}$$



Option informatique

MP

2023

CONCOURS CENTRALE-SUPÉLEC

4 heures

Calculatrice autorisée

Ce sujet comporte quatre parties.

La partie I est totalement indépendante des suivantes et étudie des transformations sur des langages.

Les autres parties s'intéressent à des structures de données représentant des ensembles d'entiers naturels E contenus dans $\llbracket 0, N-1 \rrbracket$, « denses » au sens où $|E|$ est du même ordre de grandeur que N . On cherche à optimiser le temps d'exécution des opérations de test d'appartenance, d'insertion ou de suppression d'un élément, de calcul du minimum, de calcul de l'élément de E immédiatement supérieur à x (qu'on appellera successeur de x dans E , même si x n'appartient pas à E) ou d'opérations ensemblistes telles que l'union.

Dans la partie II, on étudie des structures ordinaires dont on identifie le défaut. Dans la partie III, on étudie une structure simple d'arbre complet modélisant des parties de $\llbracket 0, 2^p - 1 \rrbracket$ et dans la partie IV, on implémente des arbres de van Emde Boas qui, sur des parties E de $\llbracket 0, 2^{2^p} - 1 \rrbracket$, donnent des opérations en temps quasi-constant.

I Langages et automates

Dans cette partie, on s'intéresse à des transformations sur des langages définis sur un alphabet à deux lettres $X = \{a, b\}$. On note $L_1 | L_2$ la réunion de deux langages L_1 et L_2 et ε le mot vide (ou le langage réduit au mot vide, suivant le contexte). Les automates considérés sont des quadruplets (Q, I, F, Δ) où Q est l'ensemble des états de l'automate, I l'ensemble de ses états initiaux, F l'ensemble de ses états finals et $\Delta \subset Q \times X \times Q$ l'ensemble de ses transitions.

Soit L et K deux langages. On définit le langage noté $L \triangleright K$ par

$$L \triangleright K = \{uvw \mid (u, v, w) \in (X^*)^3, uv \in L, w \in K\}$$

Q 1. Déterminer les langages suivants :

$$a^* \triangleright b^* \qquad (ba)^* \triangleright (ab)^* \qquad a^* \triangleright \left\{ a^p b a^q \mid (p, q) \in \mathbb{N}^2, p \leq q \right\}$$

Q 2. Soit L , K_1 et K_2 trois langages. Exprimer à l'aide de $L \triangleright K_1$, $L \triangleright K_2$, K_1 et K_2 les langages suivants :

$$L \triangleright (K_1 | K_2) \qquad L \triangleright (K_1 \cdot K_2) \qquad L \triangleright (K_1^*)$$

On ne justifiera que la formule portant sur l'étoile de Kleene.

Q 3. Montrer que si L et K sont deux langages réguliers, $L \triangleright K$ est régulier.

Q 4. Donner l'exemple d'un langage régulier L et d'un langage non régulier K tel que $L \triangleright K$ soit régulier. On justifiera en détail que le langage K proposé n'est pas régulier.

Q 5. À l'aide d'automates reconnaissant des langages réguliers L et K et d' ε -transitions, construire un automate à ε -transitions reconnaissant $L \triangleright K$. On expliquera la construction puis on formalisera l'automate correspondant. Il est inutile de justifier qu'il convient.

On définit sur X^* l'application σ par :

$$\begin{cases} \sigma(\varepsilon) = \varepsilon \\ \forall u \in X^*, \forall x \in X, \quad \sigma(ux) = xu \end{cases}$$

On note alors pour tout langage L , $\widehat{L}$ le langage défini par :

$$\widehat{L} = \bigcup_{k=0}^{+\infty} \sigma^k(L)$$

Q 6. Pour chacun des langages L suivants, déterminer $\sigma(L)$ puis $\widehat{L}$:

$$L_1 = (ab)^* \qquad L_2 = a^*(ba|b)$$

Q 7. Soit L un langage régulier et $\mathcal{A} = (Q, I, F, \Delta)$ un automate le reconnaissant. Pour $(q, q') \in Q^2$, on note $L_{q,q'}$ l'ensemble des mots qui étiquettent un chemin de l'état q à l'état q' dans $\mathcal{A}$. Exprimer $\sigma(L)$ à l'aide de langages $L_{q,q'}$ et du langage X . On portera une attention particulière au cas du mot vide et on justifiera la formule proposée.

- Q 8.** Montrer que si L est régulier, $\sigma(L)$ est régulier.
- Q 9.** Montrer que si L n'est pas régulier, $\sigma(L)$ ne l'est pas non plus.
- Q 10.** Montrer que si L est régulier, $\widehat{L}$ est régulier. Étudier la réciproque.

II Représentations classiques d'ensembles

Dans cette partie, on implémente des ensembles par des structures connues. On note $|E|$ le cardinal d'un ensemble E .

II.A – Avec une liste triée

Q 11. Dans cette question uniquement, on implémente un ensemble d'entiers positifs par la liste de ses éléments, rangés **dans l'ordre croissant**. Écrire une fonction `succ_list` de signature `int list -> int -> int` prenant en arguments une liste d'entiers distincts dans l'ordre croissant et un entier x et renvoyant le successeur de x dans la liste, c'est-à-dire le plus petit entier strictement supérieur à x de la liste (-1 si cela n'existe pas). Donner sa complexité dans le pire cas.

II.B – Avec un vecteur trié

Soit N un entier naturel strictement positif, fixé pour toute cette partie. On choisit de représenter un ensemble d'entiers E de cardinal $n \leq N$ par un tableau t de taille $N + 1$ dont la case d'indice 0 indique le nombre n d'éléments de E et les cases d'indices 1 à n contiennent les éléments de E rangés dans l'ordre croissant, les autres cases étant non significatives. Par exemple, le tableau `[13;2;5;7;9;1;14]` représente l'ensemble à 3 éléments $\{2, 5, 7\}$.

Q 12. Pour une telle implémentation d'un ensemble E , décrire brièvement des méthodes permettant de réaliser chacune des opérations ci-dessous (on ne demande pas d'écrire des programmes) et donner leurs complexités dans le pire cas :

- déterminer le maximum de E ;
- tester l'appartenance d'un élément x à E ;
- ajouter un élément x dans E (on suppose la taille du tableau suffisante et que x n'appartient pas à E).

Q 13. Par une méthode dichotomique, écrire une fonction `succ_vect` de signature `int array -> int -> int` prenant en arguments un tableau t codant un ensemble E comme ci-dessus et un entier x et renvoyant le successeur de x dans E (-1 si cela n'existe pas).

Q 14. Calculer la complexité dans le pire cas de la fonction `succ_vect` en fonction de n .

Q 15. Écrire une fonction `union_vect` de signature `int array -> int array -> int array` prenant en arguments deux tableaux t_1 et t_2 , de taille N , codant deux ensembles E_1 et E_2 et renvoyant le tableau correspondant à $E_1 \cup E_2$. On supposera que $|E_1 \cup E_2| \leq N$.

II.C – Avec un arbre binaire de recherche

On choisit maintenant de représenter un ensemble d'entiers à l'aide d'un arbre binaire de recherche, les nœuds étant étiquetés par les éléments de E . On définit le type

```
type abr = Nil | Noeud of int * abr * abr
```

Pour un tel arbre, pour **tout** nœud x , les étiquettes de tous les nœuds du sous-arbre gauche de x , s'il y en a, doivent être inférieures à l'étiquette de x et les étiquettes de tous les nœuds du sous-arbre droit de x , s'il y en a, doivent être supérieures à l'étiquette de x .

Par exemple, on peut représenter l'ensemble $\{2, 3, 5, 8, 13\}$ par l'arbre figure 1.

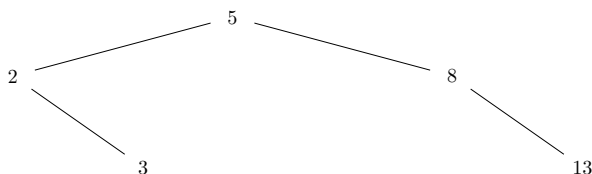


Figure 1 Un arbre binaire de recherche pour l'ensemble $\{2, 3, 5, 8, 13\}$

Q 16. Écrire une fonction `min_abr` de signature `abr -> int` prenant en argument un arbre binaire de recherche représentant un ensemble E et renvoyant son étiquette minimale (-1 si l'ensemble est vide).

Q 17. Écrire une fonction récursive `partitionne_abr` de signature `abr -> int -> (bool * abr * abr)` prenant en arguments un arbre binaire de recherche représentant un ensemble E et un entier x et renvoyant un

triplet (b, ag, ad) où b vaut **true** si x appartient à E et **false** sinon, ag est un arbre binaire de recherche codant les éléments de E strictement plus petits que x et ad un arbre binaire de recherche codant les éléments de E strictement plus grands que x .

Q 18. Écrire une fonction `insertion_abr` de signature `abr -> int -> abr` prenant en arguments un arbre binaire de recherche représentant un ensemble E et un entier x et renvoyant un arbre binaire de recherche associé à l'ensemble $E \cup \{x\}$ et de racine étiquetée par x . Calculer sa complexité dans le pire cas en fonction de l'arbre reçu puis en fonction de E .

Q 19. Écrire une fonction `union_abr` de signature `abr -> abr -> abr` prenant en arguments deux arbres binaires de recherche représentant deux ensembles E_1 et E_2 et renvoyant un arbre binaire de recherche associé à l'ensemble $E_1 \cup E_2$. On expliquera brièvement la méthode choisie.

III Représentation par arbres binaires complets

On considère dans cette partie des arbres binaires complets dont les nœuds sont étiquetés par des booléens. On rappelle que la profondeur d'un nœud est égale à la longueur du chemin reliant la racine à ce nœud et la hauteur d'un arbre est la plus grande profondeur de ses feuilles (la racine est donc à la profondeur 0). Les nœuds seront numérotés à partir de la racine qui porte le numéro 1 dans l'ordre d'un parcours en largeur (de gauche à droite à chaque profondeur).

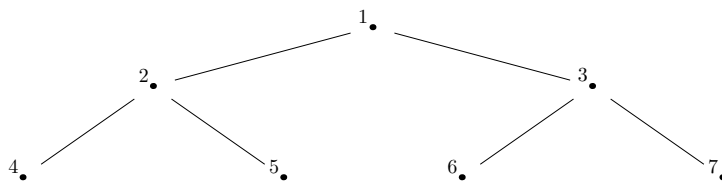


Figure 2 Numérotation des nœuds

Q 20. Dans un arbre binaire complet de hauteur $p \in \mathbb{N}$, quel numéro peut avoir un nœud à la profondeur $k \in \llbracket 0, p \rrbracket$? Combien le sous-arbre dont la racine a le numéro i a-t-il de feuilles ?

Q 21. Dans un arbre binaire complet de hauteur $p \in \mathbb{N}$, pour le nœud numéro i , donner, en les justifiant, les numéros de son fils gauche, de son fils droit et de son père.

Informatiquement, un tel arbre complet de hauteur p sera implémenté par un tableau de booléens de taille 2^{p+1} , la case d'indice $i \in \llbracket 1, 2^{p+1} - 1 \rrbracket$ contenant l'étiquette du nœud numéroté i . La case d'indice 0 n'est pas utilisée. On notera que dans tous les programmes de cette partie, on peut avoir accès à la valeur 2^p via la taille du tableau.

Soit $p \in \mathbb{N}$. On choisit de coder un ensemble $E \subset \llbracket 0, 2^p - 1 \rrbracket$ par un arbre binaire complet de hauteur p selon les règles suivantes :

- chaque feuille numérotée i est étiquetée par **true** si et seulement si $i - 2^p \in E$;
- chaque nœud interne est étiqueté par le « ou logique » des étiquettes de ses deux fils.

```
let ensemble = bool array ;;
```

Par exemple, l'ensemble $\{0, 1, 7\}$ est représenté par l'arbre représenté figure 3.

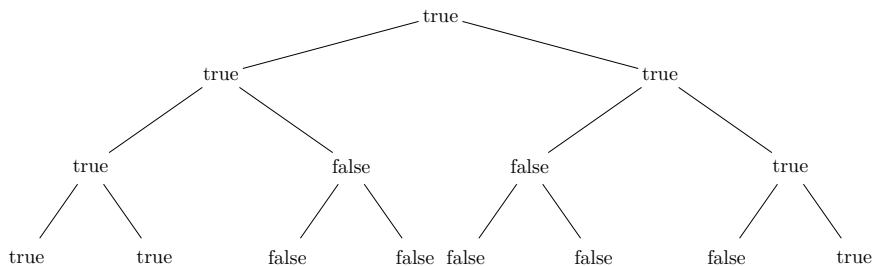


Figure 3 Arbre associé à l'ensemble $\{0, 1, 7\}$ pour $p = 3$

Q 22. Écrire une fonction `appartient` de signature `ensemble -> int -> bool` qui détermine si un entier quelconque appartient ou non à un ensemble donné. Calculer sa complexité.

Q 23. Écrire une fonction `fabrique` de signature `int list -> int -> ensemble` qui prend en arguments une liste d'entiers positifs distincts ℓ et une valeur pour 2^p et renvoie l'arbre associé à l'ensemble E dont les éléments sont ceux de la liste. On supposera que tous les éléments de ℓ appartiennent à $\llbracket 0, 2^p - 1 \rrbracket$. Cette fonction devra s'exécuter en $\mathcal{O}(2^p)$.

Q 24. Écrire une fonction `insere` de signature `ensemble -> int -> unit` qui ajoute un entier k à un ensemble E . On suppose k compatible avec la valeur de p associée à E . Cette fonction devra s'exécuter en $\mathcal{O}(1)$ dans le meilleur cas.

Q 25. Écrire une fonction `supprime` de signature `ensemble -> int -> unit` qui retire un entier k d'un ensemble E . On suppose k compatible avec la valeur de p associée à E . Calculer la complexité de cette fonction dans le pire cas.

Q 26. Écrire une fonction `minlocal` de signature `ensemble -> int -> int` qui cherche l'élément de E minimal parmi ceux codés dans le sous-arbre de racine numérotée i dans l'arbre associé à E . Si un tel élément n'existe pas, cette fonction devra renvoyer -1 . Calculer la complexité de cette fonction en fonction de p et i .

On veut maintenant écrire une fonction qui calcule le successeur d'un entier x dans un ensemble E pour une telle structure. On propose l'algorithme suivant, pour $x \in \llbracket 0, 2^p - 1 \rrbracket$:

- on part de la case numéro i codant l'entier x dans E ;
- tant que i n'est pas le nœud le plus à droite à sa profondeur et que la case $i + 1$ vaut **false**, on remplace i par son père ;
- on renvoie l'élément minimum du sous-arbre de racine $i + 1$ ou -1 si i était totalement à droite.

Q 27. Prouver l'algorithme décrit ci-dessus.

Q 28. Écrire une fonction `successeur` de signature `ensemble -> int -> int` prenant en arguments l'ensemble E et un entier x positif et renvoyant son successeur dans E (-1 si cela n'existe pas).

Q 29. Montrer que si $x \in E$ admet bien un successeur dans E , il existe une constante $K > 0$ indépendante de E et p telle que la complexité de `successeur e x` soit majorée par $K(\log_2(\text{successeur}(x) - x) + 2)$. On admet que le même type de justification montre que si x est le maximum de E , la complexité de `successeur e x` est majorée par $K(\log_2(2^p - x) + 2)$.

Q 30. En utilisant la fonction `successeur`, écrire une fonction `cardinal` de signature `ensemble -> int` prenant en argument un ensemble et renvoyant son cardinal.

Q 31. Déterminer la complexité de la fonction `cardinal` en fonction de p et $n = |E|$. On rappelle que la fonction $\log_2$ est concave.

Q 32. Quels sont les intérêts et inconvénients d'une telle structure ? Dans quels cas peut-elle s'avérer plus intéressante que des structures connues ?

IV Arbres de van Emde Boas

Soit p un entier positif et $N = 2^{2^p}$. On supposera que tous les entiers manipulés restent représentables par la structure d'entier OCaml ordinaire. On considère le type de structure suivant (appelé arbre *veb* par suite) implémentant un ensemble E d'entiers positifs strictement inférieurs à N :

```
type veb = {mutable mini : int; mutable maxi : int; table : veb array};;
```

Les champs mutables d'un arbre *veb* sont modifiables : par exemple, pour un arbre *veb* noté `v`, `v.mini <- 1` change la valeur du champ `v.mini`.

Le codage d'un ensemble $E \subset \llbracket 0, N - 1 \rrbracket$ par un arbre *veb* dit d'ordre $N = 2^{2^p}$ suit les règles suivantes :

- Les champs `mini` et `maxi` représentent toujours la valeur minimale et maximale de E . Ils sont mis arbitrairement à -1 si l'ensemble est vide.
- Si $N = 2$, *i.e.* si l'arbre doit coder une partie de $\llbracket 0, 1 \rrbracket$, le champ `table` est un tableau vide (*i.e.* `[]`), les champs `mini` et `maxi` suffisant à coder E . On veillera à ce que les fonctions demandées traitent correctement ce cas particulier.
- Pour $N > 2$, on note $\widehat{E} = E \setminus \{\min(E)\}$ (où $\min(E)$ désigne le minimum de E). On décompose $\widehat{E}$ en $\sqrt{N} = 2^{2^{p-1}}$ ensembles E_k définis par

$$\forall k \in \llbracket 0, \sqrt{N} - 1 \rrbracket, \quad E_k = \left\{ x - k\sqrt{N} \mid x \in \widehat{E} \cap \llbracket k\sqrt{N}, (k+1)\sqrt{N} - 1 \rrbracket \right\}$$

le champ `table` est alors un tableau de $\sqrt{N} + 1$ arbres *veb* d'ordre $\sqrt{N}$:

- pour $k \in \llbracket 0, \sqrt{N} - 1 \rrbracket$, l'arbre *veb* stocké dans la case `table.(k)` code l'ensemble E_k ;
- l'arbre *veb* stocké dans la case `table.( $\sqrt{N}$ )` code l'ensemble $R = \left\{ k \in \llbracket 0, \sqrt{N} - 1 \rrbracket \mid E_k \neq \emptyset \right\}$.

Par exemple, considérons l'arbre *veb* **ex** codant l'ensemble $E = \{2, 13, 14, 15\}$ avec $p = 2$ i.e. $N = 16$. On a $\widehat{E} = \{13, 14, 15\}$ puisque $\min(E) = 2$ et donc les cinq cases du tableau **ex.table** correspondent respectivement aux ensembles

$$E_0 = \emptyset \quad E_1 = \emptyset \quad E_2 = \emptyset \quad E_3 = \{13 - 3\sqrt{16}, 14 - 3\sqrt{16}, 15 - 3\sqrt{16}\} = \{1, 2, 3\} \quad R = \{3\}$$

On obtient la structure représentée figure 4.

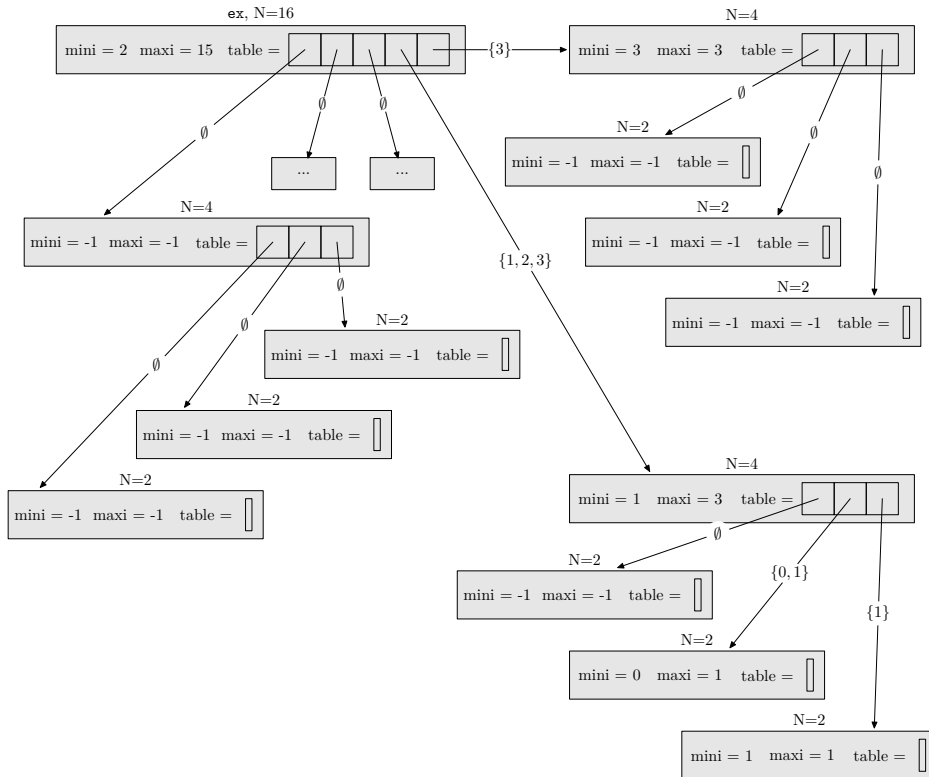


Figure 4 L'arbre *veb* **ex** associé à l'ensemble $\{2, 13, 14, 15\}$ avec $p = 2$

- Q 33.** On veut coder l'ensemble $\{0, 2, 3, 5, 7, 13, 14\}$ par un arbre *veb* d'ordre $N = 16$, noté **ex_veb**. Indiquer quel ensemble code chaque arbre *veb* stocké dans **ex_veb.table** puis préciser **ex_veb.table.(3)**.
- Q 34.** Écrire une fonction **creer_veb** de signature **int** $\rightarrow$ **veb** prenant en argument un entier p et créant un arbre *veb* d'ordre $N = 2^{2^p}$ implémentant l'ensemble vide.
- Q 35.** Résoudre en fonction de q la récurrence $C(q) = C(\sqrt{q}) + \mathcal{O}(1)$ dans le cas où $q = 2^{2^r}$.
- Q 36.** Écrire une fonction **appartient_veb** de signature **veb** $\rightarrow$ **int** $\rightarrow$ **bool** qui teste l'appartenance d'un entier x quelconque à un ensemble E codé par un arbre *veb*. Calculer sa complexité dans le pire cas.
- Q 37.** Écrire une fonction **successeur_veb** de signature **veb** $\rightarrow$ **int** $\rightarrow$ **int** qui calcule le successeur d'un entier x quelconque dans E (-1 s'il n'y en a pas). Calculer sa complexité dans le pire cas.
- Q 38.** Écrire une fonction **insertion_veb** de signature **veb** $\rightarrow$ **int** $\rightarrow$ **unit** qui insère un entier x dans un arbre *veb*. On suppose que $x \in \llbracket 0, N-1 \rrbracket$. Cette fonction devra avoir une complexité en $\mathcal{O}(\log_2(\log_2(N)))$.
- Q 39.** Soit $M(N)$ la totalité de l'espace mémoire nécessaire pour stocker un ensemble par un arbre *veb* d'ordre N avec $N = 2^{2^p}$. Donner la relation de récurrence vérifiée par $M(N)$ puis déterminer l'ordre de grandeur de $M(N)$.

• • • FIN • • •

SOLUTION

Q1. Par construction $L \triangleright K$ est formé d'un mot k de K que l'on factorise en $k = uv$ et on insère entre u et v , un mot arbitraire v de L (on insère un mot arbitraire de L dans un mot de K) Par conséquent, on obtient que :

$a^* \triangleright b^* = a^* b^* a^*$ (toute factorisation de $k \in a^*$ est constitué de deux mots u et v de a^* et on insère entre les deux, un mot de b^*).

$(ba)^* \triangleright (ab)^* = (ab)^* (ba)^* (ab)^* \mid (ab)^* a (ba)^* b (ab)^*$ (un mot $k = uv$ de $(ab)^*$ qui est scindé en deux sous mots u, v , le premier u est une succession (éventuellement vide) de ab alors v aussi puis on place entre les deux un mot de $(ba)^*$ d'où un mot de $(ab)^* (ba)^* (ab)^*$, ou bien la coupure de k s'arrête sur un a donc le mot u appartient à $(ab)^* a$ donc l'autre mot v commence par un b , pour créer le motif ab , puis une succession de ab donc un mot de $b(ab)^*$, ce qui va engendré un mot de $(ab)^* a (ba)^* b (ab)^*$).

On pose $L = \{a^p b a^q, (p, q) \in \mathbb{N}^2, p \leq q\}$ alors $a^* \triangleright L = a^* b a^*$. Si un mot $k = uv$ de L ,

- soit $u = a^r$ et $v = a^{p-r} b a^q$ avec $q \geq p$, on insère un mot de a^* c'est-à-dire un nombre arbitraire de a entre u et v donc on obtient n'importe quel mot de $a^* b a^*$ car le nombre de a avant b et après b sont arbitraires et sans lien,
- soit $u = a^p b a^s$ et $v = a^{q-s}$ avec $q \geq p$, on insère un mot de a^* donc un nombre arbitraire de a entre u et v alors ce mot avec des a entourant l'unique b , le nombre de a précédant b est a^p qui est inférieur au nombre de a ensuite (car il y en a au moins $s + (q - s) = q \geq p$) c'est-à-dire que c'est un mot de L qui est inclus dans $a^* b a^*$

Commentaires : Question de difficulté standard. Il faut être attentif à ce type de question car elle aide le candidat à comprendre sur des exemples simples mais significatifs les notions introduites par le sujet. C'est en général indispensable pour traiter convenable un certain nombre de questions suivantes.

Q2. $L \triangleright (K_1 \mid K_2) = (L \triangleright K_1) \mid (L \triangleright K_2)$

$L \triangleright (K_1.K_2) = (L \triangleright K_1).K_2 \mid K_1.L.K_2 \mid K_1.(L \triangleright K_2)$ (selon que la coupure se fait au sein d'un mot de K_1 , ou en séparant le mot de K_1 du mot de K_2 ou au sein d'un mot de K_2).

$L \triangleright (K_1)^* = L \mid (K_1)^* (L \triangleright K_1) (K_1)^*$ (si le mot initial k est vide, on pose $u = \varepsilon$, $w = \varepsilon$ et v n'importe quel mot de L , sinon il y a une succession de mots de K_1 puis une coupure sur un mot de K_1 où l'on va insérer un mot de L puis une succession de mots de K_1).

Commentaires : Même remarque que précédemment.

Q3. Première réponse. Si L et K sont réguliers, ils sont reconnus par des automates (non nécessairement déterministes) $A_L = (Q_L, I_L, F_L, \delta_L)$ et $A_K = (Q_K, I_K, F_K, \delta_K)$. On considère l'automate A qui connecte tout état de l'automate A_K à tout état initial de A_L en effectuant une ε -transition puis en connectant tout état final de A_L à tout état de A_K par une ε -transition. Les états initiaux et finaux de cet automate étant ceux de A_K , les transitions autre que les ε -transitions étant celles dans A_K ou A_L selon la position de lecture. Un mot est reconnu par cet automate si et seulement s'il le début de la lecture de ce mot est un facteur u d'un mot m de K puis la lecture d'un mot de L puis la lecture d'un mot v de sorte que uv soit forcément en mot de K (car il commence à un état initial, transite dans A_K , s'interrompt avec la ε -transition puis revient à A_K par la seconde ε -transition pour aboutir à un état final de A_K) reste v de ce mot m donc d'un mot uvw avec $uw \in K$ et $v \in L$. Ainsi, cet automate reconnaît $L \triangleright K$ donc ce langage est régulier.

Deuxième réponse. On procède par récurrence sur le nombre d'opérations formant l'expression régulière E associée à K (on identifiera les deux ensuite par simplicité de lecture). Rappelons que les langages réguliers sont stables par concaténation, $|$, étoile de Kleene. Qu'ils contiennent le langage vide, le langage réduit à un élément (ε ou une lettre de Σ).

Si $K = \emptyset$ alors $L \triangleright K = \emptyset$ est un langage régulier. Si $K = \varepsilon$ alors $L \triangleright K = L$ est régulier.

Si $K = \{a\}$ alors $L \triangleright K = aL | La$ (selon on l'on fait s'insertion). Comme L et a sont réguliers, La et aL aussi donc $aL | La = L \triangleright K$ est régulier.

Si $K = K_1 | K_2$ avec K_1 et K_2 réguliers alors

$$L \triangleright K = (L \triangleright K_1 | L \triangleright K_2)$$

Par hypothèse de récurrence, K_1 et K_2 sont formés de strictement moins d'opérations que K donc $L \triangleright K_1$ et $L \triangleright K_2$ sont réguliers donc $(L \triangleright K_1 | L \triangleright K_2) = L \triangleright K$ aussi.

Si $K = K_1.K_2$ avec K_1 et K_2 réguliers alors

$$L \triangleright K = (L \triangleright K_1) . K_2 | K_1 . L . K_2 | K_1 . (L \triangleright K_2)$$

Par hypothèse de récurrence, K_1 et K_2 sont formés de strictement moins d'opérations que K donc $L \triangleright K_1$, $L \triangleright K_2$ et $K_1.L.K_2$ sont réguliers donc

$$(L \triangleright K_1) . K_2 | K_1 . L . K_2 | K_1 . (L \triangleright K_2) = L \triangleright K$$

aussi.

Si $K = (K_1)^*$ avec K_1 régulier alors

$$L \triangleright K = L | K (L \triangleright K_1) K$$

Par hypothèse de récurrence, K_1 est formé de strictement moins d'opérations que K donc $L \triangleright K$ est régulier donc $L | K (L \triangleright K_1) K = L \triangleright K$ aussi.

Ceci achève l'hérédité donc la récurrence.

Commentaires : Question classique à ce concours même si cela reste difficile pour un nombre important de candidats. N'hésitez pas à vous entraîner sur ce type de preuves (fondées sur les définitions, les théorèmes de concours et par récurrence sur le nombre d'opérations formant l'expression régulière ou le langage régulier).

Q4. On pose $L = a^*$ et $K = \{a^p b a^q, (p, q) \in \mathbb{N}^2, p \leq q\}$ (les exemples de la question 1) alors $L \triangleright K = a^* b a^*$. Il est immédiat que L et $L \triangleright K$ sont réguliers. Montrons par le lemme de pompage (ou de l'étoile) que le langage K n'est pas régulier. Si c'était le cas, il existe un entier N tel que tout mot m de K tel que $|m| \geq N$ alors on peut l'écrire $m = uvw$ avec $|uv| \leq N$, $v \neq \varepsilon$ et pour tout entier k , $uv^k w$ appartient à K . Considérons le mot $m = a^{2N} b a^{2N}$ qui appartient à K , il peut donc s'écrire $m = uvw$ avec $|uv| \leq N$ donc u et v ne contiennent de que a d'où $u = a^s$ pour un certain entier s , $v = a^r$ pour un entier $r \geq 1$ et w s'écrit $a^t b a^{2N}$. D'après le lemme de pompage, le mot

$$uv^k w = a^{s+kr+t} b a^{2N}$$

est toujours dans K quelque soit l'entier k . Or, $s + kr + t$ tend vers $+\infty$ quand k tend vers $+\infty$ donc il dépasse strictement l'entier $2N$, ce qui contredit l'appartenance de ce mot à K . Par conséquent, K n'est pas régulier.

Commentaires : Question difficile car il s'agit d'un contre-exemple à trouver et le candidat doit savoir montrer qu'un langage n'est pas régulier (classique à ce concours avec le lemme de pompage).

Q5. Il s'agit de la première réponse à la question 3. Soient $A_K = (Q_K, I_K, F_K, \Delta_K)$ et $A_L = (Q_L, I_L, F_L, \Delta_L)$ les automates reconnaissant respectivement K et L . On crée une copie $A'_K = (Q'_K, I'_K, F'_K, \Delta'_K)$ de A_K . On présume que les numérotations choisies permettent de n'avoir aucun état commun à tous ces automates. On pose :

- $Q = Q_K \cup Q_L \cup Q_{K'} ; I = I_K ; F = F_K ;$
- $\Delta = \Delta_K \cup \Delta_L \cup \Delta_{K'} \cup \Delta_{K,\varepsilon,L} \cup \Delta_{L,\varepsilon,K}$ avec

$$\begin{aligned}\Delta_{K,\varepsilon,L} &= \{(q, \varepsilon, i), q \in Q_K, i \in I_L\} \\ \Delta_{L,\varepsilon,K} &= \{(f, \varepsilon, q), f \in F_K, q \in Q_K\}\end{aligned}$$

alors $A = (Q, I, F, \Delta)$ est l'automate demandé par le sujet.

Commentaires : Le dessin (ou la déclaration en français) de l'automate est élémentaire .. si le langage $L \supset K$ est compris par le candidat. Pour la seconde, elle demande une meilleure compréhension des automates donc elle est plus sélective que la première.

Q6. $L_1 = (ab)^*$. Tout mot m de L_1 s'écrit $m = \varepsilon$ ou $m = abab \cdots ab$ donc

$$\sigma(m) = \varepsilon \text{ ou } \sigma(m) = b(abab \cdots aba) = baba \cdots ba$$

donc $\sigma(L_1) = (ba)^*$ d'où $\sigma^2(L_1) = \sigma((ba)^*) = (ab)^* = L_1$. Il est immédiat que $\sigma^k(L_1)$ vaut alternativement $(ab)^*$ puis $(ba)^*$ d'où

$$\widehat{L_1} = (ab)^* \mid (ba)^*.$$

$L_2 = a^*(ba \mid b)$. Tout mot m de L_2 s'écrit $m = a \cdots aba$ ou $m = a \cdots ab$ (le nombre de a avant b pouvant être réduit à 0) donc

$$\begin{aligned}\sigma(m) &= a \cdots ab = b \text{ ou } \sigma(m) = ba \cdots a \\ \sigma^2(m) &= ba \cdots a \text{ ou } \sigma^2(m) = aba \cdots a \\ \sigma^3(m) &= aba \cdots a \text{ ou } \sigma^3(m) = aaba \cdots a\end{aligned}$$

etc. Chaque $\sigma^k(m)$ est un mot constitué d'un seul b entouré d'un nombre arbitraire de a (à sa gauche et sa droite). Réciproquement, tout mot de cette forme est un $\sigma^k(m)$ pour un certain entier k et $m \in L_2$. Par conséquent,

$$\widehat{L_2} = a^*ba^*.$$

Commentaires : Question sans difficulté particulière pour la première. Pour la seconde, elle demande un peu de recul et d'initiative du candidat donc elle est sélective.

Q7. Par définition, on a $L = \bigcup_{i \in I, f \in F} L_{i,f}$ donc

$$\sigma(L) = \bigcup_{i \in I, f \in F} \sigma(L_{i,f}).$$

Soient i un état initial, f un état final et un mot $m \in L_{i,f}$. Si $m = \varepsilon$ alors $\sigma(m) = \varepsilon$. Si m est non vide, il est de la forme $m = ux$ pour un mot u et x une lettre de X . On pose

$q = \delta^*(i, u)$ (l'état où l'on aboutit depuis i en lisant le mot u) alors (q, x, f) est la dernière transition de la lecture du mot m depuis l'état i . On peut affirmer que $\sigma(m) = xu \in x.L_{i,q}$. La réciproque est immédiate donc

$$\sigma(L_{i,f}) = \bigcup_{\substack{q \in Q, x \in X \\ \text{tel que } (q,x,f) \in \Delta}} xL_{i,q}$$

donc, si ε n'appartient pas à L , on a

$$\sigma(L) = \bigcup_{i \in I, f \in F} \bigcup_{\substack{q \in Q, x \in X \\ \text{tel que } (q,x,f) \in \Delta}} xL_{i,q}$$

et si $\varepsilon \in L$, on a

$$\sigma(L) = \{\varepsilon\} \cup \bigcup_{i \in I, f \in F} \bigcup_{\substack{q \in Q, x \in X \\ \text{tel que } (q,x,f) \in \Delta}} xL_{i,q}$$

Commentaires : Question difficile destinée aux bonnes copies.

Q8. Les langages réguliers sont stables par union, concaténation et contient l'ensemble $\{\varepsilon\}$ ainsi que $\{x\}$ où x est une lettre de X . Comme

$$\sigma(L) = \bigcup_{i \in I, f \in F} \bigcup_{\substack{q \in Q, x \in X \\ \text{tel que } (q,x,f) \in \Delta}} xL_{i,q} \text{ ou } \sigma(L) = \{\varepsilon\} \cup \bigcup_{i \in I, f \in F} \bigcup_{\substack{q \in Q, x \in X \\ \text{tel que } (q,x,f) \in \Delta}} xL_{i,q}$$

Si chaque langage $L_{i,q}$ est régulier alors $\sigma(L)$ est régulier. Comme L est régulier et que l'automate $A = (Q, I, F, \Delta)$ le reconnaît, si on pose $A_{i,q} = (Q, \{i\}, \{q\}, \Delta)$ est un automate qui reconnaît le langage $L_{i,q}$. Ce dernier langage est régulier donc $\sigma(L)$ l'est aussi.

Commentaires : Même remarque que précédemment.

Q9. Comme $\sigma(ux) = xu$ alors σ est bijective avec $\sigma^{-1}(xu) = ux$ et $\sigma^{-1}(\varepsilon) = \varepsilon$. Le raisonnement tenu pour σ reste valable pour σ^{-1} (changer $L_{i,q}$ par $L_{q,f}$). Soit L non régulier. Si $\sigma(L)$ est régulier alors $\sigma^{-1}(\sigma(L)) = L$ est régulier d'après la question précédente, ce qui est absurde donc $\sigma(L)$ n'est pas régulier.

Commentaires : Question difficile qui demande une autonomie certaine du candidat.

Q10. Un mot de $\widehat{L}$ s'obtient à partir d'un mot en faisant tourner un certain nombre de lettres de sa fin à son début c'est-à-dire que l'on passe d'un mot

$$m = m_0 \cdots m_i m_{i+1} \cdots m_n$$

au mot

$$m' = m_{i+1} m_{i+2} \cdots m_n m_0 \cdots m_i$$

(on applique σ une fois pour faire passer m_n à gauche puis encore une fois pour faire passer à gauche m_{n-1} , etc jusqu'à m_{i+1}) donc le mot m s'écrit $m = uv$ et le mot produit est vu . La réciproque est immédiat (si σ a été appliqué k fois sur m , on le réapplique $|m| - k$ fois). Par conséquent, on a l'égalité ensembliste

$$\widehat{L} = \{uv, vu \in L, u, v \in X^*\}.$$

Soit $m \in L$ qui s'écrit $m = uv$, on note i un état initial de I , f un état final tel que l'on puisse lire m depuis l'état i jusqu'à l'état f . Si on pose q l'état obtenu par la lecture de u depuis i alors la lecture de v depuis l'état q aboutit à l'état f donc

$$u \in L_{i,q}, v \in L_{q,f} \Rightarrow m' = vu \in L_{q,f}L_{i,q}.$$

Comme

$$L = \bigcup_{i \in I, f \in F} L_{i,f} \Rightarrow \widehat{L} = \bigcup_{q \in Q} \bigcup_{i \in I, f \in F} L_{q,f}L_{i,q}$$

(l'inclusion réciproque est immédiate). Comme L est régulier, les langages $L_{q,q'}$ le sont tous (cf. la réponse à la question 8) donc $\widehat{L}$ aussi (les langages réguliers sont stables par concaténation et union).

Considérons le langage $L = \{a^pba^q, p \leq q, p, q \in \mathbb{N}\}$ qui n'est pas rationnel (cf. question 4) alors $\widehat{L} = a^*ba^*$ est régulier (on peut faire tourner les a partout et il reste toujours un b). Ainsi, la réciproque est fausse.

Commentaires : Question difficile destinée uniquement aux meilleures copies.

Q11.

```
let rec succ_list l x = match l with
| [] -> -1
| y::q when y <= x -> succ_list q x
| y::q -> y;;
```

Commentaires : Question sans difficulté.

Q12. Pour déterminer le maximum de E , on détermine son cardinal en lisant $E[0] = n$ puis on lit le contenu de $E[n]$ qui donne le plus grand élément de E (car le tableau est organisé par valeurs croissantes).

La complexité est constante donc en $O(1)$.

Pour déterminer l'appartenance de x à E , on peut utiliser la dichotomie car le tableau est trié par ordre croissant.

On fixe une fenêtre $[debut, fin]$ avec $debut = 0$ et $fin = n = E[0]$ (la taille de l'ensemble). Tant que la fenêtre contient au moins trois éléments (i.e. $fin - debut > 1$), poser $milieu = (debut + fin)/2$ (le milieu de la fenêtre). Si $x < E[milieu]$, $fin = milieu$ (la fenêtre de recherche est la moitié inférieure de celle ci car x se trouve avant le milieu) sinon $debut = milieu$ (x est dans la seconde moitié). fin tant que. Si $E[debut] = x$ ou $E[fin] = x$, renvoie *true*, sinon renvoyer *false*. La complexité est en

$$O(\log_2(n)) = O(\log_2(|E|))$$

(la longueur de la fenêtre est divisé par 2 à chaque étape donc par 2^N après N étapes et si $N = \log_2(n)$ alors après N étapes, la fenêtre est de taille 1 donc la boucle s'arrête nécessairement).

Pour insérer x à E , si $x > E[n]$, on insère x dans $E[n+1]$. Sinon, on insère le dernier élément $E[n]$ de E dans $E[n+1]$ (pour laisser une place à x). Si $x > E[n-1]$, on insère x dans $E[n]$. Sinon, on insère $E[n-1]$ dans $E[n]$, etc jusqu'à obtenir un élément $E[i]$ strictement inférieur à x . Si cela n'arrive jamais, nous avons décaler tous les éléments de E à gauche donc la cellule $E[1]$ est " libre " et on y insère x . C'est-à-dire, on fixe $fin = n$,

tant que $fin > 0$ et $E[fin] > x$, insérer $E[fin]$ dans $E[fin + 1]$, $fin \leftarrow fin - 1$ fin tant que, insérer x dans $E[fin + 1]$, $E[0] \leftarrow E[0] + 1$ (augmenter d'une unité l'ensemble E). La complexité est en $O(n) = O(|E|)$.

Commentaires : *Il s'agit potentiellement d'une question de cours mais elle s'avère en pratique sélective pour les candidats. N'hésitez pas à consolider vos connaissances de cours (algorithmes, structures, etc).*

Q13.

```
let succ_vect t x =
  let rec scan deb fin = match fin-deb with
    | 0 -> (-1)
    | 1 -> if t.(deb) > x then t.(deb) else if t.(fin) > x then
      t.(fin) else (-1)
    | _ -> if t.(deb) > x then t.(deb) else if t.(fin) <= x
      then (-1) else
      (let milieu = (deb+fin)/2 in if t.(milieu) > x then
        scan deb milieu else scan milieu fin)
  in scan 1 (t.(0));;
```

Commentaires : *Question de difficulté standard. Revoyez la méthode générale de dichotomie (notamment la recherche dichotomique) car il s'agit de cours ... mais cela s'avère toujours sélectif aux concours. Ici c'est le cas.*

Q14. La complexité est en $O(\log_2(|E|))$ (même raisonnement que pour la complexité de la recherche dichotomique dans la réponse à la question 12).

Commentaires : *Question quasiment de cours .. si le candidat à répondu à la question précédente. Revoyez la méthode générale de dichotomie (notamment la recherche dichotomique) car il s'agit de cours ... mais cela s'avère toujours sélectif aux concours.*

Q15.

```
let union_vect t1 t2 =
  let t=Array.make (Array.length t1) 0 and n1 = t1.(0) and n2
    = t2.(0) in
  let rec aux pos i1 i2 = match i1 <= n1 , i2 <= n2 with
    | true,false -> t.(pos)<-t1.(i1);aux (pos+1) (i1+1) i2
    | false,true -> t.(pos)<-t2.(i2) ; aux (pos+1) i1 (i2+1)
    | false,false -> t.(0)<-pos
    | true,true -> match t1.(i1)-t2.(i2) with
      | 0 -> t.(pos) <- t1.(i1) ; aux (pos+1) (i1+1) (i2+1)
      | d when d < 0 -> t.(pos)<-t1.(i1); aux (pos+1) (i1+1) i2
      | d -> t.(pos)<-t2.(i2); aux (pos+1) i1 (i2+1)
  in aux 1 1 1 ; t;;
```

Commentaires : *Question de difficulté standard.*

Q16.

```
let rec min_abr a = match a with
| Nil -> failwith "min_abr"
| Noeud(x,Nil,_) -> x
| Noeud(x,g,d) -> min_abr g;;
```

Commentaires : *Il s'agit d'une question de cours sans aucune difficulté.*

Q17.

```
let rec partitionne_abr a x = match a with
| Nil -> false, Nil, Nil
| Noeud(y,g,d) when y == x -> true,g,d
| Noeud(y,g,d) when x < y -> let (b,gg,gd) = partitionne_abr
    g x
    in b,gg,Noeud(y,gd,d)
| Noeud(y,g,d) -> let (b,dg,dd) = partitionne_abr d x in b,
    Noeud(y,g,dg) ,dd;;
```

Commentaires : *Question de difficulté standard. Il s'agit de renvoyer deux arbres binaires de recherche qui forment une partition de l'arbre initial. Il ne s'agit de simplement de former les fils gauche et droite du noeud où se trouve x . Le candidat doit réfléchir à la stratégie pour penser à reconstituer chaque morceaux. Au final, cette question s'avère très sélective.*

Q18. Pour l'insertion, il ne faut le faire comme de le cours car l'énoncé exige que x soit la racine de l'arbre. On partitionne l'arbre par rapport à x , on le place comme racine du nouvel arbre, son fils gauche est celui fournit par le partionnement et son fils droit celui fournit par le partionnement.

```
let insertion_abr a x =
  let b, g, d = partionne_abr a x in Noeud(x,g,d) ;;
```

La complexité de *partitionne_abr a x* est en $O(h)$, où h est la hauteur de l'arbre a (à chaque sous-arbre, on ne parcourt que l'un des deux fils donc on ne choisit que l'une des branches, donc le nombre maximal d'appels de *partitionne_abr* est h , les opérations, en dehors de ces appels, sont en nombre fini et de complexité constante).

Commentaires : *Il s'agit pour le candidat à penser à ne pas pratiquer l'insertion habituelle mais à s'adapter au contexte du sujet. Cela rend le codage sélectif. Par contre, la complexité est une question de cours qui ne doit pas vous poser de difficulté.*

Q19.

```
let rec union_abr a1 a2 = match a1 with
| Nil -> a2
| Noeud(x,g,d) -> union_abr g (union_abr d (insertion_abr a2
    x));;
```

Commentaires : Question de difficulté standard pour ce concours.

Q20. Le numéro peut varier entre 2^p (inclus) et 2^{p+1} (exclus) c'est-à-dire entre 2^p et $2^{p+1} - 1$. Soit i le numéro d'un noeud, et p_i sa profondeur p_i dans l'arbre depuis la racine alors

$$2^{p_i} \leq i < 2^{p_i+1} \Leftrightarrow p_i \leq \log_2(i) < p_i + 1 \Rightarrow p_i = \lfloor \log_2(i) \rfloor.$$

La profondeur de l'arbre dont i est la racine est alors $p - p_i$ donc, comme c'est encore un arbre binaire complet, son nombre de feuilles est $2^{p-p_i} = 2^{p-\lfloor \log_2(i) \rfloor}$.

Commentaires : Pour cette question, le sujet n'attend pas de preuve mais juste une réponse convenable. Le candidat peut se baser sur quelques dessins d'arbres de diverses profondeurs (1, 2, 3 et 4) pour observer que les numéros des noeuds varient entre 2^p et 2^{p+1} (exclus). Pour les feuilles, il peut donner comme réponse $p - p_i$ (elle sera probablement partiellement acceptée).

Q21. Soit p_i la profondeur de i . Le noeud le plus à gauche de i a pour numéro 2^{p_i} donc les noeuds entre celui le plus à gauche et i (exclus) sont au nombre de $m = i - 2^{p_i}$ (donc $i = 2^{p_i} + m$). Ils vont produire $2m$ noeuds à la profondeur suivante. Comme le noeud le plus à gauche de la profondeur suivante est 2^{p_i+1} (premier noeud) puis $2^{p_i+1} + 1$ (deuxième noeud) puis $2^{p_i+1} + 2$ (troisième noeud) etc jusqu'à $2^{p_i+1} + 2m - 1$ (le $(2m)^e$ noeud) donc le fils gauche de i aura pour numéro

$$2^{p_i+1} + 2m = 2(2^{p_i} + m) = 2i$$

donc son fils droit a pour numéro $2i + 1$. Le père de i pour numéro $\lfloor i/2 \rfloor$ (c'est le numéro k avec $i = 2k$ ou $i = 2k + 1$ selon que i soit un fils gauche ou droit donc k est le quotient de la division euclidienne de i par 2).

Commentaires : Si la réponse est classique (c'est du cours dans les arbres binaires tas), la réponse s'avère difficile pour les candidats. Au final, seules les meilleures copies donnent une preuve convenable.

Q22.

```
let appartient e i =
  let n=(Array.length e)/2 in e.(i+n);;
```

Elle est de complexité constante $O(1)$.

Commentaires : Question élémentaire.

Q23.

```
let rec insere l t n = match l with
| [] -> ()
| x::q -> t.(x+n)<-true; insere q t n;;

let rec rempli n t = match n with
| 0 -> ()
| n -> for i=n/2 to (2*(n/2)-1) do
```

```

        t.(i) <-t.(2*i)||t.(2*i+1) done;
    rempli (n/2) t;;

let fabrique l n =
    let t = Array.make (2*n) false in
    insere l t n ; rempli n t;t ;;

```

La fonction *insere l t n* a une complexité en $O(|l|)$ ($|l|$ appels d'elle même et les opérations hors ces appels ont un coût constant $O(1)$). La fonction *rempli t n* a une complexité en $O(n)$ (elle passe par chaque noeud une seule fois et les opérations ont un coût constant $O(1)$). La fonction *fabrique l n* appelle :

- *insere* sur une liste de taille au plus 2^p donc cet appel a un coût en $O(2^p)$;
- *rempli t n* avec $n = 2^p$ (d'après l'énoncé) donc cet appel a un coût en $O(2^p)$.

Finalement, *fabrique l n* a une complexité en $O(2^p) + O(2^p) = O(2^p)$.

Commentaires : *Question de difficulté standard. Si le candidat fait le lien avec la percolation dans les arbres tas (qui est au programme, notamment pour le tri par tas), il sera bien aidé dans le codage même si cela n'est pas indispensable. Pour la complexité, il n'est pas attendu du candidat une preuve mais juste qu'il s'assure que la complexité de sa fonction est de l'ordre de grandeur demandé par le sujet.*

Q24.

```

let insere t k =
    let n = (Array.length t)/2 in
    if t.(k+n) then () else
        t.(k+n) <- true;
    let i = ref ((k+n)/2) in
    while !i>0 && (not t.(!i)) do
        t.(!i)<-true ; i:=!i/2 done;;

```

Dans le meilleur des cas, si $t.(k+n)$ est faux (i.e. k est absent), la fonction s'arrête donc elle a une complexité $O(1)$ dans le meilleur des cas.

Commentaires : *Même remarque qu'à la question précédente.*

Q25.

```

let supprime t k =
    let n = (Array.length t)/2 in
    if not t.(k+n) then () else
        t.(k+n) <- false;
    let i = ref ((k+n)/2) in
    while !i>0 && t.(!i)&& not t.(2*(!i))
        && not t.(2*(!i)+1) do
        t.(!i)<-false ; i:=!i/2 done;;

```

Si $n = 2^p$, la boucle s'exécute au plus p fois. En effet, à chaque division i est divisé par 2 donc après p fois par 2^p . Comme i est initialement strictement inférieur à 2^p initialement, après p divisions au maximum, il sera strictement inférieur à 1 et comme c'est un entier, il vaudra 0 et la boucle s'arrête. Les coûts dans l'itération de la boucle ou dans le corps de la boucle sont en nombre fini et de coût constant donc en $O(1)$. Au final, la complexité totale est en

$$O(p)O(1) = O(p).$$

Commentaires : *Même remarque que précédemment.*

Q26.

```
let rec minlocal t i =
  if not t.(i) then (-1) else
    let n = (Array.length t)/2 in
    if i >= n then i-n else
      (if t.(2*i) then minlocal t (2*i) else
        minlocal t (2*i+1));;
```

Si $n = 2^p$ est la taille maximale de l'ensemble, le nombre maximum d'appels *minlocal* par *minlocal t i* est au plus $p - p_i$ (p_i la profondeur de i i.e $p_i = \lfloor \log_2(i) \rfloor$ d'après la question 21). En effet, à chaque appel de *minlocal*, l'entrée i est au moins multiplié par 2 donc après $p - p_i$ appels, elle est multipliée par 2^{p-p_i} . Comme i est au moins égal à 2^{p_i} (il est en profondeur p_i), après $p - p_i$ multiplication au maximum, elle sera supérieure à 2^p et les appels de *minlocal* s'arrête. Les coûts, hors appels de *minlocal*, sont en nombre fini et de coût constant donc en $O(1)$. Au final, la complexité totale est en

$$O(p - p_i)O(1) = O(p - p_i) = O(p - \log_2(i)).$$

Commentaires : *Pour la programmation, question de difficulté standard. La liaison avec la navigation dans les arbres tas aidera probablement grandement le candidat. Pour la complexité, il s'agit de cours que d'affirmer qu'elle est en $O(h)$ donc en $O(p)$. Il est certainement attendu ici que le candidat tienne compte de la profondeur local (c'est-à-dire celle de i).*

Q27. Notons $pos = i$ la position courante du noeud que va parcourir l'algorithme dans l'arbre (qui sera la racine du sous-arbre le plus petit contenant à la fois i et son successeur). Supposons que l'on soit à une itération donnée de la boucle.

Supposons que pos (i tel qu'écrit dans la boucle) ne soit pas le noeud le plus à droite de sa profondeur et que $pos + 1$ vaut *false*. Cela signifie que le sous-arbre de racine $pos+1$ juste à droite de pos ne contient aucun élément de l'ensemble E . En particulier, le successeur de i ne peut s'y trouver. Il faut remonter d'un niveau dans l'arbre. Il est immédiat alors que l'affirmation « si le test booléen de la boucle est vrai, pos possède un sous-arbre juste à sa droite qui ne contient aucun élément de E après i » est vérifiée.

Lorsque la boucle s'arrête, nous distinguons deux cas.

Premier cas. pos est le noeud le plus à droite de sa profondeur donc on n'a jamais rencontré de sous-arbres pouvant contenir un élément de E après i donc i n'a pas de successeur.

Deuxième cas. Le noeud $pos+1$ vaut *true* donc pos possède un sous-arbre juste à sa droite contenant au moins un élément de E après i . Comme le fils de pos ne vérifiait pas cette

propriété, c'est que l'arbre de racine pos ne possède aucun élément de E après i . Par conséquent, le plus petit élément de l'arbre de racine $pos+1$ est le premier élément de E après i c'est-à-dire le successeur de i .

Par conséquent, nous venons de prouver la correction de l'algorithme.

Commentaires : Les questions de correction de programme ou d'algorithme sont souvent très difficile pour les candidats. Cela ne fait pas exception ici. Cette question est destinée aux meilleures copies.

Q28. Il s'agit de remonter dans l'arbre depuis x jusqu'à aboutir à un noeud possédant un voisin ayant des descendants. Dans ce cas, le plus petit de ces descendants est le successeur de x .

```
let successeur t x =
  let rec aux deb fin = match deb < fin with
    | false -> (-1)
    | true -> if not t.(deb+1) then aux (deb/2) (fin/2)
               else deb+1
  in let n = (Array.length t)/2
  in let i = aux (x+n) (2*n-1)
  in if i >= 0 then minlocal t i else (-1) ;;
```

Commentaires : Question très sélective qui demande au cas un recul certain et une maîtrise dans la programmation.

Q29. Notons x' le successeur de x alors $x' - x$ est la distance entre x et son successeur. Notons x_1 et x'_1 les pères respectifs de x et x' alors la distance entre x_1 et x'_1 est au maximum de $\frac{x'_1 - x_1}{2}$ (car x_1 et x'_1 valent moins de la moitié de x et x'). En itérant cela k fois, la distance entre x_k (père de x_{k-1}) et x'_k (père de x'_{k-1}) est au maximum de $\frac{x'_1 - x_1}{2^k}$. Dès que cet écart est strictement inférieur à 1, x_k et x'_k sont égaux donc on ne peut plus remonter dans l'arbre, on a détecté le bon sous-arbre. Il faudra alors k descentes pour accéder au minimum. Il y a aura eu alors $2k$ mouvements dans l'arbre (remontée pour accéder aux pères et descentes pour accéder aux bons fils). Comme

$$\frac{x'_1 - x_1}{2^k} < 1 \Leftrightarrow x'_1 - x_1 < 2^k \Leftrightarrow k > \log_2(x'_1 - x_1) \Leftrightarrow k \geq \lfloor \log_2(x'_1 - x_1) \rfloor + 1,$$

le nombre maximum de mouvement sera

$$2 \lfloor \log_2(x'_1 - x_1) \rfloor + 1 = 2 \lfloor \log_2(\text{successeur } x - x) \rfloor + 2$$

Comme les opérations, hors mouvement, sont en nombre fini et de coût constant, la complexité total en

$$O(\log_2(\text{successeur } x - x) + 2)$$

Commentaires : Question très difficile. Seules les meilleures copies fourniront une réponse convenable.

Q30.

```

let cardinal t =
  let rec aux i acc = match successeur t i with
    | (-1) -> acc
    | i' -> aux i' (acc+1)
  in aux (minlocal t 1) 0;;

```

Commentaires : Question de difficulté standard.

Q31. Notons $E = \{x_1, \dots, x_n\}$. Il faut chercher le premier élément x_1 de E via *minlocal t 1* qui a un coût en $O(\log_2(2^p)) = O(p)$. Ensuite, pour chaque i , il faut chercher x_i connaissant x_{i-1} via la fonction *successeur* donc la complexité est en $O(\log_2(x_i - x_{i-1}) + 2)$. Comme

$$x_i - x_{i-1} \leq x_i \leq 2^p \Rightarrow \log_2(x_i - x_{i-1}) \leq \log_2(2^p) = p.$$

Comme la fonction *successeur* est appelé au plus n fois, la complexité totale est en $nO(p + 2) = O(np)$.

Remarque. Si on utilise la concavité du log, ... on obtient rien de plus hormis un calcul inaccessible, je crains, pour tous les candidats à ce concours! La complexité totale est

$$\begin{aligned}
 & O(p) + \sum_{i=2}^n \log_2((x_i - x_{i-1}) + 2) \\
 &= O\left(p + 2n + \sum_{i=2}^n \log_2(x_i - x_{i-1})\right).
 \end{aligned}$$

Comme $\log_2$ est concave, l'inégalité de convexité (avec $y_i = x_i - x_{i-1}$ et $\lambda_i = \frac{1}{n-1}$) nous donne

$$\begin{aligned}
 \frac{1}{n-1} \sum_{i=2}^n \log_2(x_i - x_{i-1}) &\leq \log_2\left(\frac{1}{n-1} \sum_{i=2}^n (x_i - x_{i-1})\right) \\
 &= \log_2\left(\frac{x_n - x_1}{n-1}\right) \leq \log_2(x_n) \leq \log_2(2^p) = p
 \end{aligned}$$

(car $x_m \leq 2^p$) donc

$$\sum_{i=2}^n \log_2(x_i - x_{i-1}) \leq (n-1)p = O(np).$$

Par conséquent, la complexité totale est en

$$O(p + 2n + np) = O(np)$$

donc ... on a rien gagné par rapport au calcul "à la louche"!

Commentaires : Question très difficile uniquement accessible aux meilleures copies si le candidat s'évertue à utiliser la concavité du log! Par contre, s'il fait un calcul "à la louche", la difficulté est standard pour ce concours. Comme quoi, une aide peut être largement contre-productive ...

Q32. Faisons un tableau de complexité des opérations pour un ensemble E pour les listes, les tableaux et la structure du sujet (arbres binaires complets formés de booléens).

tb : tableau de booléens, lt : liste triée, tt : tableau trié, abr : arbres binaires de recherche ,
s : structure du sujet

	tb	lt	tt
maximum	$O(E)$	$O(E)$	$O(1)$
minimum	$O(E)$	$O(1)$	$O(1)$
appartenance	$O(1)$	$O(E)$	$O(\log_2(E))$
insertion	$O(1)$	$O(E)$	$O(E)$
successeur	$O(E)$	$O(E)$	$O(\log_2(E))$

	abr	s
maximum	$O(h)$ (entre $O(\log_2(E))$ et $O(E)$)	$O(\log_2(E))$
minimum	$O(h)$ (entre $O(\log_2(E))$ et $O(E)$)	$O(\log_2(E))$
appartenance	$O(h)$ (entre $O(\log_2(E))$ et $O(E)$)	$O(\log_2(E))$
insertion	$O(h)$ (entre $O(\log_2(E))$ et $O(E)$)	$O(\log_2(E))$ ou $O(E)$
successeur	$O(h)$ (entre $O(\log_2(E))$ et $O(E)$)	$O(\log_2(\text{successeur } x - x))$

On observe que cette structure est bonne (par rapport aux autres habituelles) pour la plupart des opérations (sauf maximum ou minimum face aux tableaux triés). Pour l'insertion, elle est compétitive : si $|E| < 2^p$, il faut $O(\log_2(E))$ opérations pour insérer (descendre l'arbre pour aller au bon emplacement puis remonter pour mettre à jour les étiquettes des parents successifs). Par contre, si $|E| = 2^p$ et qu'on insère un élément x , il faut recréer une structure de taille $2^{p+1} = 2|E|$ en y insérant l'ensemble E donc un coût en $O(|E|)$. Néanmoins, cela n'arrivera plus durant 2^p insertions donc, en moyenne (sur 2^p insertions consécutives), chaque insertion coûtera $\frac{O(|E|)}{|E|} = O(1)$ donc cette structure est très compétitive en moyenne en fait !

Pour le successeur, elle est la plus compétitive car le coût est au plus en $O(\log_2(|E|))$ si les éléments sont suffisamment éloignés les uns des autres. Par contre, si les éléments sont assez rapprochés, l'écart entre deux éléments successifs est borné donc la recherche du successeur est aussi bornée donc d'un coût quasiment constant !

Commentaires : Question difficile car le candidat doit avoir répondu un nombre significatif de questions ET connaître l'implémentation via différentes structures et leurs complexités.

Q33. $E = \{0, 2, 3, 5, 7, 13, 14\}$ et $N = 16 = 2^4 = 2^{2^2}$ donc $p = 2$. On a :

- $\sqrt{N} = 4$, $\min(E) = 0$, $\hat{E} = E \setminus \min(E) = \{2, 3, 5, 7, 13, 14\}$;
- $\hat{E} \cap \left[\left[0\sqrt{N}, 1\sqrt{N} - 1 \right] \right] = \{2, 3\}$, on décale tous ses éléments de $0\sqrt{N} = 0$ donc $E_0 = \{2, 3\}$;
- $\hat{E} \cap \left[\left[\sqrt{N}, 2\sqrt{N} - 1 \right] \right] = \{5, 7\}$, on décale tous ses éléments de $1\sqrt{N} = 4$ donc $E_1 = \{1, 3\}$;
- $\hat{E} \cap \left[\left[2\sqrt{N}, 3\sqrt{N} - 1 \right] \right] = \emptyset$ donc $E_2 = \emptyset$;
- $\hat{E} \cap \left[\left[3\sqrt{N}, 4\sqrt{N} - 1 \right] \right] = \{13, 14\}$, on décale tous ses éléments de $3\sqrt{N} = 12$ donc $E_3 = \{1, 2\}$;

- $R = \{0, 1, 3\}$ (les entiers k tels que $E_k \neq \emptyset$).

$veb.(3)$ code l'ensemble $F = E_3 = \{1, 2\}$ avec $N = 2^{2^1} = 4$ ($E \subset [0, 2^{2^p}[$). Son minimum est 1, son maximum est 2. Le code informatique est $veb.(3) = \{mini = 1; maxi = 2; table = tf\}$

```
veb.(3) <- {mini = 1 ; maxi = 2 ; table = tf }
```

avec tf le veb modélisant l'ensemble F . On le stratifie comme suit :

- $\sqrt{N} = 2$, $\min(F) = 1$, $\hat{F} = F \setminus \min(F) = \{2\}$;
- $\hat{F} \cap [0\sqrt{N}, 1\sqrt{N} - 1] = \emptyset$ donc $F_0 = \emptyset$;
- $\hat{F} \cap [1\sqrt{N}, 2\sqrt{N} - 1] = \{2\}$, on décale tous ses éléments de $1\sqrt{N} = 2$ donc $F_1 = \{0\}$;
- $R = \{1\}$ (unique k tel que $F_k \neq \emptyset$).

Rappelons que les ensembles vide, $\{0\}$ et $\{1\}$ sont modélisés par les veb suivants (d'après le sujet) :

```
let vide = { mini = -1 ; maxi = -1 ; table = [||]} ;;
let ens0 = { mini = 0 ; maxi = 0 ; table = [|]} ;;
let ens1 = { mini = 1 ; maxi = 1 ; table = [|]} ;;
```

Par conséquent, le veb tf modélisant l'ensemble $\{1, 2\}$ est

```
let tf = { mini = 1 ; maxi = 2 ; table = [ | vide ; ens0 ;
      ens1 | ] };;
```

Commentaires : Question sélective mais accessible si le candidat se plonge dans les définitions et les mets en place dans cet exemple concret et relativement simple. Le but est de lui permettre de s'accaparer la notion. N'hésitez pas à proposer un dessin (comme l'exemple du sujet). Si vous parvenez à écrire le code, tant mieux, sinon le correcteur pourra constater si vous avez compris le processus. Dans ce cas, il risque d'être compréhensif car il s'agit de la dernière partie du sujet et que la notion est complexe.

Q34.

```
let rec creer_veb p = match p with
| 0 -> {mini=(-1); maxi=(-1); table= [||]}
| n -> let m = exp_exp (p-1) in
      let t = creer_veb (p-1) in
      let tab = Array.make (m+1) t in
      {mini=(-1); maxi=(-1); table = tab};;
```

Commentaires : Question de programmation de difficulté standard ... si le candidat a intégré la notion de veb (ce qui n'est pas sûr, vu qu'il est probablement fatigué par les autres questions du sujet).

Q35. On pose $q = 2^{2^p}$ alors

$$\sqrt{q} = \left(2^{2^p}\right)^{1/2} = 2^{2^p \times 1/2} = 2^{p-1}$$

donc

$$C(q) = C(\sqrt{q}) + O(1) \Leftrightarrow C(2^{2^p}) = C(2^{2^{p-1}}) + O(1).$$

Si on pose $u_p = C(2^{2^p})$, on a

$$u_p = u_{p-1} + O(1) \xrightarrow[\text{arithmétique}]{\text{suite}} u_p = pO(1) + u_0 = O(p).$$

Comme $q = 2^{2^p}$ alors $\log_2(q) = 2^p$ donc $p = \log_2(\log_2(q))$ d'où

$$C(q) = O(p) = O(\log_2(\log_2(q))).$$

Commentaires : Question destinée aux meilleures copies.

Q36.

```
let rec appartient_veb v x =
  if v.mini > x || v.maxi < x then false
  else
    if v.mini == x || v.maxi == x then true
    else
      let tab = v.table
      in let m = (Array.length tab)-1
      in let k = x/m in
      appartient_veb tab.(k) x;;
```

Le veb v en entrée modélisé un ensemble de taille N et c'est un tableau de taille $m = \sqrt{N} + 1$ dont les cellules sont des veb modélisant des ensembles de taille $\sqrt{N}$. La fonction `appartient_veb v x` appelle `appartient` un certain veb v_1 (`tab.(k)`) de taille

$$\frac{N}{m} \leq \frac{N}{\sqrt{N}} = \sqrt{N} = N^{1/2}.$$

On note $C(N)$ la complexité maximale de l'appel `appartient_veb v x` sur tout veb v de taille au plus N . L'appel `appartient_veb tab.(k) x` a donc une complexité au plus de $C(\sqrt{N})$. Les autres opérations sont en nombre borné et de coût constant donc en $O(1)$. Ainsi, $C(N) = C(\sqrt{N}) + O(1)$. D'après la question précédente, on a

$$C(N) = O(\log_2(\log_2(N))).$$

(on peut utiliser proprement la question précédente, en choisissant l'unique entier p tel que $2^{2^{p-1}} < N \leq 2^{2^p}$ alors $C(N) \leq C(2^{2^p})$ (car $N \mapsto C(N)$ est croissante car c'est le **maximum** de la complexité pour **toutes** les entrées de taille au plus N). D'après la question précédente, on a $C(2^{2^p}) = O(p)$ et comme

$$p = \lfloor \log_2(\log_2(N)) \rfloor = O(\log_2(\log_2(N))),$$

on a bien

$$C(N) \leq C(2^{2^p}) = O(p) = O(\log_2(\log_2(N))).$$

Commentaires : *Même remarque que précédemment. (même si le codage informatique est accessible aux bonnes copies, encore faut-il qu'elles aient eu le temps d'arriver à cette question, c'est tout le problème certainement).*

Q37.

```
let rec successeur_veb v x =
  if x < v.mini then v.mini
  else
    if v.mini > x || v.maxi < x then (-1)
    else
      let tab = v.table
      in let m = (Array.length tab)-1
      in let k = x/m in
      let s = successeur_veb tab.(k) (x-k*m)
      in match s with
      | (-1) -> let ks = successeur_veb tab.(m) k in
        if ks < 0 then (-1) else tab.(ks).mini+ks*m
      | s -> s+k*m;;
```

Par le même argumentaire qu'à la question précédente, sa complexité est en $O(\log_2(\log_2(N)))$.

Commentaires : *Même remarque que précédemment.*

Q38.

```
let rec insertion_veb v x =
  if v.mini == (-1) then v.mini <-x
  else
    if v.mini == x || v.maxi == x then ()
    else
      let tab = v.table in
      let m = (Array.length tab)-1 in
      if m < 0 then
        if x < v.mini (*on est dans un ensemble à 2 éléments *)
        then v.mini <-x
        else
          let x' = ref x in
          if x < v.mini then (x' := v.mini; v.mini <-x);
          if x > v.maxi then (x' := v.maxi; v.maxi <-x);
          let k = !x'/m in
          if tab.(k).mini == (-1) then insertion_veb tab.(m) k;
          insertion_veb tab.(k) (!x' - k*m);;
```

Par le même argumentaire qu'à la question précédente, sa complexité est en $O(\log_2(\log_2(N)))$.

Commentaires : *Question destinée aux meilleures copies.*

Q39. Un veb modélisant en ensemble de taille N est segmenté en $\sqrt{N} + 1$ veb modélisant des ensembles de taille $\sqrt{N}$ (stockage de $M(\sqrt{N})$) auquel il faut ajouté le stockage du minimum et maximum donc

$$M(N) = (\sqrt{N} + 1) M(\sqrt{N}) + 2.$$

Comme $N = 2^{2^p}$ et $\sqrt{N} = 2^{2^{p-1}}$, si on pose $u_p = M(N) = M(2^{2^p})$, on a

$$u_p = (2^{2^{p-1}} + 1) u_{p-1} + 2$$

Il est immédiat que $u_p \geq 2$ (quelque soit p) et comme $2 \leq 2^{2^{p-1}}$, on a :

$$u_p \leq (2^{2^{p-1}} + 1) u_{p-1} + u_{p-1} = (2^{2^{p-1}} + 2) u_{p-1} \leq 2 \times 2^{2^{p-1}} u_{p-1}.$$

En itérant cette inégalité, on obtient

$$\begin{aligned} u_p &\leq 2 \times 2^{2^{p-1}} u_{p-1} \leq 2 \times 2^{2^{p-1}} \times 2 \times 2^{2^{p-2}} u_{p-2} = 2^2 2^{2^{p-1} + 2^{p-2}} u_{p-2} \\ &\leq 2^2 2^{2^{p-1} + 2^{p-2}} 2 \times 2^{2^{p-3}} u_{p-3} = 2^3 2^{2^{p-1} + 2^{p-2} + 2^{p-3}} u_{p-3} \leq \dots \\ &\leq 2^p 2^{2^{p-1} + 2^{p-2} + 2^{p-3} + \dots + 1} u_0. \end{aligned}$$

Comme d'après la fameuse somme géométrique, on a

$$2^{p-1} + 2^{p-2} + 2^{p-3} + \dots + 1 = \frac{2^p - 1}{2 - 1} = 2^p - 1 \leq 2^p$$

donc

$$u_p \leq 2^p 2^{2^p} u_0 \Leftrightarrow M(2^{2^p}) \leq 2^p 2^{2^p} u_0$$

Comme $N = 2^{2^p}$ alors $2^p = \log_2(N)$ donc

$$M(N) = M(2^{2^p}) \leq N \log_2(N) \Rightarrow M(N) = O(N \log_2(N)).$$

Une analyse mathématique plus fine (mais honnêtement inaccessible à un(e) étudiant(e) de CPGE non guidé) permet de montrer que $M(N) = O(N)$.

Commentaires : Bravo aux candidat(e)s ayant réussi l'exploit à traiter à peu près convenablement cette question. Ils sont très certainement rares ! (le pluriel est probablement de trop).



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2023

ÉPREUVE D'INFORMATIQUE COMMUNE

Durée de l'épreuve : 2 heures

L'usage de la calculatrice ou de tout dispositif électronique est interdit.

Cette épreuve est commune aux candidats des filières MP, PC et PSI.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE COMMUNE

L'énoncé de cette épreuve comporte 8 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



La typographie informatisée

Ce sujet explore quelques aspects de la typographie informatisée. Il aborde la gestion de polices vectorielles, leur manipulation, leur tracé, l’affichage de texte et la justification d’un paragraphe. Il va du texte à la page via le pixel. Les questions posées peuvent dépendre des questions précédentes. Toutefois, une question peut être abordée en supposant les fonctions précédentes disponibles, même si elles n’ont pas été implémentées. Les questions de programmation seront traitées en Python.

Partie I – Préambule

La typographie est l’art d’assembler des caractères afin de composer des pages en vue de leur impression ou de leur affichage sur un écran, en respectant des règles visuelles qui rendent un texte agréable à lire. Elle requiert des efforts importants, avantageusement simplifiés par le recours à l’outil informatique.

Donald Knuth, prix Turing 1974 notamment pour la monographie *The Art of Computer Programming*, en est un pionnier. Lassé de la piètre qualité de la typographie proposées pour son ouvrage, il développe les logiciels $\text{\TeX}$ pour la mise en page et $\text{\METAFONT}$ pour la gestion de polices. Leslie Lamport, prix Turing 2013 pour ses travaux sur les systèmes distribués, a écrit $\text{\LaTeX}$ qui facilite l’utilisation de $\text{\TeX}$. $\text{\LaTeX}$ est largement utilisé dans l’édition scientifique, y compris pour la composition du présent document.

Donald Knuth récompense toute personne qui signale une nouvelle erreur dans un de ses ouvrages par un chèque d’un montant de un hexa dollar, c’est-à-dire 100 cents où 100 est interprété en base hexadécimale (base 16).

❑ **Q1** Quel montant est effectivement versé en dollars par Donald Knuth pour une nouvelle erreur trouvée ?

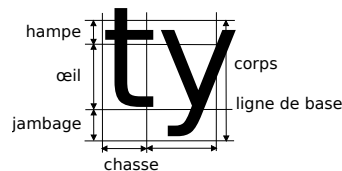
Voici la définition de quelques termes utiles pour la suite :

- un **caractère** est un signe graphique d’un système d’écriture, par exemple le *caractère latin a majuscule* « A ». Le standard Unicode donne à chaque caractère un nom et un identifiant numérique, appelé *point de code*, que nous appellerons ci-après simplement *code*. Le code de « A » dans la représentation Unicode est 65. La version 13.0 publiée en mars 2020 répertorie 143 859 caractères couvrant 154 systèmes d’écriture, modernes ou historiques comme les hiéroglyphes ;
- un **glyphe** est un dessin particulier représentant un caractère, par exemple pour le *caractère latin a majuscule* : A (roman) A (italique) A (caligraphié) A (gras), A (courrier)...
- une **police** de caractères est un ensemble coordonné de glyphes incluant différentes variantes (style roman ou italique, grasse...) et permettant de représenter un texte complet dans un système d’écriture. La police de ce document est *Computer Modern*, la police par défaut de $\text{\LaTeX}$;
- une **famille** est un groupe de polices. La classification Vox-ATypI, proposée par Maximilien Vox en 1952 et adoptée par l’Association typographique internationale, contient 11 familles. La police *Computer Modern* fait partie de la famille *Didone*.

Le corps du glyphe est sa hauteur, la chasse est sa largeur. Le corps est décomposé en trois parties : l’œil qui contient typiquement les petites lettres, le jambage et la hampe qui recouvrent les dépassements en dessous ou au dessus de l’œil. La limite inférieure de l’œil est la ligne de base. Elle définit l’alignement des caractères. La chasse peut être fixe (polices monospaces) ou variable.

La description vectorielle d’un glyphe est définie de la façon suivante :

- un **point** **p** est repéré par ses coordonnées (abscisse, ordonnée) dans le plan orthonormé classique, et sera représenté par une liste de deux flottants ;
- une **multi-ligne** **l** est une séquence de points reliés par des segments, représentée par une liste de points, éventuellement restreinte à un seul point ;



- la **description vectorielle** v d'un glyphe est un ensemble non vide de multi-lignes, représenté par une liste de multi-lignes.

Les descriptions vectorielles seront supposées normalisées de sorte que la ligne de base corresponde à l'ordonnée 0, que la hauteur de l'œil soit 1, et enfin que le glyphe soit collé à l'abscisse 0, sans dépassement vers les abscisses négatives.

Concrètement, la description vectorielle d'un glyphe est une liste de listes de listes de 2 flottants. À titre d'illustration, voici une description vectorielle d'un glyphe, composée de deux multi-lignes .

$|v = [[[0.25, 1.0], [0.25, -1.0], [0.0, -1.0]], [[0.25, 1.25]]]$

- **Q2** Dessiner ce glyphe. De quel caractère s'agit-il ?

Partie II – Gestion de polices de caractères vectorielles

Une base de données stocke les informations liées aux polices de caractères dans 4 tables ou relations.

Famille décrit les familles de polices, avec **fid** la clé primaire entière et **fnom** leur nom.

Police décrit les polices de caractères disponibles, avec **pid** la clé primaire entière, **pnom** le nom de la police et **fid** de numéro de sa famille.

Caractere décrit les caractères, avec **code** la clé primaire entière, **car** le caractère lui-même, **cnom** le nom du caractère.

Glyphe décrit les glyphes disponibles, avec **gid** la clé primaire entière, **code** le code du caractère correspondant au glyphe, **pid** le numéro de la police à laquelle le glyphe appartient, **groman** un booléen vrai pour du roman et faux pour de l'italique et **gdesc** la description vectorielle du glyphe.

Voici un extrait du contenu de ces tables.

Famille		Police			Caractere		
fid	fnom	pid	pnom	fid	code	car	cnom
1	Humane	1	Centaur	1	65	A	lettre majuscule latine a
2	Garalde	2	Garamond	2	66	B	lettre majuscule latine b
3	Réale	3	Times New Roman	3	...	...	...
4	Didone	4	Computer Modern	4	97	a	lettre minuscule latine a
5	Mécane	...	...	...	98	b	lettre minuscule latine b
6	Linéale	21	Triangle	6	99	c	lettre minuscule latine c
...	...	...	...	...	...	...	...

Glyphe				
gid	code	pid	groman	gdesc
1	65	20	True	[[[0, 0], [1, 2], [2, 0]], [[0.5, 1], [1.5, 1]]]
2	65	20	False	[[[0, 0], [2, 2], [2, 0]], [[1, 1], [2, 1]]]
...	...	...	...	...
501	97	21	True	[[[0, 0], [0.5, 1], [1, 0], [0, 0]]]
502	98	21	True	[[[0, 2], [0, 0], [1, 0.5], [0, 1]]]
503	99	21	True	[[[1, 1], [0, 0.5], [1, 0]]]
504	100	21	True	[[[1, 2], [1, 0], [0, 0.5], [1, 1]]]
...	...	...	...	...

- **Q3** Proposer une requête en SQL sur cette base de données pour compter le nombre de glyphes en roman (cf. description précédente).

- **Q4** Proposer une requête en SQL afin d'extraire la description vectorielle du caractère *A* dans la police nommée Helvetica en italique.

❑ **Q5** Proposer une requête en SQL pour extraire les noms des familles qui disposent de polices et leur nombre de polices, classés par ordre alphabétique.

Pour la suite, la requête de la question 4 est supposée paramétrée et encapsulée dans une fonction `glyphe(c, p, r)` qui renvoie la description vectorielle du caractère `c` dans la police `p` en roman ou italique selon le booléen `r`, de sorte que l'appel à `glyphe("a", "Helvetica", False)` répond à la question 4.

Partie III – Manipulation de descriptions vectorielles de glyphes

L'avantage de la description vectorielle de glyphes est qu'il est possible de réaliser des opérations sur les glyphes sans perte d'information. On peut réaliser simplement un agrandissement des glyphes, une déformation de glyphe pour en créer un nouveau etc. Cette partie propose des fonctions pour analyser et modifier des descriptions vectorielles.

Dans un premier temps, des fonctions sont créées pour extraire des informations sur des glyphes. Deux fonctions utilitaires sont implémentées.

❑ **Q6** Implémenter la fonction utilitaire `points(v:[[[float]])->[float]` qui renvoie la liste des points qui apparaissent dans les multi-lignes de la description vectorielle `v` d'un glyphe.

```
v = [ [ [ 0, 0 ], [ 1, 1 ] ], [ [ 0, 1 ], [ 1, 0 ] ] ]
print(points(v))           # affiche la liste [ [ 0, 0 ], [ 1, 1 ], [ 0, 1 ], [ 1, 0 ] ]
```

❑ **Q7** Implémenter la fonction utilitaire `dim(l:[[float]], n:int)->[float]` qui renvoie la liste des éléments d'indice `n` (en commençant à 0) des sous listes de flottants, dont on supposera qu'ils existent toujours.

```
l = [ [ 1, 2 ], [ 3, 4 ], [ 5, 6 ], [ 7, 8 ] ]
print(dim(l, 1))           # affiche la liste [ 2, 4, 6, 8 ]
```

On cherche à déterminer les dimensions (largeur et hauteur) d'un glyphe donné de manière à pouvoir les modifier par la suite si nécessaire.

❑ **Q8** Implémenter la fonction `largeur(v:[[[float]])->float` qui renvoie la largeur de la description vectorielle `v`. Il faudra utiliser les fonctions utilitaires précédentes ainsi que les fonctions `max` et `min` de Python appliquées à des listes.

❑ **Q9** Implémenter la fonction `obtention_largeur(police:str)->[float]` qui renvoie une liste de largeurs pour toutes les lettres minuscules romanes et italiques (uniquement les 26 lettres non accentuées de `a` à `z`) de la police `police` dans l'ordre `a` roman, `a` italique, `b` roman, `b` italique...

On souhaite dériver automatiquement de nouvelles représentations vectorielles de glyphes à partir de représentations existantes.

Python permet de passer simplement des fonctions en paramètre d'autres fonctions. Par exemple, la fonction `applique` ci-après renvoie une nouvelle liste constituée en appliquant la fonction `f` à tous les éléments de la liste `l`.

```
def applique(f:callable, l:[])->[]:
    return [ f(i) for i in l ]

def incremente(i:int)->int:
    return i + 1

print(applique(incremente, [ 0, 5, 8 ]))   # affiche la liste [ 1, 6, 9 ]
```

❑ **Q10** En se basant sur l'exemple de la fonction `applique`, implémenter une fonction utilitaire `transforme(f:callable, v:[[[float]])->[[[float]]]` qui prend en paramètres une fonction `f`, une description vectorielle `v` et qui renvoie une nouvelle description vectorielle construite à partir de `v` en appliquant la fonction `f` à chacun des points et en préservant la structure des multi-lignes. La fonction `f` passée en argument transforme un point en un autre point.

Soit la fonction `zzz` qui renvoie un nouveau point calculé de la façon suivante :

```
def zzz(p:[float])->[float]:
    return [ 0.5 * p[0], p[1] ]
```

□ **Q11** Expliquer comment est modifiée une description vectorielle v par `transforme(zzz, v)`. Préciser l'effet obtenu sur un glyphe.

□ **Q12** Implémenter la fonction `penche(v: [[float]])->[[float]]` qui renvoie une nouvelle description vectorielle correspondant à un glyphe penché vers la droite, obtenue en modifiant comme suit les coordonnées des points (x, y) :

- la nouvelle abscisse est $x + 0.5 * y$;
- la nouvelle ordonnée reste y .

Partie IV – Rasterisation

La rasterisation est la transformation d'une image vectorielle en image matricielle. Cette opération est indispensable notamment pour afficher à l'écran une image vectorielle.

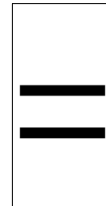
Dans cette partie, il s'agit d'analyser comment représenter le segment entre deux points d'une représentation vectorielle par des pixels encrés dans une image bitmap.

Le module PIL (Python Image Library) fournit le sous module `Image` :

- `im = Image.new(mode, size, color=0)` alloue une nouvelle image matricielle, de type bitmap si `mode` vaut "1"; le tuple `size` donne la largeur et la hauteur de l'image en pixels; le paramètre facultatif `color` précise la couleur par défaut des pixels, en bitmap 1 pour blanc et 0 pour noir;
- `im.putpixel((x, y), 1)` attribue la valeur 1 au pixel de coordonnées (x, y) de l'image `im`;
- `im.save(nom_fichier)` sauvegarde l'image dans un fichier dont on donne le nom;
- `im.show()` affiche l'image dans une fenêtre graphique.

Attention, dans le domaine graphique (images, impressions, écrans), et contrairement aux conventions mathématiques usuelles, le pixel $(0, 0)$ est le coin en haut à gauche de l'image, l'axe des ordonnées est dirigé vers le bas. Ainsi, le code suivant crée une image bitmap rectangulaire 50×100 blanche avec deux barres horizontales formant un signe « = ».

```
1 from PIL import Image
2
3 im = Image.new("1", (50, 100), color=1)
4 for y in range(60, 65):
5     for x in range(5, 45):
6         im.putpixel((x, y), 0)
7         im.putpixel((x, y-20), 0)
8 im.save("egal.png")
```



La fonction `trace_quadrant_est` implémente une partie de l'algorithme de tracé continu de segment proposé par Jack E. Bresenham en 1962.

```
1 from PIL import Image
2 from math import floor # renvoie l'entier immédiatement inférieur
3
4 def trace_quadrant_est(im:img, p0:(int), p1:(int)):
5     x0, y0 = p0
6     x1, y1 = p1
7     dx, dy = x1-x0, y1-y0
8     im.putpixel(p0, 0)
9     for i in range(1, dx):
10        p = (x0 + i, y0 + floor(0.5 + dy * i / dx))
11        im.putpixel(p, 0)
12    im.putpixel(p1, 0)
13
14 im = Image.new("1", (10, 10), color=1)
15 trace_quadrant_est(im, (0, 0), (6, 2))
16 trace_quadrant_est(im, (9, 8), (1, 9))
17 trace_quadrant_est(im, (3, 0), (5, 8))
18 im.show()
```


❑ **Q13** Préciser les coordonnées des pixels de l'image encrés (pixels que l'on met en noir) par l'exécution de la ligne 15 ?

❑ **Q14** Préciser les coordonnées des pixels de l'image encrés par l'exécution de la ligne 16 ? Indiquer d'où vient le problème rencontré. Proposer une assertion à mettre en début de fonction pour éviter ce problème.

❑ **Q15** Préciser les coordonnées des pixels de l'image encrés par l'exécution de la ligne 17 ? Expliquer le problème rencontré et à quoi il est dû.

❑ **Q16** En s'inspirant du code de la fonction `trace_quadrant_est`, implémenter une nouvelle fonction `trace_quadrant_sud` qui règle le problème précédent.

❑ **Q17** Implémenter la fonction `trace_segment(im:Image, p0:(int), p1:(int))` qui trace un segment continu entre les pixels `p0` et `p1` sur l'image `im` supposée assez grande. Cette fonction devra opérer correctement si les deux points passés en arguments sont égaux.

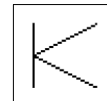
Partie V – Affichage de texte

La fonction `trace_segment` va maintenant permettre de tracer sur une page (comprendre une image matricielle) des glyphes à partir de leur description vectorielle.

❑ **Q18** Implémenter la fonction `position(p:(float), pz:(int), taille:int)->(int)` qui renvoie les coordonnées du point `p` (point d'un glyphe de coordonnées flottantes) en un point dans une page (pixel de coordonnées entières) de manière à ce que le point (0,0) de la description vectorielle soit en position `pz` sur la page, et que l'œil de taille normalisée 1 du glyphe fasse `taille` pixels de hauteur. Prendre garde à la bonne orientation du glyphe sur la page. Vérifier que, pour la taille limite 1, l'œil du glyphe fait bien 1 pixel de hauteur.

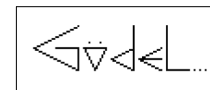
❑ **Q19** Implémenter la fonction `affiche_car(page:Image, c:str, police:str, roman:bool, pz:(int), taille:int)->int` qui affiche dans l'image `page` le caractère `c` dans la police `police`, en roman ou italique selon la valeur du booléen `roman`, et renvoie la largeur en pixel du glyphe affiché. Pour rappel, la fonction `glyphe(c:str, police:str, roman:bool)` charge la description vectorielle du caractère `c` dans la police `police` pour la variante `roman`.

```
1 from affiche import affiche_car
2 from PIL import Image
3 page = Image.new("1", (50, 50), color=1)
4 avance = affiche_car(page, "K", "Triangle", True, [ 10, 40 ], 16)
5 page.save("K.png")
```



❑ **Q20** Implémenter la fonction `affiche_mot(page:Image, mot:str, ic:int, police:str, roman:bool, pz:(int), taille:int)->int` qui affiche la chaîne de caractères `mot` dans les mêmes conditions, chaque glyphe étant séparé du suivant par `ic` pixels, et renvoie la position du dernier pixel de la dernière lettre dans la page.

```
1 from affiche import affiche_mot
2 from PIL import Image
3 page = Image.new("1", (110, 50), color=1)
4 avance = affiche_mot(page, "Gödel...", 2, "Triangle", True, [ 10, 35 ], 13)
5 page.save("goedel.png")
```



De la même manière, on pourrait implémenter une fonction `affiche_ligne(page:Image, ligne:[str], ic:int, im:int, police:str, roman:bool, pz:(int), taille:int)` qui afficherait la liste de mots `ligne` en les séparant de `im` pixels.

Partie VI – Justification d'un paragraphe

L'objectif de cette dernière partie est d'afficher un paragraphe de manière harmonieuse en le justifiant, c'est-à-dire en alignant les mots sur les bords gauche et droit de la zone d'écriture de la page.

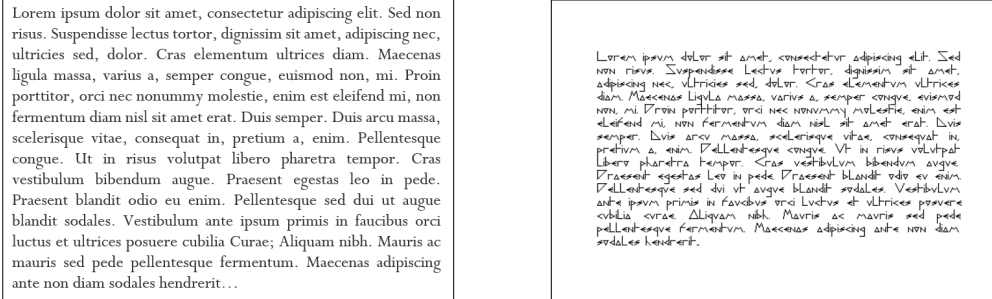


FIGURE 1 – Illustration de la justification de paragraphe pour différentes polices

Le paragraphe à justifier est constitué d'une liste de mots (chaînes de caractères). La difficulté est de placer des espaces entre les mots et de découper la liste en sous-listes pour que les lignes soient équilibrées (pas de ligne avec beaucoup d'espaces à la fin ou entre les mots par exemple). Pour simplifier le problème, on considère que le paragraphe est constitué d'une liste `lmots` d'entiers correspondant aux longueurs de chaque mot du paragraphe dans l'ordre d'apparition des mots (`lmots[i]` correspond au nombre de caractères du mot d'indice i). On note L le nombre de caractères et espaces que peut contenir une ligne au maximum. Il faut au minimum un espace entre deux mots d'une même ligne, on suppose que cet espace minimal correspond à un caractère.

On propose dans un premier temps l'algorithme glouton suivant :

```

1  def glouton(Lmots:[int],L:int)->[[int]]:
2      lignes=[]
3      nligne=[]
4      l=0
5      for c in Lmots :
6          if (c + 1) > L:
7              lignes.append(nligne)
8              nligne=[c]
9              l=c+1
10         else:
11             l=l+c+1
12             nligne.append(c)
13         lignes.append(nligne)
14     return lignes

```

□ **Q21** Expliquer en une ou deux phrases le principe de l'algorithme et pourquoi il est dit glouton.

Cet algorithme fournit une solution mais qui n'est pas nécessairement optimale. Si on teste cet algorithme sur le paragraphe suivant extrait du lorem ipsum : `ut enim ad minima veniam` pour une longueur de ligne maximale $L=10$, le résultat obtenu est le découpage noté a). Si on utilise une méthode de programmation dynamique, on obtient le découpage noté b).

a) Découpage obtenu par l'algorithme glouton	b) Découpage obtenu par programmation dynamique
<code>ut enim ad</code>	<code>ut enim</code>
<code>minima</code>	<code>ad minima</code>
<code>veniam</code>	<code>veniam</code>

Pour évaluer la pertinence du placement d'espaces et de retours à la ligne, on définit une fonction coût à minimiser pour que la répartition soit la plus harmonieuse possible.

Cette fonction coût correspond au nombre d'espaces disponibles sur une ligne élevé au carré, si on commence la ligne du mot i au mot j inclus sur cette même ligne :

$$\text{cout}(i, j) = (L - (j - i) - \sum_{k=i}^j \text{lmots}[k])^2$$

Cette fonction prend la valeur ∞ lorsque la somme des longueurs des mots de i à j est supérieure à L (ce qui signifie qu'on ne peut pas placer les mots i à j sur une même ligne).

La fonction python suivante correspond à l'implémentation de cette fonction coût.

```

1 def cout(i:int,j:int,lmots:[int],L:int)->int:
2     res=sum(lmots[i:j+1])+(j-i)
3     if res>L:
4         return float("inf")
5     else:
6         return (L-res)**2

```

□ **Q22** Évaluer pour les deux découpages a) et b) de l'exemple, ce que renvoie la fonction coût pour chacune des lignes en précisant les indices i et j pour chaque ligne ($\text{lmots}=[2,4,2,6,6]$). Conclure sur l'algorithme qui donne la solution la plus harmonieuse en sommant les différents coûts par ligne pour le découpage a) puis pour le découpage b).

La méthode de programmation dynamique consiste dans un premier temps à déterminer une équation de récurrence (équation de Bellman). Le problème peut être reformulé de la manière suivante :

Si on suppose connue la solution pour placer les mots jusqu'à un indice i , le nouveau problème consiste à placer correctement les changements de lignes du mot i jusqu'à la fin. La question est donc de savoir où placer le changement de ligne à partir du mot d'indice i , de manière à minimiser la fonction coût.

On note $d(i)$ le problème du placement optimal de changement de ligne jusqu'à l'indice i . L'équation de Bellman correspondante est alors :

$$d(i) = \min_{i < j \leq n} (d(j) + \text{cout}(i, j - 1))$$

Un algorithme récursif naïf correspondant à la résolution de ce problème est le suivant.

```

1 def algo_recuratif(i:int,lmots:[int],L:int)->int:
2     if i==len(lmots):
3         return 0
4     else:
5         mini=float("inf")
6         for j in range(i+1,len(lmots)+1):
7             d=algo_recuratif(j,lmots,L)+cout(i,j-1,lmots,L)
8             if d<mini:
9                 mini=d
10        return mini

```

□ **Q23** Proposer une modification de la fonction `algo_recuratif` pour rendre celle-ci plus efficace en introduisant une mémorisation.

On définira la nouvelle fonction récursive `progd_memo(i:int,lmots:[int],L:int,memo:{int:int})` avec la variable `memo`, dictionnaire initialisé en dehors de la fonction par `memo={len(m):0}`

On donne finalement une fonction utilisant la méthode de calcul de bas en haut. Cette fonction renvoie le coût optimal au problème de découpage de texte global de manière équivalente à la fonction `progd_memo`.

```

1 def progdbashaut(lmots:[int],L:int)->int:
2     M=[0]*(len(lmots)+1)
3     for i in range(len(lmots)-1,-1,-1):
4         mini,indi=float("inf"),-1
5         for j in range(i+1,len(lmots)+1):
6             d=M[j]+cout(i,j-1,lmots,L)
7             if d<mini:
8                 mini,indi=d,j
9         M[i]=mini
10    return M[0]

```

□ **Q24** Analyser, en fonction de n nombres de mots, la complexité temporelle asymptotique associée à l'algorithme récursif naïf (fonction `algo_recuratif`) ainsi qu'à l'algorithme de programmation dynamique de bas en haut (fonction `progd_bashaut`) et conclure sur l'intérêt de l'algorithme de programmation dynamique. On ne comptera que les opérations de type additions/soustractions. Il n'est pas attendu de développements mathématiques poussés, les résultats peuvent être donnés sans justification.

On modifie légèrement la fonction `progd_bashaut` en ajoutant un argument `t` en plus des variables précédentes pour extraire les valeurs d'indices de découpe de lignes.

```

1  def progdbashaut(lmots:[int],L:int,t:[int])>->int:
2      M=[0]*(len(lmots)+1)
3      for i in range(len(lmots)-1,-1,-1):
4          mini,indi=float("inf"),-1
5          for j in range(i+1,len(lmots)+1):
6              d=M[j]+cout(i,j-1,lmots,L)
7              if d<mini:
8                  mini,indi=d,j
9          t[i]=indi
10         M[i]=mini
11     return M[-1]
```

`t[i]` est la valeur de l'indice dans la liste `lmots` correspondant au placement optimisé sur une même ligne des mots d'indice `i` jusqu'à `t[i]` exclus. Cette liste `t` est modifiée en place dans la fonction. Pour l'exemple étudié précédemment (cas b), on a ainsi obtenu la liste : `t=[2, 3, 4, 4, 5]`.

On dispose de la liste des mots (chaînes de caractères cette fois-ci) notée `mots`.

La fonction `lignes(mots:[str],t:[int],L:int)>->[[str]]` doit renvoyer une liste de listes de mots (chaque sous-liste correspond à une ligne) en fonction de la liste `t` donnée par l'algorithme. La fonction `lignes(["Ut", "enim", "ad", "minima", "veniam"], [2,3,4,4,5],10)` renvoie :

```
[[["Ut", "enim"], ["ad", "minima"], ["veniam"]].
```

De même, en prenant,

```

t=[2, 3, 5, 5, 5, 6, 7, 9, 11, 11, 11]
L=15
mots=["Lorem", "ipsum", "dolor", "sit", "amet", "consectetur", "adipiscing",
      "elit.", "Sed", "non", "risus."]
```

la fonction `lignes(mots,t,L)` renvoie :

```

[[["Lorem", "ipsum"],
  ["dolor", "sit", "amet", "consectetur"],
  ["adipiscing", "elit.", "Sed"],
  ["non", "risus."]]]
```

□ **Q25** Proposer une implémentation de cette fonction `lignes(mots:[str],t:[int],L:int)`

Il reste à écrire une fonction `formatage(lignesdemots: [[str]], L:int)` qui renvoie une chaîne de caractères correspondant à la justification du paragraphe à partir des listes de mots par ligne `lignesdemots` et de la longueur maximale `L` d'une ligne en termes de caractères et espaces. Les retours à la ligne seront représentés par le symbole `"\n"`. Les espaces devront être répartis équitablement entre les mots pour que la justification se fasse bien entre la marge gauche et la marge droite (en respectant la longueur `L` maximale imposée). On obtient par exemple pour `L=10` :

```

ut      enim
ad minima
veniam
```

□ **Q26** Proposer une implémentation de cette fonction `formatage(lignesdemots: [[str]], L:int)>->str` qui renvoie la chaîne de caractères justifiée.

Fin de l'épreuve.

SOLUTION

Q1. 100 cents en base hexadécimale (c'est-à-dire en base 16) vaut

$$0 \times 16^0 + 0 \times 16^1 + 1 \times 16^2 = 256$$

centimes soit, sans tenir compte de l'inflation, 2 dollars et 56 cents.

Commentaires : Question sans difficulté ... si on comprend la question :-).

Q2. Cela représente la lettre *j* (on relit dans cet ordre les points $(0.25, 1)$, $(0.25, -1)$, $(0.0, -1)$) puis on place le point $(0.25, 1.25)$ (le point sur le *j*).

Commentaires : Le but de la question est de familiariser le candidat avec les notations du sujet. Il est important d'y passer un peu de temps si nécessaire afin de bien comprendre les notions en jeu. Ne pas le faire expose le candidat à être rapidement dans l'incapacité à répondre aux questions informatiques ultérieures.

Q3.

```
SELECT count(*) FROM glyphe WHERE groman = 1
```

Commentaires : Question élémentaire. Le seul piège est d'avoir ingurgité toutes les notations du sujet (notamment comment se transcrit roman).

Q4.

```
SELECT gdesc FROM glyphe
WHERE pid =(SELECT pid FROM police WHERE pnom="Helvetica") AND
groman = 0
```

Commentaires : Question étonnamment difficile comme deuxième question de SQL au concours Mines-Ponts car il demande d'utiliser une sous-requête. Le sujet aurait pu mettre en évidence le pid d'Helvetica. S'il l'avait fait, la question aurait été de difficulté standard.

Q5.

```
SELECT famille.fnom, count(*)
FROM famille JOIN police ON famille.fid = police.fid
GROUP BY police.fid
ORDER BY famille.fnom
```

Commentaires : Question sélective destinée à distinguer les candidats ayant une bonne maîtrise du SQL du programme. Le comptage par famille pose toujours problème à un nombre significatif de candidats. N'hésitez pas à le travailler, la question est récurrente à ce concours.

Q6. Une version pythonesque

```
def points(v):
    l=[]
    for e in v:
        for j in e:
            l.append(j)
    return l
```

Une version avec les indices

```
def points(v):
    l=[]
    for i in range(len(v)):
        for j in range(len(v[i])):
            l.append(v[i][j])
    return l
```

Commentaires : *Question de difficulté standard qui demande au candidat une maîtrise de la gestion des listes. Elle s'avère en pratique discriminante.*

Q7. Trois versions (première avec itérateur et append, la deuxième avec itérateur et liste en compréhension, la troisième avec indice et append)

```
def dim(l,n):
    temp=[]
    for e in l:
        temp.append(e[n])
    return temp

def dim(l,n):
    return [e[n] for e in l]

def dim(l,n):
    temp=[]
    for i in range(len(l)):
        temp.append(l[i][n])
    return temp
```

Commentaires : *Même commentaire que précédemment.*

Q8. On extrait les abscisses de tous les points de la glyphe. On choisit les deux extrêmes (plus à gauche et plus à droite) ce qui donne la largeur de la glyphe.

```
def largeur(l):
    mini,maxi=0,0
    for v in l:
        m1 = min(dim(v,0))
        m2 = max(dim(v,0))
        if m1 < mini:
```

```

        mini = m1
    if m2 > maxi:
        maxi = m2
    return maxi-mini

```

Commentaires : Question sélective qui demande un certain recul du candidat et une bonne compréhension de la gestion des listes.

Q9.

```

def obtention_largeur(police):
    temp = []
    for c in police:
        for b in [True, False]:
            temp.append(largeur(glyphe(c, police, b)))
    return temp

```

Commentaires : Question de difficulté standard

Q10. Deux versions : liste en compréhension ou avec append

```

def transforme(f , v):
    return [ [f(c) for c in e ] for e in v]

def transforme(f , v):
    temp=[]
    for e in v :
        temp1=[]
        for c in e:
            temp1.append(f(c))
        temp.append(temp1)
    return temp

```

Commentaires : Question discriminante qui demande encore une maîtrise certaine des listes par le candidat. Il est évidemment possible de traiter cette question par des indices (double indice pour accéder aux éléments atomiques) même si cela alourdit le code. Le candidat n'est absolument pas sanctionné pour cela.

Q11. L'image est rétrécie de moitié sur sa largeur.

Commentaires : Ce type de question est très discriminante car il est attendu du candidat d'interpréter ce résultat et de ne pas se limiter à décrire les opérations ("multiplier par 0.5 ..." etc). Le candidat se doit d'avoir du recul et de contextualiser le code. Ce type de question est récurrente à ce concours.

Q12.

```

def penche(v):
    def z(p):

```

```
return [p[0]+0.5*p[1],p[1]]
return transforme(z,v)
```

Commentaires : Question de difficulté standard afin de s'assurer que le candidat maîtrise les notions introduites précédemment concernant les fonctions d'ordre supérieures sur les listes.

Q13. On a

$$x_0 = 0, y_0 = 0, x_1 = 6, y_1 = 2, dx = 6, dy = 2, \frac{dy}{dx} = \frac{1}{3}.$$

Ainsi, les points tracés consécutivement ont pour coordonnées (x_i, y_i) avec

i	1	2	3	4	5
x_i	1	2	3	4	5
y_i	0	1	1	1	2

Commentaires : Le but de la question est de familiariser le candidat avec les notations du sujet. Il est important d'y passer un peu de temps si nécessaire afin de bien comprendre les notions en jeu. Ne pas le faire expose le candidat à être rapidement dans l'incapacité à répondre aux questions informatiques ultérieures.

Q14. On a

$$x_0 = 9, y_0 = 8, x_1 = 1, y_1 = 9, dx = -8, dy = 1, \frac{dy}{dx} = -\frac{1}{8}.$$

Ainsi, les points tracés consécutivement ont pour coordonnées (x_i, y_i) avec

i	1	2	3	4	5	6	7
x_i	10	11	12	13	14	15	16
y_i	8	8	8	8	7	7	7

Les points construits ne sont manifestement pas entre les points $p_0 = (9, 8)$ et $p_1 = (1, 9)$! Il faut commencer par vérifier que le point p_1 soit bien à l'est de p_0 c'est-à-dire que p_0 soit plus à " gauche " (graphiquement parlant) que le point p_1 i.e. $x_0 \leq x_1$. Si ce n'est pas le cas, on construit le segment entre p_1 et p_0 . Dans ce cas, on aurait

$$x_0 = 1, y_0 = 9, x_1 = 9, y_1 = 8, dx = 8, dy = -1, \frac{dy}{dx} = -\frac{1}{8}.$$

Ainsi, les points tracés consécutivement ont pour coordonnées (x_i, y_i) avec

i	1	2	3	4	5	6	7
x_i	2	3	4	5	6	7	8
y_i	9	9	9	9	8	8	8

Commentaires : Même commentaire que précédemment. J'y ajoute que le recul du candidat est attendu ainsi qu'une réflexion scientifique pour apporter une solutions satisfaisante au problème (même s'il n'est pas capable de le quantifier, repérer la nature du problème et ce qu'il est nécessaire de mettre en place). Question sélective.

Q15. On a

$$x_0 = 3, y_0 = 0, x_1 = 5, y_1 = 8, dx = 2, dy = 8, \frac{dy}{dx} = 4.$$

Ainsi, les points tracés consécutivement ont pour coordonnées (x_i, y_i) avec

$$\begin{array}{rcl} i & & 1 \\ x_i & & 4 \\ y_i & & 4 \end{array}$$

Il n'y a qu'un point tracé, ce qui est gênant pour représenter un segment entre deux points suffisamment éloignés. Il est préférable d'augmenter d'un pas chaque ordonnée et de faire un pas de $\frac{dx}{dy}$ pour les abscisses (car la pente est plus faible que sur les ordonnées). Cela nous donnera les points suivants

$$(x_i, y_i) = \left(\left\lfloor x_i + i \times \frac{dx}{dy} \right\rfloor, y_i + i \right)$$

avec $\frac{dx}{dy} = \frac{1}{4}$ et $i \in \{1, \dots, dy - 1\} = \{1, \dots, 7\}$.

$$\begin{array}{rcl} i & 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ x_i & 3 & 4 & 4 & 4 & 4 & 5 & 5 \\ y_i & 1 & 2 & 3 & 4 & 5 & 6 & 7 \end{array}$$

Commentaires : *Mêmes commentaires que précédemment.*

Q16. On choisit la pente $\frac{dy}{dx}$ ou $\frac{dx}{dy}$ la plus faible pour faire varier lentement les coordonnées les moins éloignées et on incrémente d'un pas les coordonnées les plus éloignées. Voici le code correspondant.

```
def trace_quadrant_sud(im, p0, p1):
    x0, y0 = p0
    x1, y1 = p1
    dx, dy = x1-x0, y1-y0
    im.putpixel(p0, 0)
    if abs(dy/dx) <= abs(dx/dy):
        for i in range(1, dx):
            p = (x0 + i, y0 + floor(0.5 + dy * i / dx))
            im.putpixel(p, 0)
    else:
        for i in range(1, dy):
            p = (x0 + floor(0.5 + dx * i / dy), y0 + i)
            im.putpixel(p, 0)
    im.putpixel(p1, 0)
```

Commentaires : *Question sans difficulté informatique particulière hormis de bien structurer son code et de n'oublier aucun cas. Par contre, elle s'avère très sélective car les candidats n'ayant pu répondre à la question précédente seront incapables de la traiter (ils sont nombreux!)*

Q17. On fait attention à ce que le point p_1 soit au sud de p_0 c'est-à-dire que $y_1 \geq y_0$ (c'est-à-dire $dy \geq 0$) sinon on rejoint le point p_1 à p_0 . Le cas particulier où $y_1 = y_0$, il suffit de relier les points p_0 et p_1 par la fonction `trace_quadrant_est`. On choisit la pente $\frac{dy}{dx}$ ou $\frac{dx}{dy}$ la plus faible pour faire varier lentement les coordonnées les moins éloignées et on incrémente d'un pas les coordonnées les plus éloignées. Voici le code correspondant.

```
def trace_quadrant_sud(im, p0, p1):
    x0, y0 = p0
    x1, y1 = p1
    dx, dy = x1-x0, y1-y0
    if dx < 0:
        return trace_quadrant_sud(im, p1, p0)
    if dy == 0 and dx == 0:
        im.putpixel(p0, 0)
    if dx == 0 and dy != 0:
        if dy < 0:
            return trace_quadrant_sud(im, p1, p0)
        im.putpixel(p0, 0)
        for i in range(1, dy):
            p = (x0, y0 + i)
            im.putpixel(p, 0)
    im.putpixel(p1, 0)
    # il reste le cas où les deux pentes dy/dx et dx/dy sont
    # non nulles
    # on prend la pente la plus faible
    im.putpixel(p0, 0)
    if abs(dy/dx) <= abs(dx/dy):
        for i in range(1, dx):
            p = (x0 + i, y0 + floor(0.5 + dy * i / dx))
            im.putpixel(p, 0)
    else :
        for i in range(1, dy):
            p = (x0 + floor(0.5 + dx * i / dy), y0 + i)
            im.putpixel(p, 0)
    im.putpixel(p1, 0)
```

Commentaires : *Mêmes commentaires que précédemment.*

Q18.

```
def position(p, pz, taille):
    x0, y0 = p
    x1, y1 = pz
    return (float(taille*x0+x1), float(taille*y0+y1))
```

Commentaires : *Si la question ne soulève aucun problème d'ordre informatique, comme dans de trop nombreuses questions de ce sujet, la difficulté se trouve dans ce que le candidat doit coder. Cela a posé un problème à un nombre important de candidats.*

Q19.

```
def affiche_car(page,c,police,roman,pz,taille):
    gly=glyphe(c, police, roman)
    pzb=pz
    for e in gly:
        if len(e) ==1 :
            p0 = position(e[0],pzb,taille)
            trace_segment(page, p0, p0)
            pzb=p0
        else:
            m=len(n)
            for i in range(m-1):
                p0=position(e[i],pzb,taille)
                p1= position(e[i+1],pzb,taille)
                trace_segment(page, p0, p1)
                pzb=p1
    return largeur(gly)
```

Commentaires : *Question très sélective car elle demande une bonne compréhension par le candidat du contexte et de la fluctuation de la position pz. Sans compter la gestion des points uniques.*

Q20.

```
def affiche_mot(page,mot,ic,police,roman,pz,taille):
    pzb=pz
    for c in mot:
        lar = affiche_car(page,c,police,roman,pzb,taille)
        pzb=(pzb[0]+ic+lar,pzb[1])
    return pzb
```

Commentaires : *Question de difficulté standard avec toujours le problème de compréhension du contexte par le candidat.*

Q21. A chaque étape, si on peut insérer le mot dans la ligne, on l'insère et on laisse un espace libre correspondant à celui d'un caractère (ligne 11,12). Sinon, on change de ligne (ligne 6,7) puis on place le mot sur la ligne suivante (ligne 8). Il est glouton car on essaie d'insérer à chaque fois le plus de mots consécutifs dans une ligne donnée.

Commentaires : *Question élémentaire mais discriminante car un nombre significatif de candidats ont du mal à interpréter et contextualiser du code informatique.*

Q22. Rappelons que le sujet a fixé $L = 10$ et la liste $lmot$ correspondant à l'exemple est $lmots=[2,4,2,6,6]$ avec

$$\text{cout}(i, j) = \left(10 - (j - i) - \sum_{k=i}^j \text{lmot}[k] \right)^2.$$

Cas de l'exemple a). Pour la première ligne, on a $i = 0$ et $j = 2$ (on traite les trois premiers mots d'indice 0, 1 et 2) donc

$$\text{cout}(0, 2) = \left(10 - (2 - 0) - \sum_{k=0}^2 \text{lmot}[k] \right)^2 = (10 - 6 - 2 - 2)^2 = 0.$$

Pour la deuxième ligne, on a $i = j = 3$ (on traite le quatrième mot qui est d'indice 3) donc

$$\text{cout}(3, 3) = (10 - (3 - 3) - \text{lmot}[3])^2 = (10 - 6)^2 = 16.$$

Pour la troisième ligne, on a $i = j = 4$ (on traite le cinquième mot qui est d'indice 4) donc

$$\text{cout}(4, 4) = (10 - (4 - 4) - \text{lmot}[4])^2 = (10 - 6)^2 = 16.$$

Le coût total de ces trois lignes est donc $0 + 16 + 16 = 32$.

Cas de l'exemple b). Pour la première ligne, on a $i = 0$ et $j = 1$ (on traite les deux premiers mots d'indice 0 et 1) donc

$$\text{cout}(0, 1) = \left(10 - (1 - 0) - \sum_{k=0}^1 \text{lmot}[k] \right)^2 = (10 - 1 - 2 - 4)^2 = 9.$$

Pour la deuxième ligne, on a $i = 2$ et $j = 3$ (on traite les troisième et quatrième mots qui sont d'indice 2 et 3) donc

$$\text{cout}(2, 3) = \left(10 - (3 - 2) - \sum_{k=2}^3 \text{lmot}[k] \right)^2 = (10 - 1 - 2 - 6)^2 = 1.$$

Pour la troisième ligne, on a $i = j = 4$ (on traite le cinquième mot qui est d'indice 4) donc

$$\text{cout}(4, 4) = (10 - (4 - 4) - \text{lmot}[4])^2 = (10 - 6)^2 = 16.$$

Le coût total de ces trois lignes est donc $9 + 1 + 16 = 26$ donc l'exemple b) a un coût total inférieur à celui de l'exemple a).

Commentaires : Le but de la question est de familiariser le candidat avec les notations du sujet. Il est important d'y passer un peu de temps si nécessaire afin de bien comprendre les notions en jeu. Ne pas le faire expose le candidat à être rapidement dans l'incapacité à répondre aux questions informatiques ultérieures.

Q23.

```
def prog_d_memo(i, lmots, L, memo):
    if i in memo.keys():
        return memo[i]
```

```

else:
    mini=float("inf")
    for j in range(i+1,len(lmots)+1):
        if j in memo.keys():
            dj=memo[j]
        else:
            dj=progd_memo(j,lmots,L,memo)
        d=dj+cout(i,j-1,lmots,L)
        if d<mini:
            mini=d
    memo[i]=mini
    return mini

```

La version sans dictionnaire externe est la suivante

```

def algo_recuratif_memo(i,lmots,L):
    dico={len(lmots):0}
    return prog_d_memo(0,lmots,L,dico)

```

Commentaires : *S'il s'agit d'une question de cours mais elle est sélective car elle demande au candidat une compréhension de la mémorisation et de sa mise en place. N'hésitez pas à travailler cette notion car elle devient un standard des concours. Un nombre significatif de candidat ont encore du mal à manipuler les dictionnaires. Si c'est votre cas, n'hésitez pas à travailler cela puisque cela deviendra probablement une question classique à ce concours.*

Q24. La complexité est au moins exponentielle en n (comme pour la suite de Fibonacci en récursif naïf).

Voici une ébauche de preuve formelle (avec de légères modifications). Si on note c_n la complexité minimale de ce programme, comme il appelle ce programme pour les entrées de taille $0, 1, 2, \dots, n-1$, sa complexité est au moins égal à $n-1 + c_0 + \dots + c_{n-1}$ donc

$$c_n \geq n-1 + c_0 + \dots + c_{n-1} \geq 1 + c_0 + \dots + c_{n-1}$$

(pour $n \geq 2$). Si $c_k \geq 2^k$ pour tout entier $k < n$ alors

$$c_n \geq 1 + 1 + 2 + \dots + 2^{n-1} = 1 + \frac{2^n - 1}{2 - 1} = 1 + 2^n - 1 = 2^n$$

donc $c_n \geq 2^n$ pour tout entier n .

La version avec dictionnaire a une complexité en $O(n^2)$ (deux boucles *for* imbriquées, chacune s'itérant au plus n fois, chaque instruction étant de coût constant dans les boucles et à l'extérieur).

Commentaires : *Question très difficile, voire inaccessible (et largement hors programme) pour la version sans dictionnaire. Seuls les meilleurs copies auront donné une réponse raisonnable (sans preuve bien entendu). La version avec dictionnaire doit pouvoir être traitée convenablement pour un candidat souhaitant être admissible au concours Mines-Ponts. Cette version est sélective.*

Q25.

```
def lignes(mots,t,L):
    resultat=[]
    n=len(t)
    indice = 0
    while indice < n:
        temp=[]
        for i in range(indice,t[indice]):
            temp.append(mots[i])
        resultat.append(temp)
        indice=t[indice]
    return resultat
```

Commentaires : *C'est dommage que cette question soit à la fin du sujet car elle est de niveau standard pour ce concours. Au final, elle aura été peu traitée car les candidats auront été perdu probablement par les questions précédentes.*

Q26. On crée une fonction *espace(s)* construisant une chaîne de caractères contenant *s* espaces consécutifs (pour séparer les mots). Ensuite, on construit la fonction *format_ligne(mots,L)* qui, à une liste de mots (d'une ligne donnée) crée des espaces équilibrés entre chacun de ces mots (on détermine l'espace vide restant, on le distribue uniformément entre chaque mot et on complète la ligne si nécessaire par des espaces vides pour qu'elle soit bien de longueur *L*). Vient ensuite la construction de la fonction principale *formatage(lignesdemots,L)*. Elle commence par extraire tous les mots de la liste de ligne *lignesdemots* (via la première fonction *points* du sujet), on applique alors la fonction *progd_bashaut_bis(lmots,L,t)* (avec un tableau *t* initialisé à 0) qui détermine l'équilibrage optimal des lignes puis on formate chacune des lignes. On les concatène ensuite en une seule chaîne de caractères en séparant chaque ligne par un changement de ligne `\n`.

```
def espace(s):
    temp = ""
    for i in range(s):
        temp = temp + " "
    return temp

def format_ligne(mots,L):
    reste = L
    for mot in mots:
        reste = reste - len(mot)
    ecart = espace(reste // len(mots))
    fin = espace(reste % len(mots))
    resultat = ""
    for mot in mots:
        resultat = resultat + mot + ecart
    resultat = resultat + fin
    return resultat

def formatage(lignesdemots,L):
    mots=points(lignesdemots)
```

```
t=[0 for k in range(len(mots))]  
lmots=[]  
for mot in mots:  
    lmots.append(len(mot))  
progd_bashaut_bis(lmots,L,t)  
ligne = lignes(mots,t,L)  
texte = ""  
for l in ligne:  
    l1 = format_ligne(l,L)  
    texte = texte + l1 + "\n"  
return texte
```

Commentaires : *Toutes mes félicitations aux candidats ayant traité (à peu près) correctement cette question. Ils ont certainement obtenu une excellente note à cette épreuve.*



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2023

PREMIÈRE ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 3 heures

L'usage de la calculatrice ou de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES I - MP

L'énoncé de cette épreuve comporte 5 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Théorème de stabilité de Liapounov

Dans tout le problème, n désigne un entier naturel non nul. On note $\langle \cdot | \cdot \rangle$ le produit scalaire usuel de $\mathbf{K}^n$, $\mathbf{K}$ pouvant être $\mathbf{R}$ ou $\mathbf{C}$, et $\|\cdot\|$ la norme euclidienne associée.

Si u et v sont deux applications linéaires pour lesquelles la notation $u \circ v$ a un sens, alors on note uv l'application $u \circ v$. De plus, si u est un endomorphisme d'un espace vectoriel E et k est un entier naturel non nul, u^k désigne l'application $u \circ \dots \circ u$, où u apparaît k fois dans l'écriture. Par convention $u^0 = id_E$.

On s'intéresse au système différentiel suivant :

$$\begin{cases} y' &= \varphi(y) \\ y(0) &= x_0 \end{cases},$$

avec $x_0 \in \mathbf{R}^n$ et φ est une application de classe $\mathcal{C}^1$ de $\mathbf{R}^n$ à valeurs dans $\mathbf{R}^n$, telle que $\varphi(0) = 0$. Cela entraîne que si $x_0 = 0$, alors la solution de ce système est la fonction nulle, et donc 0 est un point d'équilibre. Notons $d\varphi(0)$ l'application différentielle de φ en 0. L'objectif de ce problème est d'établir une condition suffisante sur le spectre de $d\varphi(0)$ pour assurer la stabilité de l'équilibre en ce point, et d'obtenir des informations quant à la dynamique des solutions au voisinage de ce point d'équilibre. Plus précisément, on établit le résultat suivant :

Théorème de Liapounov :

Soit le système différentiel suivant :

$$\begin{cases} y' &= \varphi(y) \\ y(0) &= x_0 \end{cases},$$

avec $x_0 \in \mathbf{R}^n$ et φ est une application de classe $\mathcal{C}^1$ de $\mathbf{R}^n$ à valeurs dans $\mathbf{R}^n$, telle que $\varphi(0) = 0$ et telle que toutes les valeurs propres complexes de $d\varphi(0)$ aient une partie réelle strictement négative. Alors il existe trois constantes $\tilde{\alpha}$, C et β strictement positives telles que :

$$\forall x_0 \in B(0, \tilde{\alpha}), \quad \forall t \in \mathbf{R}_+, \quad \|f_{x_0}(t)\| \leq C e^{-\beta t} \|x_0\|,$$

où f_{x_0} est l'unique solution du système différentiel et $B(0, \tilde{\alpha})$ désigne la boule ouverte, pour la norme $\|\cdot\|$, de centre 0 et de rayon $\tilde{\alpha}$.

Dans une première partie, on étudie une norme sur les endomorphismes des sous-espaces vectoriels de $\mathbf{K}^n$. Dans la seconde partie, on établit des résultats sur le système différentiel linéaire, en se servant des résultats de la partie A. Enfin, la troisième partie est consacrée à la démonstration du théorème de Liapounov. Cette dernière partie est très largement indépendante des deux premières, à l'exception du résultat obtenu à la fin de la partie B.

A.. Etude d'une norme sur $\mathcal{L}(E)$

Soit E un sous-espace vectoriel de $\mathbf{K}^n$. Soit u un endomorphisme de E .

1 ▷ Après avoir justifié l'existence des bornes supérieures, montrer que :

$$\sup_{\substack{x \in E \\ x \neq 0}} \frac{\|u(x)\|}{\|x\|} = \sup_{\substack{x \in E \\ \|x\|=1}} \|u(x)\|.$$

2 ▷ On note $\|u\| = \sup_{\substack{x \in E \\ x \neq 0}} \frac{\|u(x)\|}{\|x\|}$. Montrer que $\|\cdot\|$ est une norme sur $\mathcal{L}(E)$.

3 ▷ Montrer qu'il s'agit d'une norme sous-multiplicative, c'est-à-dire que :

$$\forall (u, v) \in \mathcal{L}(E)^2, \quad \|uv\| \leq \|u\| \cdot \|v\|,$$

et en déduire une majoration de $\|u^k\|$, pour tout entier naturel k , en fonction de $\|u\|$ et de l'entier k .

B.. Etude de la stabilité en 0 du système linéaire

Dans cette partie, a désigne un endomorphisme de $\mathbf{C}^n$.

4 ▷ Montrer qu'il existe un entier naturel non nul r , des nombres complexes distincts $\lambda_1, \lambda_2, \dots, \lambda_r$, ainsi que des entiers naturels non nuls $m_1, m_2, \dots, m_r$, tels que :

$$\mathbf{C}^n = \bigoplus_{i=1}^r E_i,$$

où pour $i \in \llbracket 1; r \rrbracket$, $E_i = \text{Ker}(a - \lambda_i \text{id}_{\mathbf{C}^n})^{m_i}$.

D'après la question précédente, si x est un élément de $\mathbf{C}^n$, il existe un unique r -uplet $(x_1, \dots, x_r) \in E_1 \times \dots \times E_r$ tel que $x = \sum_{i=1}^r x_i$. Fixons à présent $i \in \llbracket 1; r \rrbracket$. On définit alors les endomorphismes :

$$p_i : \begin{cases} \mathbf{C}^n & \rightarrow & E_i \\ x & \mapsto & x_i \end{cases} \quad \text{et} \quad q_i : \begin{cases} E_i & \rightarrow & \mathbf{C}^n \\ x_i & \mapsto & x_i \end{cases}.$$

Par ailleurs, on note $\|\cdot\|_i$ la norme sur $\mathcal{L}(E_i)$ introduite à la partie A, à savoir

$$\forall u \in \mathcal{L}(E_i), \quad \|u\|_i = \sup_{\substack{x \in E_i \\ x \neq 0}} \frac{\|u(x)\|}{\|x\|}.$$

On utilisera la notation $\|\cdot\|_c$ pour $\mathcal{L}(\mathbf{C}^n)$. Enfin, on notera a_i l'endomorphisme $p_i a q_i$.

5 ▷ Montrer que, pour tout $i \in \llbracket 1; r \rrbracket$, il existe une constante $C_i > 0$ telle que :

$$\forall u \in \mathcal{L}(E_i), \quad \|q_i u p_i\|_c \leq C_i \|u\|_i.$$

6 ▷ Montrer que, pour $i \in \llbracket 1; r \rrbracket$, E_i est stable par a .

7 ▷ Soient $(i, j) \in \llbracket 1; r \rrbracket^2$. Exprimer $p_i q_j$ puis $\sum_{i=1}^r q_i p_i$ en fonction des endomorphismes $id_{\mathbf{C}^n}$ et id_{E_j} .

8 ▷ Montrer que : $a = \sum_{i=1}^r q_i a_i p_i$.

9 ▷ En déduire que :

$$\forall t \in \mathbf{R}, \quad e^{ta} = \sum_{i=1}^r q_i e^{ta_i} p_i.$$

10 ▷ Montrer par ailleurs que :

$$\forall i \in \llbracket 1; r \rrbracket, \quad \forall t \in \mathbf{R}, \quad \|e^{ta_i}\|_i \leq |e^{t\lambda_i}| \sum_{k=0}^{m_i-1} \frac{|t|^k}{k!} \|a_i - \lambda_i id_{E_i}\|_i^k.$$

11 ▷ En déduire l'existence d'un polynôme P à coefficients réels tel que :

$$\forall t \in \mathbf{R}, \quad \|e^{ta}\|_c \leq P(|t|) \sum_{i=1}^r e^{t \operatorname{Re}(\lambda_i)},$$

où $\operatorname{Re}(z)$ désigne la partie réelle d'un nombre complexe z .

12 ▷ Pour toute matrice $A \in \mathcal{M}_n(\mathbf{R})$, on notera u_A l'endomorphisme canoniquement associé à A dans $\mathbf{R}^n$ et v_A l'endomorphisme de $\mathbf{C}^n$ canoniquement associé à A , vue comme une matrice de $\mathcal{M}_n(\mathbf{C})$. On conservera la notation $\|\cdot\|_c$ pour la norme introduite à la partie A sur $\mathcal{L}(\mathbf{C}^n)$ et on utilisera $\|\cdot\|_r$ sur $\mathcal{L}(\mathbf{R}^n)$. Montrer qu'il existe $C > 0$ telle que :

$$\forall A \in \mathcal{M}_n(\mathbf{R}), \quad \forall t \in \mathbf{R}, \quad \|e^{tu_A}\|_r \leq C \|e^{tv_A}\|_c.$$

Dans la suite de cette partie, on considère u un endomorphisme de $\mathbf{R}^n$, et $A \in \mathcal{M}_n(\mathbf{R})$ sa matrice dans la base canonique. On notera par ailleurs, $Sp(A)$ le spectre complexe de A . Notons g_{x_0} l'unique solution de classe $\mathcal{C}^1$ sur $\mathbf{R}_+$ de :

$$\begin{cases} y' &= u(y) \\ y(0) &= x_0 \end{cases}.$$

13 ▷ Montrer que :

$$\forall x_0 \in \mathbf{R}^n, \quad \lim_{t \rightarrow +\infty} \|g_{x_0}(t)\| = 0 \iff Sp(A) \subset \mathbf{R}_-^* + i\mathbf{R}.$$

14 ▷ On se place dans cette question dans le cas où toutes les valeurs propres de A ont une partie réelle strictement négative. Montrer alors qu'il existe deux constantes C_2 et α strictement positives telles que :

$$\forall t \in \mathbf{R}_+, \quad \|e^{tu}\|_r \leq C_2 e^{-\alpha t},$$

et en déduire une majoration de $\|g_{x_0}(t)\|$ pour $t \in \mathbf{R}_+$.

C.. Démonstration du théorème de Liapounov

On considère dans cette partie une application φ de $\mathbf{R}^n$ dans $\mathbf{R}^n$ de classe $\mathcal{C}^1$ telle que $\varphi(0) = 0$, et en notant $a = d\varphi(0)$, telle que toutes les valeurs propres de a aient une partie réelle strictement négative.

Soit $x_0 \in \mathbf{R}^n$. On s'intéresse au système différentiel suivant :

$$\begin{cases} y' &= \varphi(y) \\ y(0) &= x_0 \end{cases}.$$

On admettra l'existence d'une solution de ce système définie sur $\mathbf{R}_+$, que l'on notera f_{x_0} .

15 ▷ Montrer que la fonction

$$b : \begin{cases} \mathbf{R}^n \times \mathbf{R}^n & \rightarrow \mathbf{R} \\ (x, y) & \mapsto \int_0^{+\infty} \langle e^{ta}(x) | e^{ta}(y) \rangle dt \end{cases}$$

est bien définie et qu'elle définit un produit scalaire sur $\mathbf{R}^n$.

On notera q la forme quadratique associée à b , c'est-à-dire que pour tout $x \in \mathbf{R}^n$, $q(x) = b(x, x)$.

16 ▷ Démontrer alors que :

$$\forall x \in \mathbf{R}^n, \quad dq(x)(a(x)) = 2b(x, a(x)) = -\|x\|^2.$$

Pour toute fonction y définie sur $\mathbf{R}_+$, on associe la fonction $\varepsilon(y)$ définie par :

$$\varepsilon(y) : \begin{cases} \mathbf{R}_+ & \rightarrow \mathbf{R}^n \\ t & \mapsto \varphi(y(t)) - a(y(t)) \end{cases}.$$

17 ▷ Vérifier l'égalité

$$\forall t \in \mathbf{R}_+, \quad q(f_{x_0})'(t) = -\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0}(t))).$$

18 ▷ Prouver l'existence de deux nombres réels α et β strictement positifs tels que, pour tout $t \in \mathbf{R}_+$, on ait :

$$q(f_{x_0}(t)) \leq \alpha \Rightarrow -\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0}(t))) \leq -\beta q(f_{x_0}(t)).$$

On fixe un tel couple (α, β) pour la suite de ce problème.

19 ▷ Montrer alors que :

$$q(x_0) < \alpha \Rightarrow \forall t \geq 0, \quad q(f_{x_0})(t) \leq e^{-\beta t} q(x_0).$$

20 ▷ En déduire l'existence de trois constantes $\tilde{\alpha}$, C et β strictement positives telles que :

$$\forall x_0 \in B(0, \tilde{\alpha}), \quad \forall t \in \mathbf{R}_+, \quad \|f_{x_0}(t)\| \leq C e^{-\frac{\beta}{2}t} \|x_0\|,$$

où $B(0, \tilde{\alpha})$ désigne la boule ouverte, pour la norme $\|\cdot\|$, de centre 0 et de rayon $\tilde{\alpha}$.

FIN DU PROBLÈME

SOLUTION

A. Etude d'une norme sur $\mathcal{L}(E)$

1. L'application u étant linéaire sur un espace de dimension finie, elle est continue. La sphère unité de E est bornée par définition et c'est aussi un fermé (image réciproque de $\{1\}$ par l'application norme qui est continue). C'est donc un compact (puisque fermé borné en dimension finie).
Par théorème de compacité, $x \mapsto \|u(x)\|$ est bornée et atteint ses bornes sur la sphère unité et on a donc existence de $\alpha = \max\{\|u(x)\|, x \in E \text{ et } \|x\| = 1\}$.

Si $x \in E$ est non nul, $\frac{x}{\|x\|}$ est dans E et de norme 1. Ainsi

$$\frac{\|u(x)\|}{\|x\|} = \left\| u\left(\frac{x}{\|x\|}\right) \right\| \leq \alpha$$

$x \mapsto \frac{\|u(x)\|}{\|x\|}$ est donc majoré sur $E \setminus \{0\}$ par α . De plus, le majorant α est atteint puisque la borne supérieure sur la sphère unité de E est en fait un maximum. α est donc le maximum de la fonction sur $E \setminus \{0\}$. A fortiori, on a existence de la borne supérieure et

$$\sup_{x \in E, x \neq 0} \frac{\|u(x)\|}{\|x\|} = \sup_{x \in E, \|x\|=1} \|u(x)\|$$

Rappel : un maximum est un majorant qui est atteint. Pour montrer que $\sup(A) = \alpha$, il y a deux étapes. Dans la première, on montre que α majore A . Dans la seconde, ou bien on montre que $\alpha \in A$ (auquel cas on a un maximum), ou bien on montre qu'il existe des éléments de A arbitrairement proches de α (caractérisation de la borne supérieure, par exemple séquentiellement). La première situation est bien sûr la plus simple.

2. L'application $|||\cdot|||$ est bien définie d'après la question 1. On a quatre propriétés à justifier.
- L'application $|||\cdot|||$ est positive (la borne supérieure de quantités positives est positive).
 - Si $|||u||| = 0$ alors $\forall x \in E \setminus \{0\}, 0 \leq \frac{\|u(x)\|}{\|x\|} \leq 0$. Ainsi u est nulle sur $E \setminus \{0\}$ et elle l'est aussi en 0 donc $u = 0$. On a donc l'axiome de séparation.
 - On a

$$\forall x \in E, \|\lambda u(x)\| = |\lambda| \|u(x)\|$$

et ainsi

$$|||\lambda u||| = \sup\{\|\lambda u(x)\|, x \in E \text{ et } \|x\| = 1\} = |\lambda| \sup\{\|u(x)\|, x \in E \text{ et } \|x\| = 1\}$$

l'égalité étant vraie car $|\lambda| \geq 0$. $|||\cdot|||$ est ainsi positivement homogène.

- Si $u, v \in \mathcal{L}(E)$ alors pour tout $x \in E$ tel que $\|x\| = 1$,

$$\|(u+v)(x)\| \leq \|u(x)\| + \|v(x)\| \leq |||u||| + |||v|||$$

et on obtient l'inégalité triangulaire en passant à la borne supérieure pour x variant dans la sphère unité de E .

$$||| \cdot ||| \text{ est une norme sur } \mathcal{L}(E)$$

Remarque : le programme permet de dire que $\sup(kA) = k \sup A$ quand k est positif, licence que nous avons utilisée. En revanche, il ne dit rien sur $\sup(A + B)$ et il est préférable de détailler l'inégalité triangulaire.

3. On a $\forall x \in E, x \neq 0$,

$$\|u(x)\| = |||x||| \|u(x/||x||)\| \leq ||x|| |||u|||$$

(et c'est encore valable pour $x = 0$ sans passer par l'étape intermédiaire). On a donc

$$\forall x \in E, \|u \circ v(x)\| = \|u(v(x))\| \leq |||u||| \|v(x)\| \leq |||u||| |||v|||$$

ce qui montre, en passant à la borne supérieure pour x variant dans la sphère unité de E , que $||| \cdot |||$ est une norme sous-multiplicative.

On prouve alors par récurrence que

$$\forall u \in \mathcal{L}(E), \forall k \in \mathbb{N}, |||u^k||| \leq |||u|||^k$$

- Initialisation : $u^0 = \text{Id}$ et $\forall x \in E, \|u^0(x)\| = \|x\|$. Ainsi $|||u^0||| = 1 = |||u|||^0$.
- Hérédité : supposons le résultat vrai à un rang $k \geq 0$. On a alors avec la sous-multiplicativité

$$|||u^{k+1}||| = |||u^k \circ u||| \leq |||u^k||| |||u|||$$

et le résultat au rang k donne celui au rang $k + 1$.

Remarque : en général, si N est une norme sous-multiplicative sur l'algèbre A , l'inégalité $N(x^k) \leq N(x)^k$ n'est valable que si $k \geq 1$. Ici, le résultat est valable à partir du rang 0 car la norme du neutre multiplicatif est plus petite que 1 (elle vaut même 1). Rappelons que c'est à partir de cette propriété de sous-multiplicativité que l'on peut montrer l'existence de e^u pour l'endomorphisme u en montrant une convergence absolue de série, qui entraîne la convergence en dimension finie.

B. Etude de la stabilité en 0 du système linéaire

4. $\mathbb{C}$ étant algébriquement clos (théorème de D'Alembert-Gauss), le polynôme caractéristique de a est scindé. En notant $\lambda_1, \dots, \lambda_r$ ses racines distinctes (qui sont les valeurs propres de a) et $m_1, \dots, m_r$ leurs multiplicités respectives (qui sont des éléments de $\mathbb{N}^*$), on a donc (le polynôme caractéristique étant unitaire)

$$\chi_a = \prod_{k=1}^r (X - \lambda_i)^{m_i}$$

Les polynômes $(X - \lambda_i)^{m_i}$ étant deux à deux premiers entre eux, le théorème de décomposition des noyaux donne

$$\ker(\chi_a(a)) = \bigoplus_{k=1}^r \ker((a - \lambda_i \text{Id})^{m_i})$$

Enfin, le théorème de Cayley-Hamilton indique que χ_a annule a et l'espace précédent est égal à $\mathbb{C}^n$.

$$\mathbb{C}^n = \bigoplus_{k=1}^r \ker((a - \lambda_i \text{Id})^{m_i})$$

Rappel : les E_i s'appellent les sous-espaces caractéristiques de l'endomorphisme a . On doit savoir que dans le cas d'un endomorphisme trigonalisable, la dimension du sous-espace caractéristique associé à une valeur propre est égale à la multiplicité de la valeur propre.

5. On fixe $i \in \llbracket 1, r \rrbracket$. p_i est linéaire et donc continue. Elle est bornée sur la sphère unité de $\mathbb{C}^n$ et on peut poser

$$C_i = \sup_{x \in \mathbb{C}^n, \|x\|=1} \|p_i(x)\|$$

On se donne $u \in \mathcal{L}(E_i)$. Pour tout $x \in \mathbb{C}^n$, on a $up_i(x) \in E_i$ et donc $q_i up_i(x) = up_i(x)$. Comme $\|u(y)\| \leq \|u\|_i \|y\|$, on a donc

$$\|q_i up_i(x)\| = \|up_i(x)\| \leq \|u\|_i \|p_i(x)\|$$

Par choix de C_i , si $\|x\| = 1$ on a $\|q_i up_i(x)\| \leq C_i \|u\|_i$. Par définition de la norme sur $\mathcal{L}(\mathbb{C}^n)$, $\|q_i up_i\|_c \leq C_i \|u\|_i$.

$$\forall u \in \mathcal{L}(E_i), \|q_i up_i\|_c \leq C_i \|u\|_i$$

Remarque : ici, on veut un réel C_i qui soit valable pour tout u . C_i peut dépendre des données du problème, comme a , mais pas de u . Il est donc préférable quand on rédige de donner la valeur de C_i avant d'introduire un endomorphisme u . Bien sûr, lors de la recherche au brouillon, on se donne u et on essaye de majorer la norme de $q_i up_i$, et donc de majorer $\|q_i up_i(x)\|$ par une quantité indépendante de x dans la sphère unité, et c'est ce qui nous donne l'idée d'un bon C_i .

6. a et $(a - \lambda_i \text{Id})^{m_i}$ commutent (deux polynômes en a commutent) et donc

$$\forall x \in \mathbb{C}^n, (a - \lambda_i \text{Id})^{m_i}(a(x)) = a((a - \lambda_i \text{Id})^{m_i}(x))$$

En particulier, si $x \in E_i$ le membre de droite étant nul, le membre de gauche l'est aussi et

$$\forall i \in \llbracket 1, r \rrbracket, E_i \text{ est stable par } a$$

Remarquons que si $x \in E_i$, $q_i(x) = x$ puis $a q_i(x) = a(x) \in E_i$ et donc $p_i a q_i(x) = a(x)$. a_i est donc la restriction de a à E_i .

7. Soit $x_j \in E_j$. On a $p_i q_j(x_j) = p_i(x_j)$ qui est nul si $i \neq j$ et vaut x_j si $j = i$ ainsi

$$p_i q_j : E_j \rightarrow E_i = \begin{cases} 0 & \text{si } j \neq i \\ \text{Id}_{E_j} & \text{si } j = i \end{cases}$$

Soit $x \in E$ que l'on décompose en $x = x_1 + \dots + x_r$ avec $x_i \in E_i$. On a $p_i(x) = x_i \in E_i$ et donc $q_i p_i(x) = x_i$. On en déduit que

$$\sum_{i=1}^r q_i p_i = \text{Id}_{\mathbb{C}^n}$$

8. Pour montrer que deux endomorphismes de $\mathbb{C}^n$ sont égaux, il suffit de montrer qu'ils coïncident sur les E_i puisque ces espaces sont supplémentaires.
On se donne donc $x_j \in E_j$. $p_i(x_j)$ étant nul si $i \neq j$ et valant x_j si $i = j$, on a

$$\sum_{i=1}^r q_i a_i p_i(x_j) = q_j a_j(x_j)$$

a_j étant un endomorphisme de E_j , $a_j(x_j) \in E_j$ est laissé invariant par q_j et

$$\sum_{i=1}^r q_i a_i p_i(x_j) = a_j(x_j) = p_j a_j(x_j) = p_j a(x_j)$$

Comme E_j est stable par a , $a(x_j) \in E_j$ est laissé invariant par p_j et donc

$$\sum_{i=1}^r q_i a_i p_i(x_j) = a(x_j)$$

Avec la remarque initiale, on conclut que

$$\sum_{i=1}^r q_i a_i p_i = a$$

Remarque : on pourrait bien sûr partir d'un x quelconque et le décomposer. Il est cependant bien plus élégant et simple d'adopter la démarche précédente. Ici, l'énoncé introduit une décomposition de $\mathbb{C}^n$ à l'aide de supplémentaires et on privilégie le travail sur ceux-ci. Si on disposait d'une base de $\mathbb{C}^n$, il serait tout aussi naturel de se contenter de travailler avec les éléments de cette base.

9. On montre par récurrence que

$$\forall k \in \mathbb{N}, a^k = \sum_{i=1}^r q_i a_i^k p_i$$

- Initialisation : le résultat est vrai au rang 0 car $a^0 = \text{Id}_{\mathbb{C}^n}$ et avec la question 7.
- Hérédité : supposons le résultat vrai à un rang $k \geq 0$. On a alors

$$\begin{aligned} a^{k+1} &= a^k a \\ &= \left(\sum_{i=1}^r q_i a_i^k p_i \right) \left(\sum_{j=1}^r q_j a_j p_j \right) \\ &= \sum_{1 \leq i, j \leq r} q_i a_i^k p_i q_j a_j p_j \end{aligned}$$

Avec la question 7, les termes dans la somme pour $i \neq j$ sont nuls. Comme $p_i q_i$ est l'identité de E_i , on a alors

$$a^{k+1} = \sum_{i=1}^r q_i a_i^{k+1} p_i$$

ce qui montre le résultat au rang $k + 1$.

On en déduit que pour tout réel t ,

$$e^{ta} = \sum_{k=0}^{\infty} \frac{t^k}{k!} a^k = \sum_{k=0}^{\infty} \sum_{i=1}^r \frac{t^k}{k!} q_i a_i^k p_i$$

Toutes les sommes écrites existant bien,

$$e^{ta} = \sum_{i=1}^r \sum_{k=0}^{\infty} \frac{t^k}{k!} q_i a_i^k p_i = \sum_{i=1}^r q_i \left(\sum_{k=0}^{\infty} \frac{t^k}{k!} a_i^k \right) p_i$$

c'est à dire

$$\forall t \in \mathbb{R}, e^{ta} = \sum_{i=1}^r q_i e^{ta_i} p_i$$

10. Rappelons tout d'abord que si u et v sont deux endomorphismes qui commutent, $e^{u+v} = e^u e^v$. On a ainsi, pour $i \in \llbracket 1, r \rrbracket$ et $t \in \mathbb{R}$ donnés,

$$e^{ta_i} = e^{t\lambda_i \text{Id}_{E_i}} e^{t(a_i - \lambda_i \text{Id}_{E_i})}$$

La sous-multiplicativité prouvée en question 3 donne alors

$$\| \| e^{ta_i} \| \|_i = \| \| e^{t\lambda_i \text{Id}_{E_i}} \| \|_i \| \| e^{t(a_i - \lambda_i \text{Id}_{E_i})} \| \|_i$$

Comme $e^{t\lambda_i \text{Id}_{E_i}} = e^{t\lambda_i} \text{Id}_{E_i}$ et comme l'identité est de norme 1, le premier terme du membre de droite vaut $|e^{\lambda_i t}|$.

$a_i - \lambda_i \text{Id}_{E_i}$ étant nilpotent d'ordre $\leq m_i$ (par définition de E_i), on a

$$e^{t(a_i - \lambda_i \text{Id}_{E_i})} = \sum_{k=0}^{m_i-1} \frac{t^k}{k!} (a_i - \lambda_i \text{Id}_{E_i})^k$$

Par inégalité triangulaire, homogénéité et conséquence de la sous-multiplicativité (question 3)

$$\| \| e^{t(a_i - \lambda_i \text{Id}_{E_i})} \| \|_i \leq \sum_{k=0}^{m_i-1} \frac{|t|^k}{k!} \| \| (a_i - \lambda_i \text{Id}_{E_i}) \| \|_i^k$$

On a finalement

$$\forall i \in \llbracket 1, r \rrbracket, \forall t \in \mathbb{R}, \| \| e^{ta_i} \| \| \leq |e^{\lambda_i t}| \sum_{k=0}^{m_i-1} \frac{|t|^k}{k!} \| \| (a_i - \lambda_i \text{Id}_{E_i}) \| \|_i^k$$

Rappel : le programme donne peu de résultat sur les exponentielles de matrice ou d'endomorphisme. Outre la définition et la propriété rappelée ici, il faut savoir calculer l'exponentielle d'une matrice diagonale et savoir relier les exponentielles de deux matrices semblables. Le résultat sur la commutation permet de montrer que e^A est inversible d'inverse e^{-A} . On pourra aussi montrer en exercice que $\det(e^A) = e^{\text{Tr}(A)}$ (utiliser une trigonalisation en considérant A comme matrice complexe). Dans le programme, l'exponentielle d'endomorphisme est introduite afin de donner une expression des solutions d'un système différentiel à coefficients constants, ce qui servira après la question 12.

11. Soit $t \in \mathbb{R}$. On passe à la norme dans la question 9 et on utilise l'inégalité triangulaire

$$|||e^{ta}|||_c \leq \sum_{i=1}^r |||q_i e^{ta_i} p_i|||_c$$

Les questions 5 puis 10 donnent

$$|||q_i e^{ta_i} p_i|||_c \leq C_i |e^{\lambda_i t}| \sum_{k=0}^{m_i-1} \frac{|t|^k}{k!} |||(a_i - \lambda_i \text{Id}_{E_i})|||_i^k$$

On a $|e^{\lambda_i t}| = e^{\text{Re}(\lambda_i)t}$. On remplace ci-dessus et on somme les inégalités :

$$|||e^{ta}|||_c \leq \sum_{i=1}^r \sum_{k=0}^{m_i-1} C_i e^{\text{Re}(\lambda_i)t} \frac{|t|^k}{k!} |||(a_i - \lambda_i \text{Id}_{E_i})|||_i^k$$

Chaque terme $e^{\text{Re}(\lambda_i)t}$ est majoré par la somme de ceux-ci (puisque'ils sont positifs) :

$$|||e^{ta}|||_c \leq \left(\sum_{k=1}^r e^{\text{Re}(\lambda_k)t} \right) \sum_{i=1}^r \sum_{k=0}^{m_i-1} C_i \frac{|t|^k}{k!} |||(a_i - \lambda_i \text{Id}_{E_i})|||_i^k$$

La double somme du membre de droite est polynomiale en $|t|$ et à coefficients réels. On a ainsi montré que

$$\boxed{\exists P \in \mathbb{R}[X], \forall t \in \mathbb{R}, |||e^{ta}|||_c \leq P(|t|) \sum_{k=1}^r e^{\text{Re}(\lambda_k)t}}$$

Remarque : il faut savoir que si z est un nombre complexe, on a $|e^z| = e^{\text{Re}(z)}$.

12. Notons $\phi_r : A \mapsto |||u_A|||_r$ et $\phi_c : A \mapsto |||v_A|||_c$. Les propriétés de $|||\cdot|||_r$ et $|||\cdot|||_c$ se transmettent à ϕ_r et ϕ_c (positivité, homogénéité, axiome de séparation et inégalité triangulaire) et on dispose ainsi de deux normes sur $\mathcal{M}_n(\mathbb{R})$. L'espace étant de dimension finie, ces normes sont équivalentes et

$$\exists C, \forall A \in \mathcal{M}_n(\mathbb{R}), \phi_r(A) \leq C \phi_c(A)$$

En particulier, on a

$$\forall A \in \mathcal{M}_n(\mathbb{R}), \forall t \in \mathbb{R}, \phi_r(e^{tA}) \leq C \phi_c(e^{tA})$$

Mais si u est un endomorphisme du $\mathbb{K}$ -espace vectoriel E de dimension finie et si $\mathcal{B}$ est une base de E , la matrice de e^{tu} dans la base $\mathcal{B}$ est égale à e^{tA} où $A = \text{Mat}(u, \mathcal{B})$.

A représentant simultanément u_A et v_A dans la base canonique (selon l'espace considéré), on en déduit que

$$\boxed{|||e^{tu_A}|||_r \leq C |||e^{tv_A}|||_c}$$

Remarque : on peut en fait choisir $C = 1$. En effet, considérons $x \in \mathbb{R}^n$ tel que $\|x\| = 1$. Si on considère x comme élément de $\mathbb{C}^n$, on a toujours $\|x\| = 1$ (on munit $\mathbb{C}^n$ de la norme $(z_1, \dots, z_n) \mapsto \sqrt{|z_1|^2 + \dots + |z_n|^2}$). Par ailleurs, u_A est la restriction à $\mathbb{R}^n$ de v_A et donc $e^{tu_A}(x) = e^{tv_A}(x)$. Ainsi $\|e^{tu_A}(x)\| \leq \|e^{tv_A}\|_c$ et $|||e^{tu_A}|||_r \leq |||e^{tv_A}|||_c$.

13. On notera bien sur que $u = u_A$.

Soit $x_0 \in \mathbb{R}$. Le cours nous indique que $\forall t \in \mathbb{R}, g_{x_0}(t) = e^{tu_A}(x_0)$.

Supposons que $\forall x_0 \in \mathbb{R}^n, \|g_{x_0}(t)\| \rightarrow 0$ quand $t \rightarrow +\infty$. Soit λ une valeur propre complexe de A et Z_0 un vecteur propre associé. On a donc $e^{tA}Z_0 = e^{t\lambda}Z_0$.

On note $z_0 \in \mathbb{C}^n$ l'élément représenté par Z_0 dans la base canonique de $\mathbb{C}^n$.

On note X_0 et Y_0 les parties réelle et imaginaire de Z_0 que l'on interprète comme des éléments x_0, y_0 de $\mathbb{R}^n$ et on a

$$e^{tA}Z_0 = e^{tA}X_0 + ie^{tA}Y_0$$

Comme A est une matrice réelle, ceci s'écrit

$$e^{\lambda t}z_0 = e^{tu_A}(x_0) + ie^{tu_A}(y_0) = g_{x_0}(t) + ig_{y_0}(t)$$

En munissant $\mathbb{C}^n$ de la norme $N(x + iy) = \max(\|x\|, \|y\|)$, on voit que $|e^{\lambda t}|N(z_0) = e^{\operatorname{Re}(\lambda)t}N(z_0)$ est de limite nulle en $+\infty$ et comme $N(z_0) \neq 0, \operatorname{Re}(\lambda) < 0$.

Réciproquement, on suppose que toutes les valeurs propres complexes de A sont de partie réelle < 0 . La question 11 donne par croissances comparées que $\|e^{tu_A}\|_c \rightarrow 0$ quand $t \rightarrow +\infty$.

La question 12 donne alors $\|e^{tu_A}\|_r \rightarrow 0$ quand $t \rightarrow +\infty$.

En particulier, pour tout choix de $x_0 \in \mathbb{R}^n, \|g_{x_0}(t)\| \leq \|e^{tu_A}\|_r \|x_0\| \rightarrow 0$ quand $t \rightarrow +\infty$.

On a montré que

$$\forall x_0 \in \mathbb{R}^n, \lim_{t \rightarrow +\infty} \|g_{x_0}(t)\| = 0 \iff \operatorname{Sp}(A) \subset \mathbb{R}_-^* + i\mathbb{R}$$

14. Les questions 11 et 12 donnent l'existence de $C \in \mathbb{R}$ et de $P \in \mathbb{R}[X]$ tels que

$$\forall t \in \mathbb{R}^+, \|e^{tu}\|_r \leq CP(t) \sum_{i=1}^r e^{t\operatorname{Re}(\lambda_i)}$$

Le maximum des $\operatorname{Re}(\lambda_i)$ est < 0 . On peut donc trouver $\alpha > 0$ tel que $\forall i \in \llbracket 1, r \rrbracket, \operatorname{Re}(\lambda_i) < -\alpha$. On a

$$\forall t \in \mathbb{R}^+, \|e^{tu}\|_r \leq Ce^{-\alpha t} \sum_{i=1}^r e^{t(\alpha + \operatorname{Re}(\lambda_i))} P(t)$$

La somme est le terme général d'une fonction continue sur $\mathbb{R}^+$ et de limite nulle en $+\infty$ par croissances comparées. Cette somme est plus petite que 1 sur $[t_0, +\infty[$ pour t_0 assez grand et est bornée sur le compact $[0, t_0]$. C'est donc une fonction bornée sur $\mathbb{R}^+$. On a ainsi montré que

$$\exists C, \alpha > 0, \forall t \in \mathbb{R}^+, \|e^{tu}\|_r \leq C_2 e^{-\alpha t}$$

Le résultat disant qu'une fonction continue sur $\mathbb{R}^+$ et admettant une limite finie en l'infini n'est a priori pas au programme. A ce niveau du problème, on peut penser que sa preuve ne sera pas exigée. A la lecture de certains rapports, il semble cependant plus sage d'évoquer le théorème de compacité sous-jacent.

On en déduit avec l'expression $g_{x_0}(t) = e^{tu}(x_0)$ que

$$\forall t \in \mathbb{R}^+, \|g_{x_0}(t)\| \leq C_2 e^{-\alpha t} \|x_0\|$$

C. Démonstration du théorème de Liapounov

15. La question 14 fournit des constantes C_2 et α strictement négatives telles que

$$\forall t \in \mathbb{R}^+, |||e^{ta}||| \leq C_2 e^{-\alpha t}$$

On a alors, par inégalité de Cauchy-Schwarz,

$$\forall t \in \mathbb{R}^+, |\langle e^{ta}(x) | e^{ta}(y) \rangle| \leq \|e^{ta}(x)\| \|e^{ta}(y)\| \leq C_2^2 e^{-2\alpha t} \|x\| \|y\|$$

La fonction majorante est intégrable au voisinage de $+\infty$. Comme $t \mapsto \langle e^{ta}(x) | e^{ta}(y) \rangle$ est continue sur $\mathbb{R}^+$, elle est intégrable sur $\mathbb{R}^+$ et b est bien définie.

La symétrie de b découle de celle du produit scalaire sur $\mathbb{R}^n$.

La bilinéarité de b découle de la linéarité de e^{ta} et du passage à l'intégrale ainsi que de la bilinéarité du produit scalaire sur $\mathbb{R}^n$.

Enfin, si $b(x, x) = 0$ alors $t \mapsto \|e^{ta}(x)\|^2$ est positive continue et d'intégrale nulle. C'est donc la fonction nulle et $\forall t \geq 0$, $e^{ta}(x) = 0$. C'est en particulier vrai en $t = 0$ et cela donne $x = 0$ et le caractère défini positif de b .

b définit un produit scalaire sur $\mathbb{R}^n$

16. D'après le cours (b étant bilinéaire), q est différentiable en tout point et (avec la symétrie de b)

$$\forall x, dq(x) : h \mapsto b(x, h) + b(h, x) = 2b(x, h)$$

Remarque : on le retrouve en écrivant que $b(x + h, x + h) = b(x) + 2b(x, h) + b(h, h)$. Par continuité de b multilinéaire, il existe c tel que pour tout h , $|b(h, h)| \leq c\|h\|^2$ ce qui donne $b(h, h) = o(h)$. On conclut par linéarité de $h \mapsto b(x, h)$. Rappelons au passage que la continuité d'une application bilinéaire ϕ équivaut à l'existence d'une constante c telle que $\forall x, y$, $|\phi(x, y)| \leq c\|x\|\|y\|$, les normes étant celle considérées sur les différents ensembles.

En particulier, $dq(x)(a(x)) = 2b(x, a(x))$. Or, $e^{ta}(a(x)) = a(e^{ta}(x))$ (e^{ta} et a commutent) est la dérivée de $\psi_x : t \mapsto e^{ta}(x)$ et donc

$$b(x, a(x)) = \int_0^\infty \langle \psi_x(t) | \psi'_x(t) \rangle dt$$

Le terme dans l'intégrale est la dérivée de $t \mapsto \frac{1}{2} \|\psi_x(t)\|^2$ et ainsi

$$b(x, a(x)) = \frac{1}{2} [\|\psi_x(t)\|^2]_0^{+\infty}$$

a étant à valeurs propres complexes de parties réelles < 0 , la question 14 indique que ψ_x est de limite nulle en $+\infty$ (ψ_x est la solution du système linéaire $y' = a(y)$ avec la donnée initiale $\psi_x(0) = x$). Elle vaut x en 0 et donc

$$b(x, a(x)) = -\frac{1}{2} \|x\|^2$$

On a finalement

$$\forall x \in \mathbb{R}^n, dq(x)(a(x)) = 2b(x, a(x)) = -\|x\|^2$$

Remarque : on utilise ici que la solution générale du système $y' = a(y)$ quand a est linéaire est $t \mapsto e^{ta}(x)$.

17. Le cours permet de dériver $t \mapsto b(u(t), v(t))$ en $t \mapsto b(u'(t), v(t)) + b(u(t), v'(t))$ quand u, v sont dérivables. Ainsi

$$\begin{aligned}(q \circ f_{x_0})'(t) &= 2b(f'_{x_0}(t), f_{x_0}(t)) \\ &= 2b(\varphi(x_0(t)), f_{x_0}(t)) \\ &= 2b(\varepsilon(f_{x_0})(t) + a(f_{x_0}(t)), f_{x_0}(t)) \\ &= 2b(f_{x_0}(t), \varepsilon(f_{x_0})(t)) + 2b(f_{x_0}(t), a(f_{x_0}(t)))\end{aligned}$$

Avec la seconde égalité de la question 16, on en conclut que

$$(q \circ f_{x_0})'(t) = -\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0})(t))$$

18. Par inégalité de Cauchy-Schwarz utilisée avec le produit scalaire b , on a

$$b(f_{x_0}(t), \varepsilon(f_{x_0})(t)) \leq \sqrt{q(f_{x_0}(t))} \sqrt{q(\varepsilon(f_{x_0})(t))}$$

$x \mapsto \sqrt{q(x)}$ et $\|\cdot\|$ sont deux normes sur $\mathbb{R}^n$ qui est de dimension finie et il existe des constantes K_1 et K_2 strictement positives telles que

$$\forall x \in \mathbb{R}^n, K_1 \sqrt{q(x)} \leq \|x\| \leq K_2 \sqrt{q(x)}$$

On a ainsi

$$-\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0})(t)) \leq -K_1^2 q(f_{x_0}(t)) + 2\sqrt{q(f_{x_0}(t))} \sqrt{q(\varepsilon(f_{x_0})(t))}$$

Par ailleurs, par définition de la différentielle en un point, on a $\varphi(u) = \varphi(0) + a(u) + o(u)$ et donc (comme $\varphi(0) = 0$ et comme on peut travailler avec la norme $\sqrt{q}$)

$$\forall K > 0, \exists r > 0, q(u) \leq r \implies \sqrt{q(\varphi(u) - a(u))} \leq K \sqrt{q(u)}$$

Si $q(f_{x_0}(t))$ est suffisamment petit, on a donc $\sqrt{q(\varepsilon(f_{x_0})(t))} \leq \frac{K_1^2}{4} \sqrt{q(f_{x_0}(t))}$ et ainsi

$$-\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0})(t)) \leq -\frac{K_1^2}{2} q(f_{x_0}(t))$$

$$\boxed{\exists \alpha, \beta > 0, q(f_{x_0}(t)) \leq t \implies -\|f_{x_0}(t)\|^2 + 2b(f_{x_0}(t), \varepsilon(f_{x_0})(t)) \leq -\beta q(f_{x_0}(t))}$$

19. On suppose $q(x_0) < \alpha$ c'est à dire $q(f_{x_0}(0)) < \alpha$. Par continuité de f_{x_0} , il existe t_0 tel que

$$\forall t \in [0, t_0[, q(f_{x_0}(t)) \leq \alpha$$

Avec les questions 17 et 18, et en notant $\theta : t \mapsto q \circ f_{x_0}(t)$,

$$\forall t \in [0, t_0[, \theta'(t) \leq -\beta \theta(t)$$

$t \mapsto \theta(t)e^{\beta t}$ est donc décroissante sur $[0, t_0[$ (sa dérivée est négative sur l'intervalle) et donc

$$\forall t \in [0, t_0[, \theta(t) \leq \theta(0) = q(x_0)$$

ou encore

$$\forall t \in [0, t_0[, \quad q(f_{x_0}(t)) \leq e^{-\beta t} q(x_0)$$

On veut voir que cette majoration a lieu sur tout $\mathbb{R}^+$.

L'ensemble $\{T \in \mathbb{R}^+, \forall t \in [0, T[: \theta(t) \leq e^{-\beta t} q(x_0)\}$ est une partie non vide de $\mathbb{R}$. Supposons, par l'absurde, qu'elle est majorée. Elle admet alors une borne supérieure T . Par continuité de θ , T est encore dans l'ensemble et $q(f_{x_0}(T)) \leq e^{-\beta T} q(x_0) < \alpha$. On peut alors comme ci-dessus montrer que l'inégalité reste vraie sur un voisinage à droite de T ce qui contredit le fait que T est la borne supérieure de l'ensemble.

L'ensemble est donc non majoré et l'inégalité a lieu pour tout réel $t \geq 0$ (puisque'elle est vraie sur $[0, t]$ pour des t arbitrairement grands).

$$\boxed{q(x_0) < \alpha \implies (\forall t \geq 0, \quad q(f_{x_0}(t)) \leq e^{-\beta t} q(x_0))}$$

20. Par équivalence des normes, il existe des constantes K_1 et K_2 telles que

$$\forall x \in \mathbb{R}^n, \quad K_1 \sqrt{q(x)} \leq \|x\| \leq K_2 \sqrt{q(x)}$$

posons $\tilde{\alpha} = K_1 \sqrt{\alpha}$. On a alors

$$\|x_0\| < \tilde{\alpha} \implies q(x_0) \leq \frac{\|x_0\|^2}{K_1^2} < \alpha$$

En supposant $\|x_0\| < \tilde{\alpha}$, on a alors

$$\|f_{x_0}(t)\| \leq K_2 \sqrt{q(f_{x_0}(t))} \leq K_2 \sqrt{q(x_0)} e^{-\frac{\beta}{2}t} \leq \frac{K_2}{K_1} e^{-\frac{\beta}{2}t} \|x_0\|$$

En posant $C = \frac{K_2}{K_1}$, on a

$$\boxed{\forall x_0 \in B(0, \tilde{\alpha}), \quad \forall t \in \mathbb{R}, \quad \|f_{x_0}(t)\| \leq C e^{-\frac{\beta}{2}t} \|x_0\|}$$



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2023

DEUXIÈME ÉPREUVE DE MATHÉMATIQUES

Durée de l'épreuve : 4 heures

L'usage de la calculatrice ou de tout dispositif électronique est interdit.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

MATHÉMATIQUES II - MP

L'énoncé de cette épreuve comporte 5 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Fonction de Wallis

Préliminaires

Dans tout le sujet, l'intervalle $] -1, +\infty[$ de $\mathbf{R}$ est appelé I et σ et f sont les fonctions, de $\mathbf{R}$ dans $\mathbf{R}$, définies par :

$$\sigma(x) = \sum_{k=1}^{+\infty} \frac{x^k}{k^2}$$

et

$$f(x) = \int_0^{\pi/2} (\sin(t))^x dt.$$

On se propose, dans cette épreuve, d'étudier f (domaine de définition, régularité, variations, convexité, développement éventuel en série entière,...) puis, dans la dernière partie, de montrer qu'elle est la seule fonction numérique à vérifier certaines propriétés.

1 Calcul de $\sigma(1)$

1 ▷ Déterminer le domaine de définition de σ puis justifier que σ est continue sur celui-ci.

2 ▷ Exhiber deux nombres réels α et β tels que :

$$\forall n \in \mathbf{N}^*, \int_0^\pi (\alpha t^2 + \beta t) \cos(nt) dt = \frac{1}{n^2},$$

puis vérifier que si $t \in]0, \pi]$, alors :

$$\forall n \in \mathbf{N}^*, \sum_{k=1}^n \cos(kt) = \frac{\sin\left(\frac{(2n+1)t}{2}\right)}{2 \sin\left(\frac{t}{2}\right)} - \frac{1}{2}.$$

3 ▷ Justifier que, si φ est une application de classe $\mathcal{C}^1$ de $[0, \pi]$ dans $\mathbf{R}$, alors

$$\lim_{x \rightarrow +\infty} \int_0^\pi \varphi(t) \sin(xt) dt = 0,$$

et en conclure que

$$\sigma(1) = \frac{\pi^2}{6}.$$

2 Équivalents

4 ▷ Déterminer le domaine de définition de f puis vérifier que

$$\forall x \in I, (x+1)f(x) = (x+2)f(x+2). \quad (1)$$

5 ▷ Justifier que f est de classe $\mathcal{C}^2$, décroissante et convexe sur I .

6 ▷ Donner un équivalent simple de $f(x)$ lorsque x tend vers -1 .

7 ▷ Montrer que pour tout entier naturel n ,

$$f(n)f(n+1) = \frac{\pi}{2(n+1)}$$

puis que :

$$f(x) \underset{x \rightarrow +\infty}{\sim} \sqrt{\frac{\pi}{2x}}.$$

8 ▷ Représenter graphiquement f en exploitant au mieux les résultats précédents.

3 Développement en série entière

Si $n \in \mathbf{N}$, on note D_n l'intégrale généralisée $\int_0^{\pi/2} (\ln(\sin(t)))^n dt$.

9 ▷ Justifier que, si $n \in \mathbf{N}$, l'intégrale généralisée D_n est convergente, puis montrer que

$$D_1 = \int_0^{\pi/2} \ln(\cos(t)) dt.$$

10 ▷ Calculer $f'(0)$ et $f'(1)$.

11 ▷ Vérifier que si $n \in \mathbf{N}^*$, alors

$$(-1)^n D_n = \int_0^{+\infty} \frac{u^n}{\sqrt{e^{2u} - 1}} du,$$

puis que

$$D_n \underset{n \rightarrow +\infty}{\sim} (-1)^n n!$$

12 ▷ Démontrer que f est développable en série entière sur $] -1, 1[$.

4 Convergence de suite de fonctions

On se propose dans cette partie de calculer $f''(0)$. Dans ce but, on considère deux nombres réels strictement positifs a et b , et on pose

$$\rho = \frac{b-a}{b+a}.$$

On appelle Ψ l'application de $\mathbf{R}$ dans $\mathbf{R}$ définie par :

$$\forall x \in \mathbf{R}, \Psi(x) = \ln(a^2 \cos^2 x + b^2 \sin^2 x).$$

13 ▷ Montrer que Ψ est de classe $\mathcal{C}^1$ sur $\mathbf{R}$, puis que pour tout $x \in \mathbf{R}$,

$$\Psi'(x) = 4 \sum_{k=1}^{+\infty} \rho^k \sin(2kx).$$

14 ▷ En déduire que pour tout $x \in \mathbf{R}$,

$$\Psi(x) = 2 \ln \left(\frac{a+b}{2} \right) - 2 \sum_{k=1}^{+\infty} \frac{\cos(2kx)}{k} \rho^k.$$

15 ▷ En conclure que

$$\int_0^\pi \Psi(x)^2 dx = 4\pi \left(\ln \left(\frac{a+b}{2} \right) \right)^2 + 2\pi\sigma(\rho^2).$$

On définit les suites réelles $(a_n)_{n \in \mathbf{N}^*}$ et $(b_n)_{n \in \mathbf{N}^*}$ par

$$\forall n \in \mathbf{N}^*, a_n = \frac{1}{n+1} \text{ et } b_n = \frac{n}{n+1}.$$

16 ▷ Établir la convergence simple de la suite d'applications $(\Psi_n)_{n \in \mathbf{N}^*}$, de $]0, \pi]$ dans $\mathbf{R}$, définie par :

$$\forall n \in \mathbf{N}^*, \forall t \in]0, \pi], \Psi_n(t) = \ln(a_n^2 \cos^2 t + b_n^2 \sin^2 t).$$

En déduire $f''(0)$.

5 Convexité logarithmique

Une application h d'un intervalle non trivial J de $\mathbf{R}$ dans $\mathbf{R}$ est dite ln-convexe si, et seulement si, elle est à valeurs dans $\mathbf{R}_+^*$ et $\ln \circ h$ est convexe sur J .

17 ▷ Vérifier que f est une application de I dans $\mathbf{R}$ ln-convexe.

On souhaite désormais déterminer toutes les applications de I dans $\mathbf{R}$ qui sont ln-convexes et qui vérifient la propriété (1), voir question 4.

On appelle $\tilde{f}$ l'application de $\mathbf{R}_+$ dans $\mathbf{R}$, définie par :

$$\forall x \in \mathbf{R}^+, \tilde{f}(x) = \ln(f(2x)).$$

18 ▷ Montrer que

$$\forall p \in \mathbf{N}^*, \forall x \in \mathbf{R}_+, \tilde{f}(x+p) = \tilde{f}(x) + \sum_{k=0}^{p-1} \ln \left(\frac{2x+2k+1}{2x+2k+2} \right).$$

19 ▷ On suppose ici que $x \in \mathbf{R}_+^*$, $(n, p) \in (\mathbf{N}^*)^2$ et $x \leq p$. Vérifier que

$$\tilde{f}(n) - \tilde{f}(n-1) \leq \frac{\tilde{f}(n+x) - \tilde{f}(n)}{x} \leq \frac{\tilde{f}(n+p) - \tilde{f}(n)}{p}$$

et que $(\tilde{f}(n+x) - \tilde{f}(n))$ admet une limite lorsque n tend vers $+\infty$.

- 20** ▷ En conclure que f est la seule application de I dans $\mathbf{R}$, qui soit ln-convexe, qui vérifie (1) et telle que

$$f(0) = \frac{\pi}{2}.$$

- 21** ▷ Plus généralement, déterminer, si $T \in \mathbf{R}_+^*$, toutes les applications g de $] -T, +\infty[$ dans $\mathbf{R}$, ln-convexes et vérifiant

$$\forall t \in] -T, +\infty[, (t+T)g(t) = (t+2T)g(t+2T).$$

- 22** ▷ Existe-t-il une application h , de $\mathbf{R}$ dans $\mathbf{R}$ et ln-convexe, vérifiant

$$\forall t \in \mathbf{R}, (t+T)h(t) = (t+2T)h(t+2T) ?$$

SOLUTION

Calcul de $\sigma(1)$

1. Si $|x| > 1$, on a $\frac{|x^k|}{k^2} \rightarrow +\infty$ quand $k \rightarrow +\infty$ et la série de terme général $\frac{x^k}{k^2}$ diverge grossièrement.

Si $|x| \leq 1$, on a $\frac{|x^k|}{k^2} \leq \frac{1}{k^2}$ qui est le terme général d'une série qui converge. Par comparaison des séries positives, la série de terme général $\frac{x^k}{k^2}$ converge absolument et donc converge ($\mathbb{R}$ est de dimension finie).

Ainsi σ est définie sur $[-1, 1]$.

$u_k : x \mapsto \frac{x^k}{k^2}$ est continue sur $[-1, 1]$ et ce qui précède montre que $\|u_k\|_{\infty, [-1, 1]} \leq \frac{1}{k^2}$ et que $\sum (u_k)$ converge normalement sur $[-1, 1]$. Par théorème de continuité des sommes de séries de fonctions,

σ est définie et continue sur $[-1, 1]$

Il est possible d'aborder cette question par le prisme des séries entières. Le rayon de convergence vaut 1 et le domaine D vérifie $] -1, 1[\subset D \subset [-1, 1]$. On a continuité (et même classe C^∞) sur $] -1, 1[$ comme somme de série entière. On peut utiliser le lemme d'Abel radial pour obtenir la convergence en 1 et -1 et la continuité en ces points. L'approche avec le théorème sur les séries de fonctions me semble cependant plus facile.

2. Soit $n \in \mathbb{N}^*$. Une intégration par parties donne

$$\int_0^\pi t \cos(nt) dt = \left[\frac{t}{n} \sin(nt) \right]_0^\pi - \frac{1}{n} \int_0^\pi \sin(nt) dt = \frac{1}{n^2} [\cos(nt)]_0^\pi = \frac{(-1)^n - 1}{n^2}$$

Et de même

$$\int_0^\pi t^2 \cos(nt) dt = -\frac{2}{n} \int_0^\pi t \sin(nt) dt = \left[\frac{2t}{n^2} \cos(nt) \right]_0^\pi - \frac{2}{n^2} \int_0^\pi \cos(nt) dt = 2\pi \frac{(-1)^n}{n^2}$$

On en déduit que

$$\forall n \in \mathbb{N}^*, \int_0^\pi \left(\frac{1}{2\pi} t^2 - t \right) \cos(nt) dt = \frac{1}{n^2}$$

Attention à la logique du raisonnement. On ne demande que l'existence de α et β et aucune unicité. Il s'agit donc bien de vérifier une propriété après avoir choisi judicieusement les deux réels. Il est certainement possible de partir de α et β quelconques et, par une intégration par parties comme ci-dessus, de montrer que

$$\int_0^\pi (\alpha t^2 + \beta t) \cos(nt) dt = \frac{(-1)^n (2\pi\alpha + \beta) - \beta}{n^2}$$

Puis de dire qu'il suffit de choisir $\alpha = \frac{1}{2\pi}$ et $\beta = -1$. Mais attention, ces égalités ne sont pas une conclusion. Il est vrai que SI la quantité précédente est égale à $1/n^2$ alors il faut choisir α et β ainsi (écrire les égalités pour $n = 1$ et $n = 2$ et résoudre le système). Mais c'est bien la seule réciproque qui nous intéresse. Ainsi, si vous raisonnez comme ci-dessus, il faut travailler par équivalence (ce qui n'est cependant pas conseillé) ou bien

parler d'analyse et synthèse et sans oublier cette dernière.

Montrons par récurrence sur n que

$$\forall t \in]0, \pi], \sum_{k=1}^n \cos(kt) = \frac{\sin\left(\frac{(2n+1)t}{2}\right)}{2 \sin\left(\frac{t}{2}\right)} - \frac{1}{2}$$

On notera $C_n(t)$ le terme du membre de gauche.

- Initialisation : soit $t \in]0, \pi]$. $C_1(t) = \cos(t)$ et par ailleurs $\sin(a) - \sin(b) = 2 \cos\left(\frac{a+b}{2}\right) \sin\left(\frac{a-b}{2}\right)$ qui donne

$$\sin\left(\frac{3t}{2}\right) - \sin\left(\frac{t}{2}\right) = 2 \cos(t) \sin\left(\frac{t}{2}\right)$$

Comme $\sin(t/2) \neq 0$ on peut diviser par ce terme et obtenir le résultat au rang $n = 1$.

- Hérédité : supposons le résultat vrai à un rang $n \geq 1$. Soit $t \in]0, \pi]$. On a alors

$$C_{n+1}(t) = C_n(t) + \cos((n+1)t) = \frac{\sin\left(\frac{(2n+1)t}{2}\right) + 2 \cos((n+1)t) \sin\left(\frac{t}{2}\right)}{2 \sin\left(\frac{t}{2}\right)} - \frac{1}{2}$$

Par la même formule que plus haut $\sin\left(\frac{(2n+3)t}{2}\right) - \sin\left(\frac{(2n+1)t}{2}\right) = 2 \cos((n+1)t) \sin\left(\frac{t}{2}\right)$ et on obtient avec ce qui précède le résultat au rang $n+1$.

Le résultat est en fait aussi vrai au rang 0 avec la convention usuelle qu'une somme vide est nulle. Ceci simplifie l'initialisation. Cependant, comme la formule trigonométrique importante sert aussi dans l'hérédité, on ne gagne pas grand chose avec cette remarque. On aurait aussi pu aborder cette question en disant que $\cos(kt) = \operatorname{Re}(e^{ikt})$ et le calcul de la somme se ramène à celui de la partie réelle d'une somme géométrique de raison différente de 1.

3. On suppose $\varphi \in C^1([0, \pi], \mathbb{R})$. Une intégration par parties donne pour $x \neq 0$:

$$\int_0^\pi \varphi(t) \sin(xt) dt = \left[-\frac{\varphi(t)}{x} \cos(xt) \right]_0^\pi + \frac{1}{x} \int_0^\pi \varphi'(t) \cos(xt) dt$$

φ et φ' sont continues sur $[0, \pi]$ et y sont donc bornées. Notons M et M' leurs normes infinies sur ce segment. Une majoration grossière donne

$$\left| \int_0^\pi \varphi(t) \sin(xt) dt \right| \leq \frac{2M}{x} + \frac{M'\pi}{x}$$

Le majorant est de limite nulle et ainsi, par domination,

$$\forall \varphi \in C^1([0, \pi], \mathbb{R}), \lim_{x \rightarrow +\infty} \int_0^\pi \varphi(t) \sin(xt) dt = 0$$

Ce résultat est connu sous le nom de "lemme de Riemann-Lebesgue". Le résultat reste valable avec la seule hypothèse que f est continue par morceaux. La preuve est alors un peu différente : on prouve le résultat pour les fonctions en escalier sur $[0, \pi]$ puis on utilise le fait que toute fonction continue par morceaux f est arbitrairement proche d'une fonction en escalier φ au sens de la norme infinie. Il s'agit alors d'un raisonnement "par densité".

Notons aussi qu'on pourrait utiliser le théorème de Weierstrass (qui permet d'approcher uniformément sur un segment une fonction continue par des fonctions polynomiales et donc de classe C^1) pour prouver le résultat dans le cas où φ est continue. Ceci simplifierait la seconde partie de la question.

En sommant les relations de la question 2, on trouve que

$$\sum_{k=1}^n \frac{1}{k^2} = \sum_{k=1}^{\infty} \int_0^{\pi} \left(\frac{1}{2\pi} t^2 - t \right) \cos(nt) \, dt$$

La somme étant finie, on peut intervertir les symboles somme et intégrale et la seconde égalité de la question 2 donne alors

$$\sum_{k=1}^n \frac{1}{k^2} = \int_0^{\pi} \left(\frac{1}{2\pi} t^2 - t \right) \left(\frac{\sin\left(\frac{(2n+1)t}{2}\right)}{2 \sin\left(\frac{t}{2}\right)} - \frac{1}{2} \right) dt$$

Posons $\varphi : t \in]0, \pi] \mapsto \frac{\frac{t^2}{2\pi} - t}{2 \sin(\frac{t}{2})}$. φ est de classe C^1 sur $]0, \pi]$ et (puisque $\sin(u) \sim u$ quand $u \rightarrow 0$)

$$\varphi(t) \underset{t \rightarrow 0}{\sim} \frac{-t}{t} = -1$$

On obtient donc une fonction continue sur $[0, \pi]$ en posant $\varphi(0) = 1$.

On a de plus

$$\forall t \in]0, \pi], \varphi'(t) = \frac{2(-1 + \frac{t}{\pi}) \sin(t/2) + (t - \frac{t^2}{\pi}) \cos(t/2)}{4 \sin^2(t/2)}$$

Le dénominateur équivaut à t^2 quand $t \rightarrow 0$.

En utilisant $\sin(u) = u + o(u^2)$ et $\cos(u) = 1 + o(u)$, on obtient que le numérateur équivaut à $\frac{t^2}{2\pi}$.

On a donc $\varphi'(t) \rightarrow \frac{1}{2\pi}$ quand $t \rightarrow 0$.

L'énoncé ne donnant pas le résultat, il ne semble pas utile de détailler les calculs. Donner les développements élémentaires utilisés me semble suffire.

Par théorème de limite de la dérivée, φ est dérivable en 0, $\varphi'(0) = \frac{1}{2\pi}$ et la dérivée est continue en 0.

Observez bien l'ordre des conclusions. Dans ce théorème, on prolonge l'application φ' et ce que dit le théorème, c'est que le fait de pouvoir prolonger entraîne la dérivabilité au point (et la continuité de la dérivée au point). Sans ce théorème, on pourrait très bien imaginer que l'on peut prolonger φ' sans qu'on ait dérivabilité au point (c'est à dire que le théorème indique que la limite de φ' en 0 est bien la dérivée de φ en 0).

Finalement φ est de classe C^1 sur $[0, \pi]$. On peut découper l'intégrale plus haut en deux morceaux (qui existent bien) :

$$\sum_{k=1}^n \frac{1}{k^2} = \int_0^{\pi} \varphi(t) \sin\left(\frac{(2n+1)t}{2}\right) dt - \frac{1}{2} \int_0^{\pi} \left(\frac{1}{2\pi} t^2 - t \right) dt$$

La seconde intégrale du membre de droite vaut $\frac{\pi^2}{6}$ (primitivation immédiate).

La première du membre de droite est de limite nulle quand $n \rightarrow +\infty$ avec le début de la question (et puisque $\frac{2n+1}{2} \rightarrow +\infty$ quand $n \rightarrow +\infty$). Le passage à la limite donne ainsi

$$\sigma(1) = \frac{\pi^2}{6}$$

Cette fois, l'énoncé donne le résultat. Peut-être serait-il plus judicieux de détailler un peu le calcul de l'intégrale (même si le calcul de l'intégrale d'une fonction polynomiale est vraiment immédiat).

Equivalents

4. Soit $x \in \mathbb{R}$. $t \mapsto \sin(t)^x$ est continue sur $]0, \pi/2]$ et équivaut en 0 à t^x . Elle est donc intégrable sur $]0, \pi/2]$ si et seulement si $x > -1$. Comme notre fonction est positive, la condition d'intégrabilité est la même que celle d'existence d'intégrale et

$$f \text{ est définie sur } I =]-1, +\infty[$$

Soit $x \in I$. On a

$$\begin{aligned} f(x+2) &= \int_0^{\pi/2} \sin(t)^x (1 - \cos^2(t)) dt \\ &= f(x) - \int_0^{\pi/2} \sin(t)^x \cos(t) \cos(t) dt \end{aligned}$$

Dans la dernière intégrale, on effectue une intégration par parties en primitivant $\sin(t)^x \cos(t)$ en $\frac{1}{x+1} \sin(t)^{x+1}$ et en dérivant $\cos$. $\cos$ et la primitive sont de classe C^1 sur $]0, \pi/2]$ et leur produit admet une limite finie (nulle en 0). On peut ainsi effectuer l'intégration par parties, on a un terme "entre crochets" nul et on trouve que

$$\int_0^{\pi/2} \sin(t)^x \cos(t) \cos(t) dt = \frac{1}{x+1} \int_0^{\pi/2} \sin(t)^{x+2} dt$$

En injectant dans la relation plus haut et en simplifiant, on obtient

$$\forall x \in I, (x+1)f(x) = (x+2)f(x+2)$$

Quand on effectue une intégration par parties dans une intégrale généralisée, il faut toujours prendre des précautions. Une option est d'annoncer qu'on mène l'intégration par parties "sous réserve de validité". Puis de voir si le terme entre crochets admet une limite finie ce qui valide alors le calcul. Mais dans tous les cas, vous devez montrer que vous avez compris qu'il peut y avoir un souci.

5. On veut utiliser le théorème de régularité des intégrales à paramètres. Posons $g(x, t) = \sin(t)^x$. On a
- $\forall x > -1$, $g(x, \cdot)$ est intégrable sur $]0, \pi/2]$ (question 4).
 - $\forall t \in]0, \pi/2]$, $g(\cdot, t)$ est de classe C^2 de dérivée k -ième $x \mapsto \ln(\sin(t))^k \sin(t)^x$ (pour $k \in \{1, 2\}$).
 - $\forall k \in \{1, 2\}$, $\forall x > -1$, $t \mapsto \ln(\sin(t))^k \sin(t)^x$ est continue sur $]0, \pi/2]$.

- Soient a, b tels que $-1 < a < 0 < b$. On a $\forall x \in [a, b], \forall k \in \{1, 2\}, \forall t \in]0, \pi/2]$,

$$|\ln(\sin(t))^k \sin(t)^x| \leq |\ln(\sin(t))|^k \sin(t)^a = \phi_k(t)$$

ϕ_k est continue sur $]0, \pi/2]$. Au voisinage de 0, $\phi_k(t) = o(\sin(t)^{\frac{a-1}{2}})$. Or, $\sin(t)^{\frac{a-1}{2}} \sim t^{\frac{a-1}{2}}$ est intégrable au voisinage de 0 (car $\frac{a-1}{2} > -1$). ϕ_k est donc intégrable sur $]0, \pi/2]$.

Le théorème de régularité des intégrales à paramètres indique que $f \in C^2(-1, +\infty[)$ et que

$$\forall x > -1, f'(x) = \int_0^{\pi/2} \ln(\sin(t)) \sin(t)^x dt \text{ et } f''(x) = \int_0^{\pi/2} \ln(\sin(t))^2 \sin(t)^x dt$$

La dérivée première de f est négative sur I (on intègre une fonction négative) et la dérivée seconde est positive. Ainsi

f est décroissante et convexe sur I

Dans le théorème ci-dessus, les points importants sont la classe C^2 vis à vis de x et l'hypothèse de domination pour x dans un segment. Ici, on a dominé les deux dérivées. On aurait pu se contenter de dominer la seconde mais il aurait alors fallu justifier que $t \mapsto \ln(\sin(t)) \sin(t)^x$ est intégrable sur $]0, \pi/2]$.

6. La relation (1) donne

$$\forall x > -1, f(x) = \frac{x+2}{x+1} f(x+2)$$

f étant continue en 1, on a $(x+2)f(x+2) \rightarrow f(1)$ quand $x \rightarrow -1$. De plus, $f(1) = \int_0^{\pi/2} \sin(t) dt = 1 \neq 0$. On en déduit que

$$f(x) \underset{x \rightarrow -1}{\sim} \frac{1}{x+1}$$

7. Pour tout entier $n \in \mathbb{N}$, on a (avec la question 4)

$$(n+2)f(n+2)f(n+1) = (n+1)f(n)f(n+1)$$

ce qui montre que la suite $((n+1)f(n+1)f(n))$ est constante. De plus

$$f(0) = \int_0^{\pi/2} 1 dt = \frac{\pi}{2} \text{ et } f(1) = \int_0^{\pi/2} \sin(t) dt = 1$$

et la valeur de la constante est donc $\frac{\pi}{2}$. Ainsi

$$\forall n \in \mathbb{N}, f(n)f(n+1) = \frac{\pi}{2(n+1)}$$

Ceci est un résultat très classique sur les intégrales de Wallis.

On remarque tout d'abord, avec la décroissance de f et la relation (1), que

$$\forall n \in \mathbb{N}, \frac{n+1}{n+2} f(n) = f(n+2) \leq f(n+1) \leq f(n)$$

On en déduit que $f(n) \sim f(n+1)$ (le quotient tend vers 1) et avec la première relation de cette question que $f(n+1)^2 \sim \frac{\pi}{2(n+1)}$. On peut élever un équivalent à une puissance constante et ainsi

$$f(n) \underset{n \rightarrow +\infty}{\sim} \sqrt{\frac{\pi}{2n}}$$

Il faut maintenant passer de l'équivalent discret (valable pour la variable entière) à un équivalent continu (valable pour la variable réelle). Pour cela, on encadre x entre deux entiers et on utilise la décroissance de f :

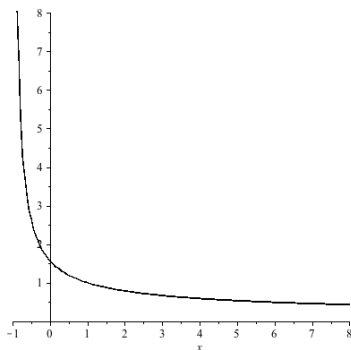
$$f(\lceil x \rceil) \leq f(x) \leq f(\lfloor x \rfloor)$$

Comme $\lceil x \rceil$ et $\lfloor x \rfloor$ équivalent tous deux à x quand $x \rightarrow +\infty$, majorant et minorant équivalent à $\sqrt{\frac{\pi}{2x}}$ avec l'équivalent discret. L'encadrement donne alors

$$\boxed{f(x) \underset{x \rightarrow +\infty}{\sim} \sqrt{\frac{\pi}{2x}}}$$

La technique utilisée ici consistant à établir un résultat dans le cas entier puis à généraliser dans le cas réel à l'aide d'un encadrement est assez courante.

8. Les éléments à reporter sur le dessin sont : la valeur en 0, la valeur en 1, l'asymptote verticale au point d'abscisse -1 , l'asymptote horizontale $y = 0$ (quand $x \rightarrow +\infty$). On pourrait tracer au voisinage de $+\infty$ le graphe de $x \mapsto \sqrt{\frac{\pi}{2x}}$ et au voisinage de 0 celui de $x \mapsto \frac{1}{x+1}$. Sur un dessin à main levée, il est cependant délicat de reporter ces graphes. L'allure de la courbe est la suivante.



Développement en série entière

9. Soit $n \in \mathbb{N}$. $f_n : t \mapsto \ln(\sin(t))^n$ est continue sur $]0, \pi/2]$. Comme, au voisinage de 0, $\ln(\sin(t)) = \ln(t + o(t)) = \ln(t) + \ln(1 + o(1)) \sim \ln(t)$, on a $f_n(t) \sim \ln(t)^n$ quand $t \rightarrow 0$. Par croissances comparées, cet équivalent est $o(1/\sqrt{t})$ et donc intégrable par comparaison aux fonctions de Riemann. Ainsi

$$\boxed{D_n \text{ existe pour tout entier } n}$$

Le changement de variable $x = \pi/2 - t$ donne

$$D_1 = \int_0^{\pi/2} \ln(\sin(\pi/2 - x)) dx$$

Comme $\sin(\pi/2 - x) = \cos(x)$, on conclut que

$$D_1 = \int_0^{\pi/2} \ln(\cos(x)) \, dx$$

10. On utilise l'expression de f' obtenue plus haut :

$$f'(0) = \int_0^{\pi/2} \ln(\sin(t)) \, dt = D_1$$

La question précédente donne (par propriété de morphisme du logarithme)

$$2D_1 = \int_0^{\pi/2} \ln(\cos(t) \sin(t)) \, dt = \int_0^{\pi/2} \ln(\sin(2t)) \, dt - \frac{\pi}{2} \ln(2)$$

Le changement de variable $x = 2t$ m'indique que

$$2 \int_0^{\pi/2} \ln(\sin(2t)) \, dt = \int_0^{\pi} \ln(\sin(x)) \, dx = D_1 + \int_{\pi/2}^{\pi} \ln(\sin(x)) \, dx$$

et en posant $u = \pi - x$, $\int_{\pi/2}^{\pi} \ln(\sin(x)) \, dx = \int_0^{\pi/2} \ln(\sin(\pi - u)) \, du = D_1$. On en déduit enfin que $2D_1 = D_1 - \frac{\pi}{2} \ln(2)$ et

$$f'(0) = D_1 = -\frac{\pi}{2} \ln(2)$$

Ce calcul repose assez essentiellement sur une astuce calculatoire. Il n'est pas simple à deviner si on ne l'a jamais rencontré. Il ne faut pas y passer trop de temps. Il ne faut a priori pas craindre que le résultat soit utile dans la suite car dans ce cas, le résultat aurait sans doute été donné.

On a de même,

$$f'(1) = \int_0^{\pi/2} \ln(\sin(t)) \sin(t) \, dt$$

On effectue le changement de variable $u = \cos(t)$ ($\cos$ est de classe C^1 sur $]0, \pi/2[$, à dérivée ne s'annulant pas et le changement est licite) en remarquant que $\sin(t) = \sqrt{1 - \cos(t)^2}$ car $\sin(t) \geq 0$ pour $t \in]0, \pi/2[$:

$$f'(1) = \int_0^1 \ln(\sqrt{1 - u^2}) \, du = \frac{1}{2} \int_0^1 (\ln(1 - u) + \ln(1 + u)) \, du$$

En posant $v = 1 - u$ dans la première intégrale et $v = 1 + u$ dans la seconde, on se ramène à un calcul d'intégrale de $\ln$ et une primitive est $x \mapsto x \ln(x) - x$. On obtient ainsi

$$f'(1) = \ln(2) - 1$$

Il peut être tentant de commencer le calcul par une intégration par parties en primitivant $\sin(t)$ et en dérivant $\ln(\sin(t))$. Si on effectue le calcul sans précaution, on sera amené à écrire

$$f'(1) = [-\cos(t) \ln(\sin(t))]_0^{\pi/2} + \int_0^{\pi/2} \frac{\cos(t)^2}{\sin(t)} \, dt$$

Cela n'est cependant pas permis puisque le terme entre les crochets est de limite infinie en 0 (voir plus haut les soucis avec les intégrations par parties dans les intégrales généralisées).

On peut contourner le problème en choisissant une autre primitive pour $\sin$: en utilisant $1 - \cos$. On obtient alors un terme dans le crochet de limite nulle et

$$f'(1) = - \int_0^{\pi/2} \frac{(1 - \cos(t)) \cos(t)}{\sin(t)} dt$$

On a alors une fraction rationnelle en $\cos$ et $\sin$. Il existe des règles (qui ne sont pas à connaître) permettant de primitiver de telles fonctions. Ici, le changement de variable $u = \cos(t)$ nous ramène à une fraction rationnelle en u . Il est à noter que le changement de variable $u = \tan(t/2)$ permettrait aussi de conclure (il marche pour toutes les fractions rationnelles en $\cos$ et $\sin$) mais par des calculs moins élémentaires.

11. $t \mapsto -\ln(\sin(t))$ est de classe C^1 sur $]0, \pi/2[$ de dérivée $t \mapsto -\frac{\cos(t)}{\sin(t)}$ qui ne s'annule pas sur $]0, \pi/2[$ (notre fonction réalise donc une bijection de $]0, \pi/2[$ dans son image). On peut donc poser $u = -\ln(\sin(t))$. On a alors $t = \arcsin(e^{-u})$ et donc $dt = -\frac{e^{-u}}{\sqrt{1-e^{-2u}}} du = -\frac{1}{\sqrt{e^{2u}-1}}$ et donc

$$D_n = \int_0^\infty \frac{(-u)^n}{\sqrt{e^{2u}-1}} du$$

On a donc

$$(-1)^n D_n = \int_0^\infty \frac{u^n}{\sqrt{e^{2u}-1}} du$$

Pour les changement de variables "simples" dans les intégrales, on n'a pas à justifier la validité du changement. Ici, la fonction de changement de variable est suffisamment compliquée pour qu'on effectue la justification.

Posons $I_n = \int_0^\infty u^n e^{-u} du$. $u \mapsto u^n e^{-u}$ est continue sur $\mathbb{R}^+$ et $o(1/u^2)$ au voisinage de $+\infty$ et I_n est bien définie. Dans I_{n+1} , on effectue une intégration par parties en primitivant e^{-u} en $-e^{-u}$. Comme $u^{n+1}(-e^{-u})$ est de limite nulle en $+\infty$, l'intégration par parties est licite. Elle donne $I_{n+1} = (n+1)I_n$. Comme $I_0 = 1$, on en déduit que $I_n = n!$.

L'exponentielle est rapidement de limite infinie en l'infini. Le terme $\sqrt{e^{2u}-1}$ est donc vite proche de e^u (dès que u n'est pas trop petit). Le pari fait ici est que D_n va donc se comporter comme I_n . En réalité, la question à se poser est : d'où vient le $n!$ du résultat à prouver ? Comme $I_n = \Gamma(n+1) = n!$ est un résultat classique, on pense à la méthode proposée ci-dessous.

Par ailleurs

$$(-1)^n D_n - I_n = \int_0^{+\infty} u^n \left(\frac{1}{\sqrt{e^{2u}-1}} - e^{-u} \right) du$$

On écrit que

$$\left| \frac{1}{\sqrt{e^{2u}-1}} - e^{-u} \right| = \left| \frac{(1 - \sqrt{1 - e^{-2u}})}{\sqrt{e^{2u}-1}} \right| = \left| \frac{e^{-2u}}{\sqrt{e^{2u}-1}(1 + \sqrt{1 - e^{-2u}})} \right| \leq \frac{e^{-2u}}{\sqrt{e^{2u}-1}}$$

et ainsi

$$|(-1)^n D_n - I_n| \leq \int_0^\infty \frac{e^{-2u} u^n}{\sqrt{e^{2u}-1}} du$$

Pour conclure, il s'agit de montrer que ce terme est très petit devant $n!$ puisque $u_n \sim v_n$ équivaut à $u_n - v_n = o(v_n)$.

L'application $u \mapsto \frac{u}{\sqrt{e^{2u}-1}}$ est continue sur $\mathbb{R}^{+*}$, est de limite nulle en $+\infty$ et tend vers $1/2$ en 0 . Elle est ainsi bornée au voisinage de 0 et de $+\infty$. Comme elle est aussi bornée sur tout segment de $\mathbb{R}^{+*}$ (fonction continue sur un compact), elle est bornée sur $\mathbb{R}^{+*}$. Notons M un majorant de son module. On a alors (en posant $v = 2u$)

$$|(-1)^n D_n - I_n| \leq M \int_0^\infty e^{-2u} u^{n-1} du = \frac{M}{2^n} \int_0^\infty e^{-v} v^{n-1} dv = \frac{M}{2^n} (n-1)! = o(n!)$$

Ceci montre que $(-1)^n D_n - I_n = o(I_n)$ et donc que $(-1)^n D_n \sim I_n$ ou encore que

$$D_n \sim (-1)^n n!$$

12. $\sin(t)^x = e^{x \ln(\sin(t))}$ et avec le DSE de l'exponentielle

$$\forall x > -1, f(x) = \int_0^{\pi/2} \sum_{n=0}^{\infty} \frac{\ln(\sin(t))^n}{n!} x^n dt$$

Pour $x \in]-1, 1[$, on veut intervertir la somme et l'intégrale. On utilise pour cela le théorème d'intégration terme à terme.

- $f_n : t \mapsto \frac{\ln(\sin(t))^n}{n!} x^n$ est continue sur $]0, \pi/2[$ et $\sum (f_n)$ converge simplement sur $]0, \pi/2[$ vers $t \mapsto \sin(t)^x$ elle-même continue sur $]0, \pi/2[$.
- Pour tout entier $n \in \mathbb{N}$ (et puisque $\ln(\sin(t))$ est négatif sur $]0, \pi/2[$)

$$\int_0^\infty |f_n| = \frac{(-1)^n |x|^n}{n!} D_n \underset{n \rightarrow +\infty}{\sim} |x|^n$$

est le terme général d'une série convergente (puisque $|x| < 1$).

On peut utiliser le théorème pour conclure que

$$\forall x \in]-1, 1[, f(x) = \sum_{n=0}^{\infty} \frac{D_n}{n!} x^n$$

ce qui montre que f est DSE sur $] -1, 1[$.

Pour intervertir une somme infinie et une intégrale, le programme nous fournit trois outils. Le premier ne s'utilise que quand on intègre sur un segment et nécessite une convergence uniforme sur ce segment. Le second est le théorème de convergence dominée et nécessite de majorer le module des sommes partielles par une quantité indépendante de l'indice n et intégrable. Le troisième est le théorème d'intégration terme à terme utilisé ici. C'est le plus naturel au sens où l'énoncé nous a fait évaluer l'intégrale. Attention, dans ce théorème (et avec les notations précédentes), c'est $\int_0^{+\infty} |f_n|$ qui nous intéresse (regardez bien la place des valeurs absolues).

Convergence de suite de fonctions

13. Comme $a, b > 0$, pour tout x on a $a^2 \cos(x)^2 + b^2 \sin(x)^2 > 0$ (somme de deux termes positifs qui ne sont pas simultanément nuls). Ψ est donc bien définie sur $\mathbb{R}$ et par théorèmes d'opération $\Psi \in C^1(\mathbb{R})$. De plus (après simplification)

$$\Psi'(x) = \frac{(b^2 - a^2) \sin(2x)}{a^2 \cos(x)^2 + b^2 \sin(x)^2}$$

Par ailleurs, $|\rho| < 1$ et ainsi (somme géométrique de raison ρe^{2ix} de module < 1)

$$\sum_{k=1}^{\infty} (\rho e^{2ix})^k = \frac{\rho e^{2ix}}{1 - \rho e^{2ix}} = \frac{\rho e^{2ix}(1 - \rho e^{-2ix})}{1 + \rho^2 - 2\rho \cos(2x)}$$

et en passant à la partie imaginaire

$$\sum_{k=1}^{\infty} \rho^k \sin(2kx) = \frac{\rho \sin(2x)}{1 + \rho^2 - 2\rho \cos(2x)}$$

En remplaçant ρ par sa valeur et comme $\cos(2x) = \cos^2(x) - \sin^2(x)$,

$$\begin{aligned} 1 + \rho^2 - 2\rho \cos(2x) &= \frac{2(a^2 + b^2)}{(b+a)^2} - \frac{2(b-a)}{b+a}(\cos^2(x) - \sin^2(x)) \\ &= \frac{2}{(b+a)^2} (a^2 + b^2 - (b^2 - a^2)(\cos(x)^2 - \sin(x)^2)) \\ &= \frac{2}{(b+a)^2} (2a^2 \cos(x)^2 + 2b^2 \sin(x)^2) \\ &= \frac{4}{(b+a)^2} (a^2 \cos(x)^2 + b^2 \sin(x)^2) \end{aligned}$$

et ainsi

$$\sum_{k=1}^{\infty} \rho^k \sin(2kx) = \frac{b^2 - a^2}{4} \frac{\sin(2x)}{a^2 \cos(x)^2 + b^2 \sin(x)^2}$$

En comparant à l'expression de $\Psi'(x)$, on obtient

$$\boxed{\Psi'(x) = 4 \sum_{k=1}^{\infty} \rho^k \sin(2kx)}$$

On aurait pu partir de la somme qui est la partie imaginaire de la somme, géométrique, des e^{2ikx} . On sait donc calculer cette somme et il convient de la simplifier pour faire apparaître l'expression de ψ' .

14. Par théorème fondamental (avec Ψ de classe C^1)

$$\Psi(x) - \Psi(0) = \int_0^x \Psi'(t) dt = 4 \int_0^x \sum_{k=1}^{\infty} \rho^k \sin(2kt) dt$$

Fixons un réel x et posons $g_n : t \mapsto \rho^k \sin(2kt)$. On a $\|g_n\|_{\infty, [0, x]} \leq |\rho|^k$ qui est le terme général d'une série convergente (car $|\rho| < 1$). Ainsi $\sum (g_n)$ converge normalement sur le segment $[0, x]$ et on peut intervertir les symboles :

$$\Psi(x) - \Psi(0) = 4 \sum_{k=1}^{\infty} \rho^k \int_0^x \sin(2kt) dt = 4 \sum_{k=1}^{\infty} \frac{\rho^k}{2k} (1 - \cos(2kx))$$

Comme $|\rho| < 1$, on peut scinder la somme en deux (les deux morceaux existent) et on obtient l'existence d'une constante c telle que

$$\forall x \in \mathbb{R}, \quad \Psi(x) = c - 2 \sum_{k=1}^{\infty} \frac{\rho^k}{k} \cos(2kx)$$

On obtient la valeur de c en évaluant en $x = 0$. On reconnaît un développement connu :

$$c = 2 \ln(a) + 2 \sum_{k=1}^{\infty} \frac{\rho^k}{k} = 2 \ln(a) - 2 \ln(1 - \rho)$$

En remplaçant ρ par sa valeur et en simplifiant (par propriété de morphisme du logarithme) on obtient

$$\Psi(x) = 2 \ln\left(\frac{a+b}{2}\right) - 2 \sum_{k=1}^{\infty} \frac{\cos(2kx)}{k} \rho^k$$

15. On a tout d'abord

$$\Psi(x)^2 = 2 \ln\left(\frac{a+b}{2}\right) \Psi(x) - 2 \sum_{k=1}^{\infty} \frac{\cos(2kx)}{k} \rho^k \Psi(x)$$

Comme $\|x \mapsto \frac{\cos(2kx)}{k} \rho^k \Psi(x)\|_{\infty, [0, \pi]} \leq |\rho^k| \|\Psi\|_{\infty, [0, \pi]}$ est le terme général d'une série convergente, quand on intègre on peut intervertir somme et intégrale (cas de convergence uniforme sur un segment) et ainsi

$$\int_0^{\pi} \Psi^2 = 2 \ln\left(\frac{a+b}{2}\right) \int_0^{\pi} \Psi - 2 \sum_{k=1}^{\infty} \frac{\rho^k}{k} \int_0^{\pi} \cos(2kx) \Psi(x) dx$$

Le même type d'argument donne aussi (en rappelant que $\int_0^{\pi} \cos(px) dx = 0$ quand $p \in \mathbb{Z}^*$)

$$\int_0^{\pi} \Psi = 2\pi \ln\left(\frac{a+b}{2}\right) - 2 \sum_{k=1}^{\infty} \frac{\rho^k}{k} \int_0^{\pi} \cos(2kx) dx = 2\pi \ln\left(\frac{a+b}{2}\right)$$

$$\begin{aligned} \int_0^{\pi} \cos(2kx) \Psi(x) dx &= 2 \ln\left(\frac{a+b}{2}\right) \int_0^{\pi} \cos(2kx) - 2 \sum_{j=1}^{\infty} \frac{\rho^j}{j} \int_0^{\pi} \cos(2jx) \cos(2kx) dx \\ &= - \sum_{j=1}^{\infty} \frac{\rho^j}{j} \int_0^{\pi} (\cos(2(j+k)x) + \cos(2(j-k)x)) dx \\ &= - \sum_{j=1}^{\infty} \frac{\rho^j}{j} \int_0^{\pi} \cos(2(j-k)x) dx \\ &= -\pi \frac{\rho^k}{k} \end{aligned}$$

En regroupant les calculs :

$$\int_0^{\pi} \Psi^2 = 4\pi \left(\ln\left(\frac{a+b}{2}\right) \right)^2 + 2\pi \sum_{k=1}^{\infty} \frac{\rho^{2k}}{k^2}$$

c'est à dire

$$\int_0^{\pi} \Psi^2 = 4\pi \left(\ln\left(\frac{a+b}{2}\right) \right)^2 + 2\pi \sigma(\rho^2)$$

On aurait sans doute pu commencer par exprimer $\psi^2(x)$ à l'aide d'une somme double mais le cadre du programme, permettant une interversion d'une somme simple et d'une intégrale, nous incite à procéder comme nous l'avons fait.

16. On a

$$\forall t \in]0, \pi], \Psi_n(t) = \ln \left(\frac{\cos(t)^2}{(n+1)^2} + \frac{n^2}{(n+1)^2} \sin(t)^2 \right)$$

Le terme dans le logarithme tend vers $\sin(t)^2 > 0$ quand $t \in]0, \pi[$ (problème d'énoncé) et donc

$$\boxed{(\Psi_n) \text{ converge simplement sur }]0, \pi[\text{ vers } t \mapsto \ln(\sin(t)^2)}$$

On note que

$$\int_0^\pi \Psi_n(t)^2 dt = 4\pi \ln(2)^2 + 2\pi\sigma \left(\frac{(n-1)^2}{(n+1)^2} \right)$$

On fait tendre n vers l'infini et par continuité de σ en 1, on obtient

$$\lim_{n \rightarrow +\infty} \int_0^\pi \Psi_n(t)^2 dt = 4\pi \ln(2)^2 + \frac{\pi^3}{3}$$

Supposons provisoirement que l'on puisse intervertir limite et intégrale. On aurait alors (on écrit que $\ln(\sin(t)^2) = 2 \ln(\sin(t))$)

$$4 \int_0^\pi (\ln(\sin(t)))^2 dt = 4\pi \ln(2)^2 + \frac{\pi^3}{3}$$

Le changement de variable $u = \pi - t$ montre que l'intégrale de $(\ln(\sin(t)))^2$ est la même sur $[0, \pi/2]$ et sur $[\pi/2, \pi]$. L'égalité précédente donne la valeur de cette intégrale et donc

$$\boxed{f''(0) = \frac{\pi \ln(2)^2}{2} + \frac{\pi^3}{24}}$$

Mieux vaut commencer par vérifier que sous réserve de validité, le théorème utilisé permet de conclure. En effet, à quoi bon justifier les hypothèses d'un théorème si celui-ci ne permet pas de conclure ? Ceci est particulièrement vrai à l'oral. Dans un sujet à l'écrit, le type de rédaction proposé est tout à fait acceptable. On peut bien sûr préférer "mettre les choses dans l'ordre" et justifier les hypothèses puis conclure.

Il nous reste à justifier l'interversion. Pour cela, on utilise le théorème de convergence dominée.

- (Ψ_n^2) est une suite de fonctions continues sur $]0, \pi[$ qui converge simplement sur cet ensemble vers $t \mapsto (\ln(\sin(t)))^2$ elle-même continue sur $]0, \pi[$.
- Pour tout $n \in \mathbb{N}^*$ et tout $t \in]0, \pi]$, on a

$$a_n^2 \cos(t)^2 + b_n^2 \sin(t)^2 = a_n^2 + (b_n^2 - a_n^2) \sin(t)^2 = \frac{1}{(n+1)^2} + \frac{n-1}{n+1} \sin(t)^2$$

$\frac{n-1}{n+1}$ croît avec n et pour $n \geq 2$, sa valeur est dans $[1/3, 1]$. Par ailleurs, $0 \leq \frac{1}{(n+1)^2} \leq$

1. Ainsi

$$\forall n \geq 2, \frac{\sin(t)^2}{3} \leq a_n^2 \cos(t)^2 + b_n^2 \sin(t)^2 \leq 1 + \sin(t)^2$$

Par croissance de $\ln$,

$$\forall n \geq 2, \ln \left(\frac{\sin(t)^2}{3} \right) \leq \Psi_n(t) \leq \ln(1 + \sin(t)^2)$$

et donc (si $a \leq y \leq b$ alors $y^2 \leq a^2 + b^2$)

$$\forall n \geq 2, 0 \leq \Psi_n(t)^2 \leq \left(\ln \left(\frac{\sin(t)^2}{3} \right) \right)^2 + (\ln(1 + \sin(t)^2))^2$$

Le majorant $\theta(t)$ est continu sur $]0, \pi[$. Au voisinage de 0, il équivaut à $4 \ln(|\sin(t)|)$ ou encore à $4 \ln(|t|)$ et y est intégrable (en 0^+ et en 0^-). La π -périodicité de θ (qui est bien définie sur $\mathbb{R} \setminus \pi\mathbb{Z}$) donne l'intégrabilité au voisinage de π . Finalement, θ est intégrable sur $]0, \pi[$.

Le théorème de convergence dominée s'applique et justifie l'interversion admise plus haut.

La domination n'est pas du tout simple à obtenir. En effet, majorer $|\psi_n(t)|$, c'est majorer une expression qui s'écrit $|\ln(u_n(t))|$. Une erreur classique (et grossière) consiste à majorer $u_n(t)$ pour en déduire une majoration de $\ln(u_n(t))$ et passer ensuite aux valeurs absolues. C'est bien sûr faux car l'application $x \mapsto |x|$ n'est pas croissante. Nous utilisons donc un encadrement et le fait que si $\alpha \leq y \leq \beta$, alors $|y| \leq \max(|\alpha|, |\beta|) \leq |\alpha| + |\beta|$.

Convexité logarithmique

17. f est bien définie sur I et y est > 0 (on intègre une fonction positive, continue et non nulle). $\ln \circ f$ est donc bien définie sur I . De plus

$$(\ln \circ f)'' = \frac{f''f' - (f')^2}{f^2}$$

Fixons $x \in I$. La question 5 donne

$$\begin{aligned} f'(x) &= \int_0^{\pi/2} \ln(\sin(t)) \sin(t)^x dx = \int_0^{\pi/2} u(t)v(t) dt \text{ avec} \\ u(t) &= \sin(t)^{x/2}, v(t) = \ln(\sin(t)) \sin(t)^{x/2} \end{aligned}$$

u et v sont des fonctions de carré intégrable sur $]0, \pi/2[$. Par inégalité de Cauchy-Schwarz dans l'espace des fonctions continues sur $]0, \pi/2[$ et de carré intégrable muni du produit scalaire $(u, v) \mapsto \int_0^{\pi/2} uv$, on obtient

$$|f'|^2 \leq \left(\int_0^{\pi/2} u^2 \right) \left(\int_0^{\pi/2} v^2 \right) = f(x)f''(x)$$

On en déduit que $ff'' - (f')^2$ est positive sur I et que $\ln \circ f$ est de dérivée seconde positive et donc convexe. Ainsi

f est ln-convexe sur I

En utilisant une technique similaire, on prouve la ln-convexité sur $\mathbb{R}^{+*}$ de la célèbre fonction Γ d'Euler. Notez aussi qu'on peut utiliser une borne flottante $\varepsilon > 0$ et utiliser l'inégalité de Cauchy-Schwarz dans $C^0([\varepsilon, \pi/2])$ pour majorer $\left(\int_\varepsilon^{\pi/2} \ln(\sin(t)) \sin(t)^x dx \right)^2$ puis faire tendre ε vers 0.

18. La relation (1) donne (quand $2x > -1$)

$$\tilde{f}(x+1) = \ln(f(2x+2)) = \ln\left(\frac{2x+1}{2x+2}f(2x)\right) = \tilde{f}(x) + \ln\left(\frac{2x+1}{2x+2}\right)$$

On écrit cette relation pour $x+k$ et on somme pour k allant de 0 à $p+1$. Par télescopage

$$\tilde{f}(x+p) - \tilde{f}(x) = \sum_{k=0}^{p-1} (\tilde{f}(x+k+1) - \tilde{f}(x+k)) = \sum_{k=0}^{p-1} \ln\left(\frac{2x+2k+1}{2x+2k+2}\right)$$

On en conclut que

$$\tilde{f}(x+p) = \tilde{f}(x) + \sum_{k=0}^{p-1} \ln\left(\frac{2x+2k+1}{2x+2k+2}\right)$$

Cette identité est vraie pour $p \in \mathbb{N}^*$ (et même $p = 0$ en convenant qu'une somme vide est nulle) et pour $x > -1/2$.

On peut bien sûr préférer procéder par récurrence.

19. $\tilde{f}$ est convexe sur $] -1/2, +\infty[$ par $\ln$ -convexité de f sur I ($(\tilde{f})'' = 4(\ln \circ f)'' \geq 0$). Si $a < b < c$ alors (inégalité des pentes)

$$\frac{\tilde{f}(b) - \tilde{f}(a)}{b-a} \leq \frac{\tilde{f}(c) - \tilde{f}(a)}{c-a} \leq \frac{\tilde{f}(c) - \tilde{f}(b)}{c-b}$$

On utilise ceci avec $n-1 < n < n+x$ et avec $n < n+x < n+p$ (si $x = p$, la seconde inégalité obtenue restera valable) pour conclure que (quand $n, p \in \mathbb{N}^*$ et $0 < x \leq p$)

$$\tilde{f}(n) - \tilde{f}(n-1) \leq \frac{\tilde{f}(n+x) - \tilde{f}(n)}{x} \leq \frac{\tilde{f}(n+p) - \tilde{f}(n)}{p}$$

Il faut connaître toutes les caractérisations de la convexité données en première année.

La question 18 indique que

$$\forall n \in \mathbb{N}^*, \tilde{f}(n) = \tilde{f}(0) + \sum_{k=0}^{n-1} \ln\left(\frac{2k+1}{2k+2}\right)$$

et on en déduit que

$$\tilde{f}(n+1) - \tilde{f}(n) = \ln\left(\frac{2n+1}{2n+2}\right) \xrightarrow{n \rightarrow +\infty} 0$$

puis pour tout entier $p \geq 1$,

$$\tilde{f}(n+p) - \tilde{f}(n) = \sum_{k=n}^{n+p-1} (\tilde{f}(k+1) - \tilde{f}(k)) \xrightarrow{n \rightarrow +\infty} 0$$

puisque l'on a p suites de limite nulle.

Pour $x > 0$ fixé, on choisit $p = \lceil x \rceil$. Dans la double inégalité obtenue plus haut, majorant et minorant sont de limite nulle quand $n \rightarrow +\infty$. Il en est de même du terme du milieu. Et comme x est une constante,

$$\lim_{n \rightarrow +\infty} (\tilde{f}(n+x) - \tilde{f}(n)) = 0$$

20. On remarque tout d'abord que f est bien ln-convexe sur I , vérifie (1) et $f(0) = \frac{\pi}{2}$. C'est donc l'unicité qui nous intéresse.

On se donne une fonction g convenable et on lui associe $\tilde{g}$. Les résultats des questions 18 et 19 s'appliquent à g (puisque l'on a seulement utilisé (1) et la ln-convexité et pas l'expression de f). On a en particulier

$$\forall n \in \mathbb{N}^*, \tilde{g}(n) = \tilde{g}(0) + \sum_{k=0}^{n-1} \ln \left(\frac{2k+1}{2k+2} \right)$$

et comme $\tilde{f}(0) = \tilde{g}(0)$,

$$\forall n \in \mathbb{N}^*, \tilde{g}(n) = \tilde{f}(n)$$

De même, pour $x \geq 0$,

$$\forall n \in \mathbb{N}, \tilde{f}(x+n) - \tilde{f}(x) = \tilde{g}(x+n) - \tilde{g}(x)$$

et en combinant,

$$\forall n \in \mathbb{N}, (\tilde{f}(x+n) - \tilde{f}(n)) - \tilde{f}(x) = (\tilde{g}(x+n) - \tilde{g}(n)) - \tilde{g}(x)$$

Quand $n \rightarrow +\infty$, le membre de gauche tend vers $-\tilde{f}(x)$ (question 19).

Pour le membre de droite, on a besoin de la convexité de $\tilde{g}$ sur $] -1/2, +\infty[$ pour appliquer aussi la question 19. Elle découle de la ln-convexité de g (si $x, y > -1/2$ alors $x, y \in I$ et l'inégalité de convexité pour $\ln \circ g$ entre $2x$ et $2y$ donne celle pour $\tilde{g}$ entre x et y).

En faisant tendre n vers $+\infty$ et avec la question 19, $\tilde{f}(x) = \tilde{g}(x)$ quand $x > 0$. En composant par l'exponentielle, $f(2x) = g(2x)$. Quand x varie dans $\mathbb{R}^{+*}$, il en est de même de $2x$ et ainsi (on a bien l'égalité en 0)

$$\forall x \geq 0, f(x) = g(x)$$

Si $x \in]-1, 0]$, on écrit que $f(x) = \frac{x+2}{x+1} f(x+2) = \frac{x+2}{x+1} g(x+2) = g(x)$. On a ainsi $f = g$.

f est la seule solution du problème

21. Soit g une solution au problème. g est donc à valeurs dans $\mathbb{R}^{+*}$ et $c = g(0) > 0$. Posons $h : x \mapsto \frac{\pi}{2c} g(Tx)$ (définie pour $x > -1$). On a

$$(x+1)h(x) = \frac{\pi}{2c} (x+1)g(Tx) = \frac{\pi}{2cT} (Tx+T)g(Tx) = \frac{\pi}{2cT} (Tx+2T)g(Tx+2T) = (x+2)h(x+2)$$

De plus la ln-convexité de g entraîne celle de g puisque

$$\begin{aligned} \ln(h(\lambda x + (1-\lambda)y)) &= \ln\left(\frac{\pi}{2c}\right) + \ln(g(\lambda Tx + (1-\lambda)Ty)) \\ &\leq \ln\left(\frac{\pi}{2c}\right) + \lambda \ln(g(Tx)) + (1-\lambda) \ln(g(Ty)) \\ &\leq \lambda \ln(h(x)) + (1-\lambda) \ln(h(y)) \end{aligned}$$

Enfin $h(0) = \frac{\pi}{2}$. Avec la question 20, $h = f$ et donc $g : x \mapsto \frac{2c}{\pi} f(\frac{x}{T})$.

Réciproquement, les fonctions $x \mapsto \alpha f(\frac{x}{T})$ avec $\alpha > 0$, sont ln-convexes et vérifient la relation voulue.

Les fonctions convenables sont les $x \mapsto \alpha f(\frac{x}{T})$ avec $\alpha > 0$

22. Supposons que h convienne. La restriction de h à $] -T, +\infty[$ est solution du problème de la question 21 et il existe donc $\alpha > 0$ tel que

$$\forall x > T, h(x) = \alpha f\left(\frac{x}{T}\right)$$

On en déduit avec la question 6 que

$$\lim_{x \rightarrow (-T)^+} h(x) = +\infty$$

De plus $\ln \circ h$ et $\exp$ étant convexes, h l'est aussi (une composée de fonctions convexes est convexe) et h est donc continue sur $\mathbb{R}$ (d'après le lemme des pentes, une fonction convexe sur $\mathbb{R}$ est dérivable à droite et gauche en tout point et donc continue en tout point). On obtient une contradiction. Ainsi

Le problème n'admet pas de solution

On peut, alternativement, remarquer que si h convient alors

$$\forall t \in \mathbb{R}, (t+T)h(t) = (t+2T)h(t+2T)$$

L'évaluation en $t = -T$ donne $h(T) = 0$ ce qui contredit la stricte positivité de h .

Le résultat de continuité sur l'intérieur de J d'une fonction convexe sur l'intervalle J n'est pas au programme. Peut-être faudrait-il développer le raisonnement évoqué ci-dessus. En dernière question, la mention faite de la dérivabilité à droite et gauche me semble suffisante. Il faut cependant faire plus que simplement noter la continuité.



ÉCOLE DES PONTS PARISTECH,
ISAE-SUPAERO, ENSTA PARIS,
TÉLÉCOM PARIS, MINES PARIS,
MINES SAINT-ÉTIENNE, MINES NANCY,
IMT ATLANTIQUE, ENSAE PARIS,
CHIMIE PARISTECH - PSL.

Concours Mines-Télécom,
Concours Centrale-Supélec (Cycle International).

CONCOURS 2023

ÉPREUVE D'INFORMATIQUE MP

Durée de l'épreuve : 3 heures

L'usage de la calculatrice ou de tout dispositif électronique est interdit.

Cette épreuve concerne uniquement les candidats de la filière MP.

*Les candidats sont priés de mentionner de façon apparente
sur la première page de la copie :*

INFORMATIQUE - MP

L'énoncé de cette épreuve comporte 11 pages de texte.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il le signale sur sa copie et poursuit sa composition en expliquant les raisons des initiatives qu'il est amené à prendre.

Les sujets sont la propriété du GIP CCMP. Ils sont publiés sous les termes de la licence
Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
Tout autre usage est soumis à une autorisation préalable du Concours commun Mines Ponts.



Préliminaires

L'épreuve est composée d'un problème unique, comportant 27 questions. Après cette section de préliminaires, qui présente le *jeu du hanjie*, le problème est divisé en trois sections qui peuvent être traitées séparément, à condition d'avoir lu les définitions introduites jusqu'à la question traitée. Dans la première section (page 2), nous résolvons le jeu par des raisonnements logiques. Dans la deuxième section (page 3), nous modélisons le jeu et le résolvons par une stratégie algorithmique de retour sur trace. Dans la troisième section (page 6), nous appliquons une méthode de parcours de graphe en théorie des automates pour accélérer la résolution du jeu.

Dans tout l'énoncé, un même identificateur écrit dans deux polices de caractères différentes désigne la même entité, mais du point de vue mathématique pour la police en italique (par exemple n) et du point de vue informatique pour celle en romain avec espacement fixe (par exemple `n`).

Des rappels de logique et des extraits du manuel de documentation de OCaml sont reproduits en annexe. Ces derniers portent sur le module `Array` et le module `Hashtbl`.

Travail attendu

Pour répondre à une question, il est permis de réutiliser le résultat d'une question antérieure, même sans avoir réussi à établir ce résultat.

Il faudra coder des fonctions à l'aide du langage de programmation OCaml, en reprenant l'en-tête de fonction fourni par le sujet, sans s'obliger à recopier la déclaration des types. Quand l'énoncé demande de coder une fonction, sauf demande explicite de l'énoncé, il n'est pas nécessaire de justifier que celle-ci est correcte ou de vérifier que des préconditions sont satisfaites.

Le barème tient compte de la clarté des programmes : nous recommandons de choisir des noms de variables intelligibles ou encore de structurer de longs codes par des blocs ou par des fonctions auxiliaires dont on décrit le rôle.

Description du jeu du hanjie

Le *hanjie* est un jeu de réflexion à l'intersection de l'art des pixels et de la tomographie discrète, technique d'imagerie courante en médecine, en géophysique ou en science des matériaux entre autres domaines.

Il consiste à retrouver une image par le noircissement de certaines cases d'une grille rectangulaire sur la base d'indications laissées sur les côtés de la grille. Pour chaque rangée, qu'elle soit horizontale ou verticale, le joueur dispose d'une suite d'entiers non nuls t_1, t_2, t_3 , etc. qui indiquent que la rangée contient une série de t_1 cases noires consécutives, suivie plus loin d'une série de t_2 cases noires consécutives, et ainsi de suite. Un nombre quelconque de cases blanches peut se trouver en tête ou en queue de rangée ; au moins une case blanche sépare deux séries de cases noires.

Voici un exemple, que nous résolvons à la main. Une grille vide est fournie ci-contre. Elle est de dimension 5×7 . Nous marquons les cases par des symboles « ? » tant que nous ignorons leur couleur.

Nous comptons les rangées à partir de 0, de gauche à droite pour les colonnes et du haut vers le bas pour les lignes.

La colonne 0 et la colonne 5 sont de longueur 5. Or l'indication est [5] dans les deux cas. Elles doivent donc chacune contenir 5 blocs noirs consécutifs. Nous les noirçissons en totalité.

Dans la ligne 0, il n'y a qu'une seule manière de placer un bloc de 4 cases noires puis 2 cases noires. De même, dans la ligne 2, il n'y a qu'une seule manière de positionner le bloc de longueur 2 : il doit se trouver au milieu entre les deux blocs déjà isolés. Dans la ligne 3, nous avons déjà placé deux cases noires : les autres sont donc toutes blanches. Enfin, dans la ligne 4, il n'y a plus qu'une seule manière de placer un bloc de 6 cases noires.

Enfin, en reprenant les indications des colonnes, nous voyons que dans les colonnes 1, 2 et 4, la dernière case inconnue est blanche. Dans la colonne 3 et 6 en revanche, nous devons noirçir la dernière case inconnue. Nous avons obtenu une solution de notre hanjie. Elle est unique.

		[1,1,1]	[3,1]	[1]	[5]	[2]
[5]	[1,1]	[4,2]	[1,1,2]	[1,2,1]	[1,1]	[6]
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

		[1,1,1]	[3,1]	[1]	[5]	[2]
[5]	[1,1]	[4,2]	[1,1,2]	[1,2,1]	[1,1]	[6]
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

		[1,1,1]	[3,1]	[1]	[5]	[2]
[5]	[1,1]	[4,2]	[1,1,2]	[1,2,1]	[1,1]	[6]
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

		[1,1,1]	[3,1]	[1]	[5]	[2]
[5]	[1,1]	[4,2]	[1,1,2]	[1,2,1]	[1,1]	[6]
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?
?	?	?	?	?	?	?

1. Hanjie et calcul de vérité

Dans cette section, nous raisonnons avec la logique propositionnelle pour déterminer la couleur de certaines cases. Nous nous concentrons sur le hanjie h_0 défini par

		[2]	[1]	[1]
[2]	[1,1]	?	?	?
?	?	?	?	?

et dont une solution est la suivante :

		[2]	[1]	[1]
[2]	[1,1]	?	?	?
?	?	?	?	?

□ 1 – Établir, par raisonnement en langue française, que la solution du hanjie h_0 est unique.

Nous introduisons six variables booléennes, nommées $x_0, x_1, \dots, x_5$ et correspondant aux cases ci-contre. Nous associons la valeur de vérité V (vrai) à la couleur noire et F (faux) à la couleur blanche.

x_0	x_1	x_2
x_3	x_4	x_5

Épreuve d'informatique d'option MP 2023

Soit L_0 le prédicat : « l'indication de la ligne zéro du hanjie h_0 est satisfaite ».

□ 2 – Dresser la table de vérité du prédicat L_0 portant sur les variables x_0, x_1, x_2 . En déduire une formule de logique φ sous forme normale conjonctive qui décrit le prédicat L_0 .

Soit C_1 le prédicat : « l'indication de la colonne du milieu du hanjie h_0 est satisfaite ».

□ 3 – Dresser la table de vérité du prédicat C_1 portant sur les variables x_1, x_4 . En déduire une formule de logique ψ sous forme normale conjonctive qui décrit le prédicat C_1 .

Les règles d'inférence de la déduction naturelle sont rappelées dans l'annexe B.

□ 4 – Construire un arbre de preuve qui démontre le séquent $\varphi \vdash x_1$ à partir des règles d'inférence de la déduction naturelle.

□ 5 – Construire de même un arbre de preuve qui démontre le séquent $\psi, x_1 \vdash \neg x_4$.

Nous notons ψ' la formule de logique obtenue à partir de ψ en remplaçant la variable x_1 par x_2 et la variable x_4 par x_5 .

□ 6 – Démontrer qu'il n'existe pas d'arbre de preuve qui démontre la formule $\varphi \wedge \psi' \rightarrow \neg x_2$.

2. Cadre de résolution systématique du hanjie

2.1. Le hanjie

Nous fixons un alphabet $\mathcal{C} = \{\mathbf{N}, \mathbf{B}\}$ et notons $\bar{\mathcal{C}}$ l'alphabet complété $\bar{\mathcal{C}} = \{\mathbf{N}, \mathbf{B}, \mathbf{I}\}$. Les symboles $\mathbf{N}$, $\mathbf{B}$ et $\mathbf{I}$ désignent respectivement les couleurs **Noir**, **Blanc** et **Inconnu**. Nous déclarons en OCaml :

```
type couleur = N | B | I
```

□ 7 – Écrire une fonction OCaml `est_connu (c:couleur) : bool` dont la valeur de retour est le booléen **V** (vrai) si $c \in \mathcal{C} = \{\mathbf{N}, \mathbf{B}\}$ et le booléen **F** (faux) si c égale la couleur $\mathbf{I}$.

Pour l'ensemble du sujet, nous fixons deux entiers naturels non nuls m et n qui désignent respectivement un nombre de lignes et un nombre de colonnes.

Définition : Une *présolution* d'un hanjie est un tableau de dimension $m \times n$ à valeurs dans l'ensemble $\bar{\mathcal{C}}$.

Les figures de l'introduction (en page 2) sont des exemples de présolutions.

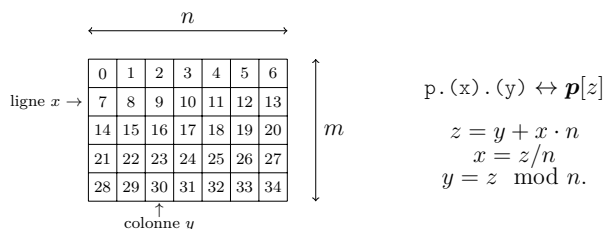
Indication OCaml : Nous déclarons des constantes globales, le type et le constructeur suivants :

```
let m = 5
let n = 7
type presolution = couleur array array

let presolution_init () : presolution = (* cree une presolution *)
    Array.make_matrix m n I             (* toute egale a I *)
```

Lorsqu'un tableau $\mathbf{p}$ est de dimension $m \times n$, nous numérotons ses cases ligne après ligne avec les entiers compris entre 0 et $m \cdot n - 1$. La case en ligne x et colonne y a pour numéro $z = y + x \cdot n$. Nous notons alors $\mathbf{p}[z]$ la valeur de $\mathbf{p}$ en position z .

Dans notre exemple, nous aurions les numéros :



□ 8 – Écrire en langage OCaml un accesseur `get (p:presolution) (z:int) : couleur` dont la valeur de retour est la couleur $\mathbf{p}[z]$.

□ 9 – Écrire en langage OCaml un transformateur `set (p:presolution) (z:int) (c:couleur) : presolution` dont la valeur de retour $\mathbf{p}'$ est une copie profonde de $\mathbf{p}$ avec $\mathbf{p}'[z] = c$, c'est-à-dire une copie n'ayant aucun lien avec la présolution initiale.

Dans tout le sujet, on s'astreint à manipuler le type `presolution` exclusivement en employant les fonctions `presolution_init`, `set` et `get`. De la sorte, on pourra considérer que le type `presolution` est immuable.

□ 10 – Définir le terme *immuable* et citer un avantage à utiliser des variables immuables.

La *ligne* x d'un tableau, avec $0 \leq x < m$, désigne l'ensemble des positions de numéro $x \cdot n$, $x \cdot n + 1$, $\dots$, $(x + 1) \cdot n - 1$ (lire n numéros au total). La *colonne* y , avec $0 \leq y < n$, désigne l'ensemble des positions de numéro y , $y + n$, $\dots$, $y + (m - 1) \cdot n$ (lire m numéros au total). Nous utilisons le terme *rangée* pour parler indifféremment d'une ligne ou d'une colonne.

Définition : Nous disons qu'une rangée d'une présolution est *complète* si elle est à valeurs dans $\mathcal{C}$.

□ 11 – Écrire une fonction OCaml `est_complete_lig (p:presolution) (x:int) : bool` qui teste si la ligne x de la présolution $\mathbf{p}$ est complète.

Nous supposons écrite de même une fonction `est_complete_col (p:presolution) (y:int) : bool`, qui teste si la colonne y de la présolution $\mathbf{p}$ est complète.

Définition : Si une rangée est complète, nous appelons *trace* la suite des longueurs des sous-suites maximales de termes consécutifs égaux à **Noir**. Par exemple, la trace de la rangée

$\text{B, N, N, B, N, B, N, N, N, B, N}$

est $[2; 1; 3; 1]$.

□ 12 – Écrire une fonction OCaml `trace_lig (p:presolution) (x:int) : int list` dont la valeur de retour est la trace de la ligne x de la présolution $\mathbf{p}$.

Nous supposons écrite de la même manière une fonction `trace_col (p:presolution) (y:int) : int list`, dont la valeur de retour est la trace de la colonne y de la présolution $\mathbf{p}$.

Définition : Le terme *indication* désigne toute suite finie d'entiers naturels non nuls. Nous appelons *hanjie de dimension $m \times n$* la donnée d'un tableau d'indications $\mathbf{ind}_{lig}$, de longueur m , et d'un tableau d'indications $\mathbf{ind}_{col}$, de longueur n .

Indication OCaml : Nous déclarons :

```
type hanjie = { ind_lig : int list array;
                ind_col : int list array }
```

Définition : Une *solution* d'un hanjie $h = (\mathbf{ind}_{lig}, \mathbf{ind}_{col})$ de dimension $m \times n$ est un tableau $\mathbf{p}$ de même dimension $m \times n$ et à valeurs dans l'ensemble $\mathcal{C}$ tel que le tableau des traces des lignes de $\mathbf{p}$ égale $\mathbf{ind}_{lig}$ et le tableau des traces des colonnes de $\mathbf{p}$ égale $\mathbf{ind}_{col}$.

Par exemple, le hanjie étudié en introduction et déclaré par :

```
let h_escargot = {ind_lig = [|4;2|; |1;1;2|; |1;2;1|; |1;1|; |6| |];
                  ind_col = [|5|; |1;1|; |1;1;1|; |3;1|; |1| ; |5| ;
                             |2| |] }
```

admet comme solution la valeur OCaml :

```
let p_escargot = [| [|N; N; N; N; B; N; N|];
                    [|N; B; B; N; B; N; N|];
                    [|N; B; N; N; B; N; B|];
                    [|N; B; B; B; B; N; B|];
                    [|N; N; N; N; N; N; B|] |]
```

□ 13 – Écrire une fonction OCaml `est_admissible (h:hanjie) (p:presolution) : bool` dont la valeur de retour est le booléen **V** (vrai) si et seulement si, pour toute rangée complète de $\mathbf{p}$, la trace égale l'indication du hanjie. Il n'est pas nécessaire de contrôler que la présolution et le hanjie ont même dimension.

2.2. Recherche de solutions

Définition : Nous disons qu'une présolution $\mathbf{p}'$ *étend* une présolution $\mathbf{p}$, si pour tout numéro z avec $\mathbf{p}[z] \in \mathcal{C}$, les couleurs $\mathbf{p}'[z]$ et $\mathbf{p}[z]$ sont égales.

Par exemple, la présolution $\mathbf{p}' = \boxed{\text{B N I N B}}$ étend $\mathbf{p} = \boxed{\text{B I I N I}}$ puisque deux cases sont passées de la couleur **I** à une couleur connue **N** ou **B** et les autres sont restées inchangées.

Indication OCaml : Le type paramétré 'a option permet de distinguer l'absence d'une valeur définie, avec le constructeur None, de la présence d'une valeur v de type 'a, avec la construction Some v. Il est défini par la déclaration :

```
type 'a option = None | Some of 'a
```

□ 14 – Écrire une fonction OCaml `etend_trivial` (h :hanjie) (p :presolution) (z :int) (c :couleur) : presolution option qui, si la copie p' de p avec $p'[z] = c$ est une extension de p et est admissible au sens de la question 13, renvoie Some p' et sinon renvoie None.

Plus généralement, nous appelons *extenseur* toute fonction OCaml `etend` (h :hanjie) (p :presolution) (z :int) (c :couleur) : presolution option ainsi spécifiée :

Précondition : Le numéro z est valide ($0 \leq z < m \cdot n$). La couleur c appartient à $\mathcal{C}$.

Postcondition : Si la valeur de retour vaut Some p' , alors

- (i) p' est une extension admissible de p avec $p'[z] = c$
- (ii) et toute présolution complète p'' qui satisfait (i) est une extension de p' .

Sinon, si la valeur de retour est None, alors il n'existe pas de présolution p' complète vérifiant (i).

Nous définissons le type

```
type extenseur = hanjie -> presolution ->
                int -> couleur -> presolution option
```

Nous souhaitons implémenter une recherche de solutions par une stratégie de *retour sur trace* dans laquelle les cases d'une présolution sont considérées par numéro croissant.

□ 15 – Compléter le code suivant afin que, si p est une présolution dont au moins les $z - 1$ premières cases appartiennent à $\mathcal{C}$, alors `explore p z` renvoie si possible une solution qui étend p à partir du numéro z à l'aide d'itérations sur l'extenseur `ext` et sinon renvoie la valeur None :

```
let resout (h:hanjie) (ext:extenseur) : presolution option =
  let p0 = presolution_init () in
  let rec explore (p:presolution) (z:int) : presolution option =
    (* A COMPLETER *)
  in
  explore p0 0
```

3. Résolution autonome de lignes et extenseur

La stratégie d'extension de la question 14 est très naïve. Elle ignore les déductions intermédiaires qui pourraient être faites grâce à une seule indication à partir d'une rangée partiellement complétée.

Soit $w \in \mathcal{C}^n$ le mot formé des inscriptions dans une rangée dont l'indication est $\tau = [t_1; t_2; \dots; t_s]$ avec $s \geq 1$. Nous notons $[w]_\tau$ l'ensemble des mots de $\mathcal{C}^n$ de trace τ qui étendent w . Lorsque l'ensemble $[w]_\tau$ n'est pas vide, nous posons, pour tout indice i compris entre 0 et $n - 1$, l'ensemble des couleurs

$$E_i = \{z_i \in \mathcal{C}; \mathbf{z} = z_0 \cdots z_{n-1} \in [w]_\tau\}.$$

Épreuve d'informatique d'option MP 2023

Enfin nous posons, pour tout indice i compris entre 0 et $n - 1$,

$$x_i = \begin{cases} \mathbf{B} & \text{si } E_i = \{\mathbf{B}\} \\ \mathbf{N} & \text{si } E_i = \{\mathbf{N}\} \\ \mathbf{I} & \text{si } E_i = \{\mathbf{B}, \mathbf{N}\}. \end{cases}$$

Définition : Nous appelons *plus grande extension commune du mot w par rapport à l'indication τ* le mot $\mathbf{x} = x_0 \cdots x_{n-1} \in \overline{\mathcal{C}}^*$, lorsqu'il est défini.

Par exemple, si nous rencontrons la ligne

N	I	I	I	I	I	N	I

 avec l'indication $[1, 2, 1]$, il y a deux extensions complètes possibles :

N	B	N	N	B	B	N	B

 ou

N	B	B	N	N	B	N	B

. La plus grande extension commune est dans ce cas

N	B	I	N	I	B	N	B

.

□ 16 – Dans cette question uniquement, nous supposons que n est de la forme $n = 3s - 1$, où s est un entier naturel non nul, que l'indication τ est égale à $[1; 1; \dots; 1]$ avec s occurrences de 1 et que w vaut $\mathbf{I}^n$. Compter le nombre de mots du langage $[w]_\tau$.

Nous notons L_τ le langage des mots de $\mathcal{C}^*$ de trace $\tau = [t_1; t_2; \dots; t_s]$ quelconque. Nous supposons toujours que l'on a $s \geq 1$.

□ 17 – Proposer une expression régulière e qui dénote le langage L_τ . Aucune justification n'est attendue.

□ 18 – Dessiner un automate fini déterministe incomplet à $s + \sum_{i=1}^s t_i$ états qui reconnaît le langage L_τ . Aucune justification n'est attendue.

□ 19 – Dire combien d'états l'automate de Glushkov qui dérive de l'expression régulière e de la question 17 possède et si cet automate coïncide avec celui dessiné à la question 18.

Dans ce qui suit, nous nous intéressons à des *automates finis déterministes incomplets* qui ne possèdent qu'un seul état final et dont l'alphabet est toujours $\mathcal{C}$. Nous numérotions systématiquement les états de sorte que l'état initial soit 0 et l'unique état final soit $r - 1$, où r est le nombre total d'états. Le terme *transition* désigne un triplet (q, c, q') où q et q' sont des états (avec $0 \leq q, q' < r$) et c est une couleur (avec $c \in \mathcal{C}$). Nous représentons la fonction de transition par un tableau de longueur r ; la case q du tableau contient un dictionnaire qui, quand il existe une transition (q, c, q') , associe la couleur c à l'état q' . Nous déclarons :

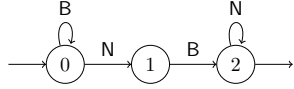
```
type automate = { r : int;
                  transitions : (couleur, int) Hashtbl.t array}
```

Définition : Le *déploiement* d'un automate $\mathcal{A}$ ayant r états par un mot $\mathbf{w} = w_0 w_1 \cdots w_{n-1} \in \overline{\mathcal{C}}^*$ de longueur n est un nouvel automate, noté $\mathcal{A} \bowtie \mathbf{w}$, qui possède $r \cdot (n + 1)$ états. Pour toute transition (q, c, q') dans l'automate $\mathcal{A}$ et pour tout indice i compris entre 0 et $n - 1$,

Épreuve d'informatique d'option MP 2023

pour $w_i \in \mathcal{C}$ et $c = w_i$ ou encore pour $w_i = \mathbf{l}$, il y a dans l'automate $\mathcal{A} \bowtie \mathbf{w}$ une transition $(i \cdot r + q, c, (i + 1) \cdot r + q')$.

□ 20 – Dans cette question uniquement, on considère l'exemple avec $n = 4$, $\mathbf{w}_0 = \text{INBI} \in \overline{\mathcal{C}}^*$ et où $\mathcal{A}_0$ est l'automate défini par



Dessiner le déploiement $\mathcal{A}_0 \bowtie \mathbf{w}_0$. Marquer les états accessibles. Marquer les états co-accessibles, c'est-à-dire les états à partir desquels il existe un chemin vers l'état final.

□ 21 – Soient $\mathcal{A}$ un automate et $\mathbf{w} \in \mathcal{C}^*$ un mot. Écrire une fonction OCaml accessible (a:automate) (w:couleur array) : bool array dont la valeur de retour $\mathbf{b}$ est un tableau de booléens tels que $\mathbf{b}[q]$ est vrai si et seulement si l'état q de $\mathcal{A} \bowtie \mathbf{w}$ est accessible.

□ 22 – Calculer la complexité en temps de la fonction accessible définie à la question 21.

□ 23 – Soient $\mathcal{A}$ un automate à r états et $\mathbf{w} \in \mathcal{C}^*$ un mot de longueur n . Écrire une fonction coaccessible (a:automate) (w:couleur array) : bool array dont la valeur de retour $\mathbf{b}$ est un tableau de booléens tels que $\mathbf{b}[q]$ est vrai si et seulement si l'état q de $\mathcal{A} \bowtie \mathbf{w}$ est co-accessible.

Définition : Soient $\mathcal{A}$ un automate à r états et $\mathbf{w} \in \overline{\mathcal{C}}^*$ un mot de longueur n . Nous supposons qu'il existe un mot de longueur n reconnu par l'automate $\mathcal{A} \bowtie \mathbf{w}$. Nous rappelons que l'automate émondé de $\mathcal{A} \bowtie \mathbf{w}$ est la copie de l'automate duquel ont été retirées toutes les transitions depuis ou vers des états qui ne sont pas à la fois accessibles et co-accessibles (nous ne chassons pas les états inutiles et conservons la numérotation des états). Nous notons $\hat{\Delta}$ l'ensemble des transitions restantes dans l'automate émondé. Nous notons, pour tout indice i compris entre 0 et $n - 1$, l'ensemble des étiquettes

$$H_i = \left\{ c \in \mathcal{C}; (q, c, q') \in \hat{\Delta} \cap \llbracket i \cdot r, (i + 1) \cdot r - 1 \rrbracket \times \mathcal{C} \times \llbracket (i + 1) \cdot r, (i + 2) \cdot r - 1 \rrbracket \right\}.$$

Nous posons, pour tout entier i compris entre 0 et $n - 1$,

$$y_i = \begin{cases} \mathbf{B} & \text{si } H_i = \{\mathbf{B}\} \\ \mathbf{N} & \text{si } H_i = \{\mathbf{N}\} \\ \mathbf{l} & \text{si } H_i = \{\mathbf{B}, \mathbf{N}\}. \end{cases}$$

Nous appelons *mot projeté du déploiement* $\mathcal{A} \bowtie \mathbf{w}$ le mot $\mathbf{y} = y_0 y_1 \dots y_{n-1} \in \overline{\mathcal{C}}^*$.

□ 24 – Avec les notations du paragraphe qui précède et lorsque l'automate $\mathcal{A}$ est l'automate dessiné à la question 18, vérifier que le mot projeté $\mathbf{y} \in \overline{\mathcal{C}}^*$ est la plus grande extension commune du mot $\mathbf{w} \in \overline{\mathcal{C}}^*$ par rapport à l'indication τ (cf. définition p. 7).

Épreuve d'informatique d'option MP 2023

□ 25 – Écrire une fonction `projete (a:automate) (w:couleur array) : couleur array` dont la valeur de retour est le mot projeté du déploiement $\mathcal{A} \bowtie w$.

Nous rappelons la *formule de Stirling* :

$$n! \sim \sqrt{2\pi n} \left(\frac{n}{e}\right)^n.$$

□ 26 – Comparer la complexité en temps de la fonction `projete` (question 25) avec un calcul direct de la plus grande extension commune par énumération de l'ensemble $[w]_{\tau}$. Argumenter en s'appuyant sur la question 16.

Nous nous donnons une fonction OCaml `pgec_lig (h:hanjie) (p:presolution) (x:int) : presolution option`, et respectivement `pgec_col (h:hanjie) (p:presolution) (y:int) : presolution option`, qui étend la ligne x , respectivement la colonne y , par la plus grande extension commune à l'image de la question 25 ou bien renvoie `None` si la plus grande extension commune n'est pas définie.

□ 27 – Écrire un extenseur `etend_nontrivial (h:hanjie) (p:presolution) (z:int) (c:couleur) : presolution option` qui modifie la couleur de $\mathbf{p}$ au numéro z et applique les fonctions `pgec_lig` et `pgec_col` aussi longtemps que cela permet de faire progresser la présolution. Il est demandé de détailler la stratégie de l'extenseur avant d'en fournir le code.

A. Annexe : aide à la programmation en OCaml

Opérations sur les tableaux : Le module `Array` offre les fonctions suivantes :

- `length : 'a array -> int`
Return the length (number of elements) of the given array.
- `make : int -> 'a -> 'a array`
`Array.make n x` returns a fresh array of length n , initialized with x . All the elements of this new array are initially physically equal to x (in the sense of the `==` predicate). Consequently, if x is mutable, it is shared among all elements of the array, and modifying x through one of the array entries will modify all other entries at the same time.
- `make_matrix : int -> int -> 'a -> 'a array array`
`Array.make_matrix dimx dimy e` returns a two-dimensional array (an array of arrays) with first dimension `dimx` and second dimension `dimy`. All the elements of this new matrix are initially physically equal to e . The element (x, y) of a matrix m is accessed with the notation `m.(x).(y)`.
- `init : int -> (int -> 'a) -> 'a array`
`Array.init n f` returns a fresh array of length n , with element number i initialized to the result of $f(i)$. In other terms, `init n f` tabulates the results of f applied to the integers 0 to $n - 1$.
- `copy : 'a array -> 'a array`
`Array.copy a` returns a copy of a , that is, a fresh array containing the same elements as a .
- `mem : 'a -> 'a array -> bool`
`mem a l` is true if and only if a is structurally equal to an element of l (i.e. there is an x in l such that `compare a x = 0`).
- `for_all : ('a -> bool) -> 'a array -> bool`
`Array.for_all f [|a1; ...; an|]` checks if all elements of the array satisfy the predicate f . That is, it returns `(f a1) && (f a2) && ... && (f an)`.
- `exists : ('a -> bool) -> 'a array -> bool`
`Array.exists f [|a1; ...; an|]` checks if at least one element of the array satisfies the predicate f . That is, it returns `(f a1) || (f a2) || ... || (f an)`.
- `map : ('a -> 'b) -> 'a array -> 'b array`
`Array.map f a` applies function f to all the elements of a , and builds an array with the results returned by f : `[| f a.(0); f a.(1); ...; f a.(length a - 1) |]`.
- `iter : ('a -> unit) -> 'a array -> unit`
`Array.iter f a` applies function f in turn to all the elements of a . It is equivalent to `f a.(0); f a.(1); ...; f a.(length a - 1); ()`.

D'après <https://v2.ocaml.org/api/Array.html>

Opérations sur les tables de hachage : Le module `Hashtbl` offre les fonctions suivantes :

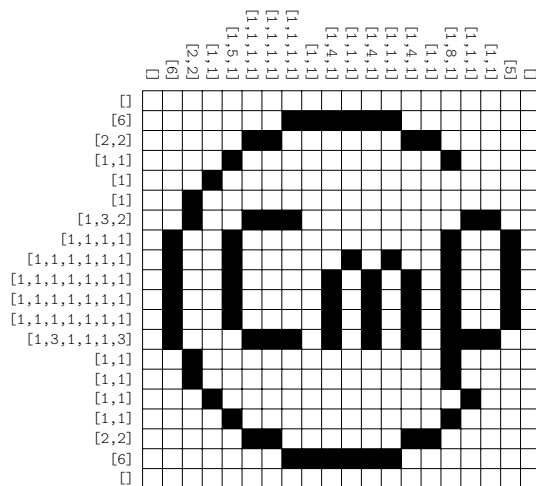
- `('a, 'b) Hashtbl.t`
The type of hash tables from type $'a$ to type $'b$.
- `create : int -> ('a, 'b) t`
`Hashtbl.create n` creates a new, empty hash table, with initial size n . For best results, n should be on the order of the expected number of elements that will be in the table. The table grows as needed, so n is just an initial guess.
- `add : ('a, 'b) t -> 'a -> 'b -> unit`
`Hashtbl.add tbl key data` adds a binding of key to data in table `tbl`.
- `remove : ('a, 'b) t -> 'a -> unit`
`Hashtbl.remove tbl x` removes the current binding of x in `tbl`, restoring the previous binding if it exists. It does nothing if x is not bound in `tbl`.
- `mem : ('a, 'b) t -> 'a -> bool`
`Hashtbl.mem tbl x` checks if x is bound in `tbl`.
- `iter : ('a -> 'b -> unit) -> ('a, 'b) t -> unit`
`Hashtbl.iter f tbl` applies f to all bindings in table `tbl`. f receives the key as first argument, and the associated value as second argument. Each binding is presented exactly once to f .
- `find_opt : ('a, 'b) t -> 'a -> 'b option`
`Hashtbl.find_opt tbl x` returns the current binding of x in `tbl`, or `None` if no such binding exists.

D'après <https://v2.ocaml.org/api/Hashtbl.html>

B. Annexe : règles de la déduction naturelle

Dans les tableaux suivants, la lettre Δ désigne un ensemble de formules de logique ; les lettres A , B et C désignent des formules de logique.

Axiome		
$\frac{}{\Delta, A \vdash A} \text{ (ax)}$		
	Introduction	Élimination
$\rightarrow$	$\frac{\Delta, A \vdash B}{\Delta \vdash A \rightarrow B} (\rightarrow i)$	$\frac{\Delta \vdash A \quad \Delta \vdash A \rightarrow B}{\Delta \vdash B} (\rightarrow e)$
$\wedge$	$\frac{\Delta \vdash A \quad \Delta \vdash B}{\Delta \vdash A \wedge B} (\wedge i)$	$\frac{\Delta \vdash A \wedge B}{\Delta \vdash A} (\wedge e)$ $\frac{\Delta \vdash A \wedge B}{\Delta \vdash B} (\wedge e)$
$\vee$	$\frac{\Delta \vdash A}{\Delta \vdash A \vee B} (\vee i)$ $\frac{\Delta \vdash B}{\Delta \vdash A \vee B} (\vee i)$	$\frac{\Delta \vdash A \vee B \quad \Delta, A \vdash C \quad \Delta, B \vdash C}{\Delta \vdash C} (\vee e)$
$\neg$	$\frac{\Delta, A \vdash B \quad \Delta, A \vdash \neg B}{\Delta \vdash \neg A} (\neg i)$	$\frac{\Delta \vdash A \quad \Delta \vdash \neg A}{\Delta \vdash B} (\neg e)$



FIN DE L'ÉPREUVE

SOLUTION

Q1. La première colonne est constituée de 2 cases noires donc elle est entièrement définie par

N		
N		

La deuxième ligne contient deux blocs de 1 cases noires. Comme tout bloc de cases noires est suivi d'un bloc de cases blanches (s'il existe) donc nécessairement la deuxième ligne est constituée d'une case noire suivie d'une case blanche puis d'une case noire.

Ainsi, la deuxième ligne est entièrement déterminée par

N		
N	B	N

La deuxième et la troisième colonne contiennent chacune exactement 1 case noire donc le hanjje nécessairement

N	N	B
N	B	N

donc la solution est unique.

Commentaires : Question de difficulté standard. Il est important d'avoir une explication claire (et si possible simple).

Q2. Voici sa table de vérité (il n'est pas demandé d'exprimer L_0 en fonction des x_i).

x_0	V	F	V	V	F	F	V	F
x_1	V	V	F	V	F	V	F	F
x_2	V	V	V	F	V	F	F	F
L_0	F	V	F	V	F	F	F	F

Ainsi, L_0 est vraie si et seulement si (x_0 est faux et x_1 est vraie et x_2 est vraie) ou (x_0 est vraie et x_1 est vraie et x_2 est faux) donc

$$\begin{aligned} L_0 &\equiv (\neg x_0 \wedge x_1 \wedge x_2) \vee (x_0 \wedge x_1 \wedge \neg x_2) \\ &\equiv x_1 \wedge ([\neg x_0 \wedge x_2] \vee [x_0 \wedge \neg x_2]) \\ &\equiv x_1 \wedge ([\neg x_0 \wedge x_2] \vee [x_0 \wedge \neg x_2]) \end{aligned}$$

Par distributivité du connecteur $\vee$ sur $\wedge$, on obtient

$$L_0 \equiv x_1 \wedge ([\neg x_0 \vee x_0] \wedge [x_2 \vee x_0] \wedge [\neg x_0 \vee \neg x_2] \wedge [x_2 \vee \neg x_2])$$

Comme $x \vee \neg x \equiv V$ et que $V \wedge y \equiv y$, on obtient

$$L_0 \equiv x_1 \wedge [x_2 \vee x_0] \wedge [\neg x_0 \vee \neg x_2]$$

donc

$$\varphi = x_1 \wedge [x_2 \vee x_0] \wedge [\neg x_0 \vee \neg x_2]$$

est une forme normale conjonctive qui décrit L_0 .

Commentaires : Il n'est pas demandé d'exprimer L_0 à l'aide des variables propositionnelles. Si la table de vérité ne pose aucune difficulté, la notion de forme normale conjonctive a posé problème pour une fraction importante des candidats (confusion avec la formule normale disjonctive résultant directement de la table alors qu'il fallait utiliser celle de sa négation $\neg L_0$ puis passer à la négation pour avoir la forme normale conjonctive). A partir de la forme normale disjonctive, on peut obtenir la formule conjonctive en utilisant la distributivité de $\wedge$ sur $\vee$ (et réciproquement !)

Q3. Voici sa table de vérité.

x_1	V	F	V	F
x_4	V	V	F	F
C_1	F	V	V	F

Ainsi, C_1 est vraie si et seulement si (x_1 est faux et x_4 est vraie) ou (x_1 est vraie et x_4 est faux) donc

$$\begin{aligned} C_1 &\equiv (\neg x_1 \wedge x_4) \vee (x_1 \wedge \neg x_4) \\ &\equiv [\neg x_1 \vee x_1] \wedge [x_1 \vee x_4] \wedge [\neg x_1 \vee \neg x_4] \wedge [x_4 \vee \neg x_4] \\ &\equiv [x_1 \vee x_4] \wedge [\neg x_1 \vee \neg x_4]. \end{aligned}$$

Ainsi

$$\psi = [x_1 \vee x_4] \wedge [\neg x_1 \vee \neg x_4]$$

est une forme normale conjonctive qui traduit C_1 .

Commentaires : *Même remarque que précédemment.*

Q4. Posons $\varphi' = \neg x_0 \wedge x_2$ et $\varphi'' = x_0 \wedge \neg x_2$ alors $\varphi = x_1 \wedge \varphi' \wedge \varphi''$. On peut alors écrire :

$$\frac{\frac{\varphi \vdash x_1 \wedge \varphi' \wedge \varphi''}{\varphi \vdash x_1 \wedge \varphi'} \quad (\wedge e)}{\varphi \vdash x_1} \quad (\wedge e)$$

Commentaires : *Question d'application du cours. Néanmoins, la déduction naturelle étant nouvelle au programme cette année là, la question s'est avérée sélective.*

Q5. Posons $\psi' = x_1 \vee x_4$ et $\psi'' = \neg x_1 \vee \neg x_4$ alors $\psi = \psi' \wedge \psi''$. On peut alors écrire :

$$\frac{\frac{\psi, x_1 \vdash \psi' \wedge \psi''}{\psi, x_1 \vdash \psi''} \quad (\wedge e) \quad \frac{\psi, x_1, \neg x_1 \vdash \neg x_1}{\psi, x_1, \neg x_1 \vdash \neg x_4} \quad (\wedge e) \quad \frac{\psi, x_1, \neg x_1 \vdash \neg x_4}{\psi, x_1 \vdash \neg x_4} \quad (\vee e)}{\psi, x_1 \vdash \neg x_4} \quad (\wedge e)$$

Commentaires : *Question demandant une certaine aisance dans les règles de déduction naturelle. Elle s'avère difficile au final.*

Q6. Par définition, on a

$$\psi' = [\neg x_2 \vee \neg x_5] \wedge [\neg x_2 \vee \neg x_5]$$

donc

$$\varphi \wedge \psi' = x_1 \wedge (x_2 \vee x_0) \wedge (\neg x_0 \vee \neg x_2) \wedge (x_2 \vee x_5) \wedge (\neg x_2 \vee \neg x_5).$$

D'après le cours, si la formule $f = \varphi \wedge \psi' \rightarrow \neg x_2$ admet un arbre de preuve, elle est une tautologie. Prouvons alors que ce n'est pas une tautologie comme cela cette formule n'aura aucun arbre de preuve. Si on affecte à x_1 la valeur V, x_2 la valeur V, à x_0 la valeur F, x_5 la valeur F alors $\varphi \wedge \psi'$ prend la valeur V tandis que $\neg x_2$ prend la valeur F. Par conséquent, f prend la valeur F d'où f n'est pas une tautologie.

Commentaires : *Question pour les meilleures copies.*

Q7.

```
let est_connu c = match c with
| I -> false
| c -> true;;
```

Commentaires : *Question élémentaire.*

Q8.

```
let get (p:presolution) (z:int) : couleur =
  let x = z/n and y = z mod n in p.(x).(y);;
```

Commentaires : *Question élémentaire.*

Q9.

```
let set (p:presolution) (z:int) (c:couleur) : presolution =
  let p' = presolution_init() in
  for i=0 to m-1 do
    for j=0 to n-1 do
      p'.(i).(j) <- p.(i).(j)
    done;done;
  let x = z/n and y = z mod n in p'.(x).(y) <- c;
  p';;
```

Commentaires : *Question de difficulté standard. La copie profonde ne peut se limiter à écrire $p'.(i) <- p.(i)$.*

Q10. Une quantité est immuable si elle ne changera jamais de valeur, ce qui assure un véritable contrôle de cette variable (notamment de sécurité, etc).

Commentaires : *Cette notion est au programme (on dit aussi persistant) mais les candidats ont généralement du mal avec.*

Q11.

```
let est_complete_lig (p:presolution) (x:int) : bool =
  let z = ref (x*n) and fin = (x+1)*n in
  while !z < fin && est_connu (get p !z) do
    z := !z + 1 ; done;
  !z == fin;;

let est_complete_col (p:presolution) (y:int) : bool =
  let z = ref y and fin = y+n*m in
  while !z < fin && est_connu (get p !z) do
    z := !z + n; done;
  !z == fin;;
```

Commentaires : *Question de difficulté standard.*

Q12.

```
let trace_lig (p:presolution) (x:int) : int list =
  let res = ref [] and z = ref (x*n) and fin = (x+1)*n and c =
    ref 0 in
  while !z < fin do

    if get p !z == N then c := !c+1
    else
      if !c > 0 then begin res := !c::!res ; c:=0 ; end else c:=0
      ;
      z:=!z+1;
    done;
    if !c>0 then res :=!c::!res;
    List.rev !res;;

let trace_col (p:presolution) (y:int) : int list =
  let res = ref [] and z = ref y and fin = y+n*m and c = ref 0
  in
  while !z < fin do

    if get p !z == N then c := !c+1
    else
      if !c > 0 then begin res := !c::!res ; c:=0 ; end else c:=0
      ;
      z:=!z+n;
    done;
    if !c>0 then res :=!c::!res;
    List.rev !res;;
```

Commentaires : *Cette question demande une aisance du candidat en programmation. Il est nécessaire de bien découper son code, éventuellement avec des fonctions effectuant une tâche un peu complexe. Elle s'avère donc sélective (mais accessible).*

Q13.

```
let rec egal l1 l2 = match l1,l2 with
  | [],[] -> true
  | x::q,y::r -> (x==y)&&(egal q r)
  | _ , _-> false;;

let est_admissible (h:hanjie) (p:presolution) =
  let l = h.ind_lig and c = h.ind_col in
  let x = ref 0 and continue = ref true in
  while !continue &&(!x < m) do
    begin if est_complete_lig p !x then continue := egal (
      trace_lig p !x) l.(!x) end;
```

```

    x:= !x+1 done;
  let y = ref 0 in
  while !continue && (!y < n) do
    begin if est_complete_col p !y then continue := egal (
      trace_col p !y) c.(!y) end;
    y:= !y+1 done;
  !continue;;

```

Commentaires : *Même remarque que précédemment.*

Q14.

```

let etend_trivial (h:hanjie) (p:presolution) (z:int)
  (c:couleur) : presolution option =
  let c' = get p z in
  if (c = I || c' = c) then let p' = set p z c in
    (if est_admissible h p' then Some p' else None)
  else None;;

```

Commentaires : *Question de niveau standard.*

Q15.

```

let resout (h:hanjie) (ext:extenseur) : presolution option =
  let p0 = presolution_init () in
  let rec explore (p:presolution) (z:int) : presolution option
    =
    if z > m*n then Some p else
      let pB = ext h p z B and pN = ext h p z N in
      match pB,pN with
      |None,None -> None
      |Some p'B , None -> explore p'B (z+1)
      |None, Some p'N -> explore p'N (z+1)
      |Some p'B, Some p'N -> let p''B = explore p'B (z+1) in
        if p''B == None then
          explore p'N (z+1) else None
  in explore p0 0 ;;

```

Commentaires : *Question difficile qui demande au candidat d'avoir synthésé correctement les codes informatiques précédentes.*

Q16. Comme tout N est forcément suivi d'un B sauf éventuellement le dernier, il faut placer $s - 1$ blocs NB puis un 1 seul N ensuite. Pour déterminer chaque bloc, il suffit de définir la position de son début. Remarquons que si nous fusionnons les $s - 1$ premiers blocs NB en un bloc N' et le dernier le nommer N' , cela revient à placer s éléments N' parmi les $n - (s - 1) = 2s$ cellules disponibles (on réduit au vide les cellules suivant immédiatement les $s - 1$ premiers N). Pour cela, il suffit de sélectionner s positions (où placer les N') parmi

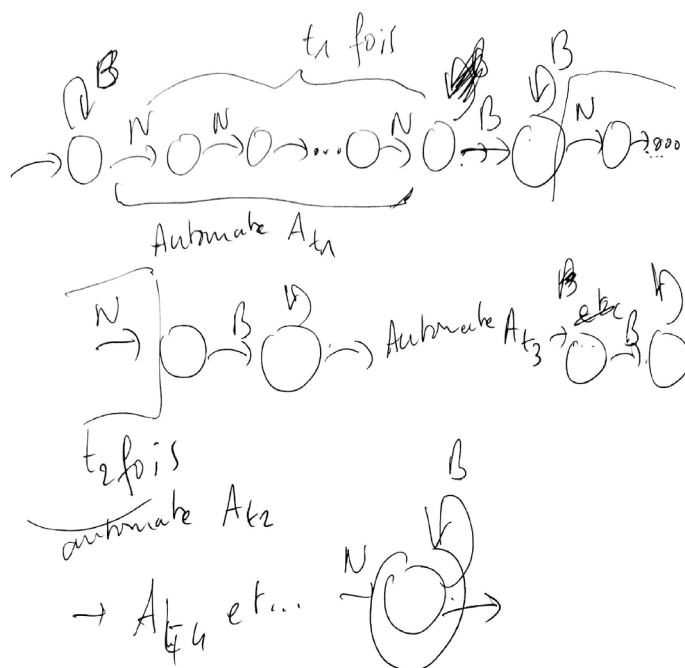
les $2s$ positions disponibles donc il y a $\binom{2s}{s}$ possibilités d'où le nombre de mots de $[w]_T$ est $\binom{2s}{s}$.

Commentaires : Question difficile car il s'agit d'un dénombrement assez complexe, y compris en mathématiques. Bravo aux candidats ayant fourni une réponse convenable.

Q17. Il s'agit de l'expression rationnelle $e = B^*N^{t_1}BB^*N^{t_2}\dots BB^*N^{t_s}B^*$.

Commentaires : Question sans difficulté.

Q18.



Commentaires : Question d'application directe de cours.

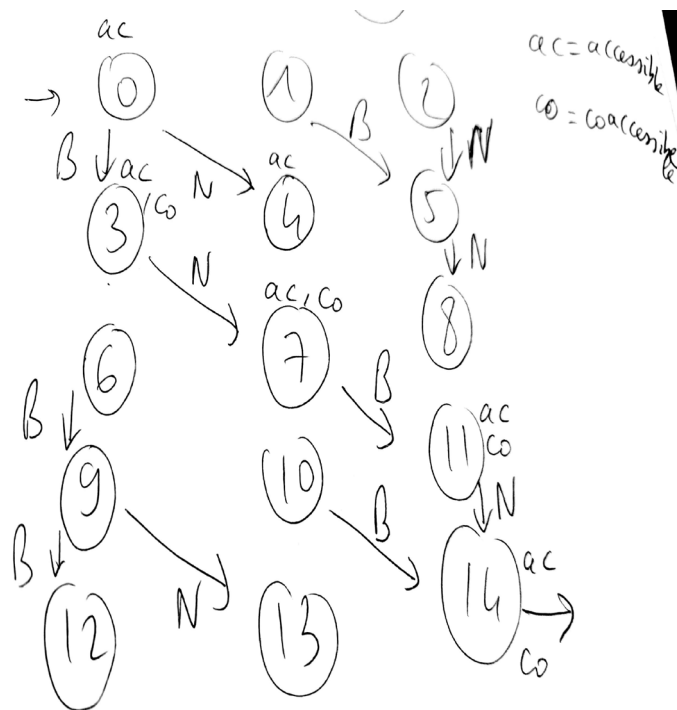
Q19. D'après le cours, l'automate de Glushkov de e possède autant d'états que les lettres que e plus 1 (pour l'état initial ε). Comme cette dernière en compte s lettres B (pour les B^*), $s-1$ lettres B (ceux qui sont avant les B^* entre les N^{t_i}) et $t_1 + \dots + t_s$ lettres N , ($N^3 = NNN$ par définition pour 3 N apparaissent, N^{t_i} comporte t_i lettres N), son nombre d'états est

$$1 + s + (s-1) + t_1 + \dots + t_s = 2s + t_1 + \dots + t_s.$$

L'automate de la question 18 a $s + t_1 + \dots + t_s$ noeuds donc il ne peut être l'automate de Glushkov (car $2s > s!$).

Commentaires : Cette question est potentiellement standard car il s'agit d'une question de cours. Malheureusement, les candidats ne maîtrisant guère les automates de Glushkov, elle s'avère en pratique très sélective et destinée aux meilleures copies.

Q20.



Commentaires : Question élémentaire

Q21.

```

let accessible (a:automate) (w:couleur array) : bool array =
  let r = a.r and t=a.transitions in
  let vu = Array.make ((n+1)*r) false in
  let rec parcourt q i =
    if not vu.(i*r+q) then
      vu.(i*r+q) <- true ;
    if i < n then
      match w.(i), Hashtbl.find_opt t.(q) N with
      | wi, Some q' when (wi==N || wi==I) -> parcourt q' (i+1)
      | _, _ -> () ;
      match w.(i), Hashtbl.find_opt t.(q) B with
      | wi, Some q' when (wi==B || wi==I) -> parcourt q' (i+1)
      | _, _ -> ()
    in parcourt 0 0;
  vu;;

```


Commentaires : Question à priori de niveau standard pour ce concours mais elle rendu difficile car l'automate n'a pas été implémenté donc il faut gérer "à la main" les transitions et, pour cela, il faut manipuler des dictionnaires dont on sait que les candidats n'y sont pas encore acclimatés à ce niveau. Conclusion : la question est difficile et réussie par les candidat(e)s ayant une maîtrise certaine de la programmation en Ocaml.

Q22. Par construction du code, chaque noeud n'est parcouru qu'une seule fois (grâce au tableau *vu*) donc *parcours* n'est appelé qu'aucun plus $r(n+1)$ fois (nombres d'états de l'automate. En dehors de ces appels, il n'y a qu'un nombre fini d'opérations de coût constant (coût en $O(1)$) donc la complexité totale est au plus en

$$r(n+1)O(1) = O(r(n+1)) = O(nr).$$

Commentaires : Question sans difficulté particulière ... pour les candidats ayant réussi la question précédente. Elle est donc peu réussie...

Q23.

```
(*on fixe un état q' au niveau i, présumé strictement positif
et on récupère les couples (q,c) de sorte que (q,c,q') soit
une transition dans l'automate*)

let coaccres t q' r=
  let res = ref [] in
  for q=0 to r-1 do
    match Hashtbl.find_opt t.(q) N with
    |Some q1 when q1==q' -> res :=(q,N)::(!res)
    |_->();
    match Hashtbl.find_opt t.(q) B with
    |Some q1 when q1==q' -> res :=(q,B)::(!res)
    |_->();
    match Hashtbl.find_opt t.(q) I with
    |Some q1 when q1==q' -> res :=(q,I)::(!res)
    |_->()
  done;
  !res;;

let action f w q c q' i =
  match w.(i),c with
  |wi,N when (wi==N || wi==I) -> f q (i-1)
  |wi,B when (wi==B || wi==I) -> f q (i-1)
  |_,_ -> ();;

let rec action_liste f w q' l i = match l with
|[]->()
|(q,c)::reste -> action f w q c q' i ;
    action_liste f w q' reste i;;

let coaccessible (a:automate) (w:couleur array) : bool array =
```

```

let r = a.r and t=a.transitions in
let vu = Array.make ((n+1)*r) false in
let rec parcourt q' i =
  if not vu.((i+1)*r+q') then
    vu.((i+1)*r+q') <- true ;
  if i > 0 then
    let antecedent = coaces t q' r in
    action_liste parcourt w q' antecedent i
in parcourt (n*r+r-1) (n-1);
vu;;

```

Commentaires : Question difficile destinée aux meilleures copies.

Q24. Il suffit de prouver que $H_i = E_i$ pour chaque i . Commençons par vérifier que

$$\mathcal{L}(\mathcal{A} \bowtie w) = [w]_\tau.$$

Soit $x = x_0 \cdots x_{n-1}$ un mot accepté par $\mathcal{A} \bowtie w$ donc il existe des états $q_0, \dots, q_n$ de cet automate avec q_0 un état initial et q_n un état final avec

$$q_0 \xrightarrow{x_0} q_1 \xrightarrow{x_1} q_2 \rightarrow \cdots \xrightarrow{x_{n-1}} q_n.$$

Par définition, chaque transition $q_i \xrightarrow{x_i} q_{i+1}$ est associé à la transition $q_i \bmod(r) \xrightarrow{x_i} q_{i+1} \bmod(r)$ dans A donc $x_0 \cdots x_{n-1} = m$ est reconnu par A . En outre, par définition des transition dans $\mathcal{A} \bowtie w$, on a $x_i = w_i$ ou $w_i = I$. Comme x est accepté par A , il respecte τ donc $x \in [w]_\tau$. Réciproquement, supposons que $x \in [w]_\tau$. Comme x respecte τ , il est accepté par l'automate A en utilisant le chemin

$$q_0 \xrightarrow{x_0} q_1 \xrightarrow{x_1} q_2 \rightarrow \cdots \xrightarrow{x_{n-1}} q_n.$$

Comme $q_i \xrightarrow{x_i} q_{i+1}$ est une transition dans A et que $x \in [w]_\tau$, on peut affirmer que $x_i = w_i$ ou $x_i = I$ donc

$$q'_i = ir + q_i \xrightarrow{x_i} q'_{i+1} = (i+1)r + q_{i+1}$$

est une transition dans $\mathcal{A} \bowtie w$. On a ainsi déterminer un chemin

$$q'_0 \xrightarrow{x_0} q'_1 \xrightarrow{x_1} q'_2 \rightarrow \cdots \xrightarrow{x_{n-1}} q'_n$$

dans $\mathcal{A} \bowtie w$ qui reconnaît x d'où l'égalité $\mathcal{L}(\mathcal{A} \bowtie w) = [w]_\tau$.

Soit $c \in E_i$ alors il existe $z = z_0 \cdots z_{n-1} \in [w]_\tau$ avec $c = z_i$. Comme $z \in [w]_\tau$, z respecte τ donc z est accepté par A d'où il existe un transition $q \xrightarrow{z_i=c} q'$ dans A . Par conséquent, $c = w_i$ ou $w_i = I$ donc

$$ir + q_i \xrightarrow{c} (i+1)r + q_{i+1}$$

est une transition dans $\mathcal{A} \bowtie w$ donc elle appartient à $\hat{\Delta}$ car $z \in [w]_\tau$ est accepté par $\mathcal{A} \bowtie w$. Nous avons prouvé l'inclusion $E_i \subset H_i$.

Si $c \in H_i$, il existe deux états q et q' de $\mathcal{A} \bowtie w$ tel que $q \xrightarrow{c} q'$ appartient à $\hat{\Delta}$ avec

$$q \in [[ir; (r+1)i-1]], c \in \mathcal{C} \text{ et } q' \in [(i+1)r, (i+2)r-1].$$

Puisque q et q' sont accessibles et coaccessibles, il existe $z \in \mathcal{L}(\mathcal{A} \bowtie w) = [w]_\tau$ tel que le chemin acceptant z passe par la transition $q \xrightarrow{c} q'$ donc $c \in E_i$, ce qui prouve l'inclusion réciproque $H_i \subset E_i$ d'où l'égalité $E_i = H_i$, ce qui permet de conclure à la question.

Commentaires : Question difficile destinée aux meilleures copies. Bravo aux quelques valeureux qui ont réussi.

Q25.

```

(*on détermine si l'état k est utile et si deb <= k <fin*)

let etat_utile_zone k deb fin acces coaccs =
  (deb <= k) && (k < fin) && acces.(k) && coaccs.(k);;

(*on filtre toutes les transitions (q,c,q') avec q variant
entre
i*r et (i+1)*r-1 de sorte que q' soit entre (i+1)r et (i+2)r-1
on utilise des booléens pour savoir si on rencontre la couleur
blanche ou noire ou les deux*)

let choix_couleur i r t acces coaccs =
  let b = ref false and n = ref false
  and debq = i*r and finq = (i+1)*r
  and debq' = (i+1)*r and finq' = (i+2)*r
  in let q = ref debq in
  while !q < finq && (not (!b)&&(!n)) do
    (if etat_utile_zone (!q) debq finq acces coaccs then
      match Hashtbl.find_opt t.(!q) N,Hashtbl.find_opt t.(!q)
        B with
      | None,None -> ()
      | None,Some q2' -> if etat_utile_zone (q2') debq' finq'
        acces coaccs
        then b:=true;
      | Some q1',None -> if etat_utile_zone (q1') debq' finq'
        acces coaccs
        then n:=true;
      | Some q1' , Some q2' -> if etat_utile_zone (q1') debq'
        finq' acces coaccs
        then n:=true; if etat_utile_zone (q2') debq' finq'
        acces coaccs
        then b:=true);
    q:=!q+1;
  done;
  match !b,!n with
  | true,false -> B
  | false,true -> N
  | _,_ -> I;;

let projete (a:automate) (w:couleur array) : couleur array =
  let r = a.r and t=a.transitions and acces = accessible a w
  and coaccs = coaccessible a w
  and mot_projete = Array.make n I in
  for i=0 to n-1 do
    mot_projete.(i) <- choix_couleur i r t acces coaccs
  done;
  mot_projete;;

```

Commentaires : *Même remarque que précédemment.*

Q26. La fonction *projete* a une complexité en $O(rn)$ (raisonnement semblable à la question 22)..

Considérons un calcul direct sur l'exemple $\tau = [1; \dots; 1]$ de la question 16 de la plus grande extension nous amène à traiter tous les mots de $[w]_\tau$. Or, le nombre de tels mots est

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

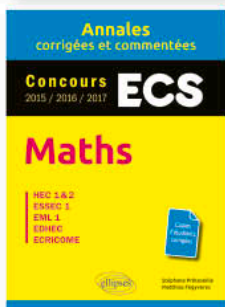
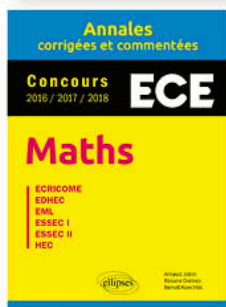
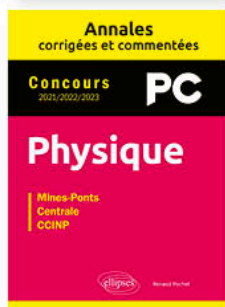
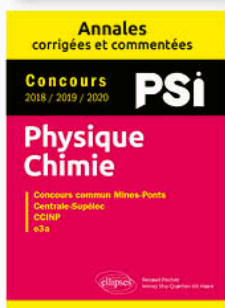
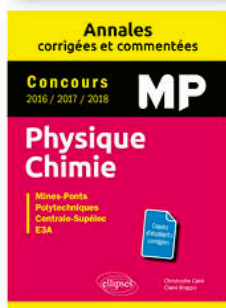
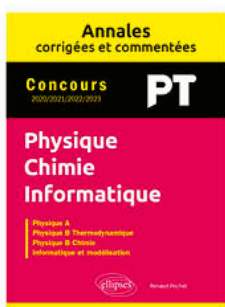
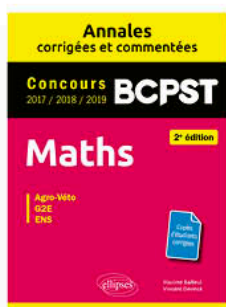
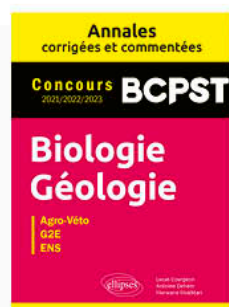
donc sa complexité est au moins exponentielle ($\frac{2^{2s}}{\sqrt{s}2^s} = \frac{2^s}{\sqrt{s}} \xrightarrow{s \rightarrow +\infty} +\infty$) donc la recherche directe a une complexité au moins exponentielle alors que *projete* sur cet exemple a une complexité en $O(s^2)$!

Commentaires : *Question difficile destinée aux meilleures copies.*

Q27. Question inaccessible aux candidats ...

Les épreuves corrigées et commentées des grandes écoles

- **Biologie-Géologie BCPST** - Annales corrigées et commentées. Concours 2021-2022-2023 - L. Courgeon, A. Dehem, M. Mokhtari 9782340080492
- **Informatique MP, PC, PSI, PT** - Annales corrigées et commentées. Concours 2015-2016-2017 - S. Jauvert, L. Jospin 9782340019980
- **Maths BCPST** - Annales corrigées et commentées 2017-2018-2019 - 2^e édition - M. Bailleul, V. Devinck 9782340023888
- **Physique-Chimie-Informatique PT** - Annales corrigées et commentées. Concours 2021-2022-2023 - R. Pochet 9782340079625
- **Maths appliquées ECG** - Annales corrigées et commentées. Concours 2019/2020/2021/2022 - R. Duroux, A. Jobin, B. Koehlin 9782340071056
- **Chimie PC** - Annales corrigées et commentées. Concours 2018/2019/2020 - J. Lejeune 9782340040434
- **Physique-Chimie MP** - Annales corrigées et commentées. Concours 2016-2017-2018 - C. Caire, C. Boggio 9782340025837
- **Physique-Chimie PSI** - Annales corrigées et commentées. Concours 2018/2019/2020 - R. Pochet, W. Shu-Quartier-dit-Maire 9782340040519
- **Physique-Chimie BCPST** - Annales corrigées et commentées. Concours 2018/2019/2020/2021 - C. Bérengé, S. Dakhloul, J. Dupont, B. Godde, C. Reynaud 9782340056626
- **Physique PC** - Annales corrigées et commentées. Concours 2021-2022-2023 - R. Pochet 9782340079618
- **Maths ECE** - Annales corrigées et commentées. Concours 2016-2017-2018 - A. Jobin, R. Duroux, B. Koehlin 9782340026636
- **Maths ECS** - Annales corrigées et commentées. Concours 2015-2016-2017 - S. Prêteselles, M. Fegyveres 9782340019508



www.editions-ellipses.fr



9 782340 085589

