Annales des Concours

PSI
Mathématiques · Informatique
2015

Sous la coordination de

Guillaume Batog
Professeur en CPGE
Ancien élève de l'École Normale Supérieure (Cachan)

Julien Dumont
Professeur en CPGE
Ancien élève de l'École Normale Supérieure (Cachan)

Vincent Puyhaubert
Professeur en CPGE
Ancien élève de l'École Normale Supérieure (Cachan)

Par

Guillaume Batog
Professeur en CPGE

Michel Blockelet
ENS Cachan

Céline Chevalier
Enseignant-chercheur à l'université

Julien Dumont
Professeur en CPGE

Jean-Julien Fleck
Professeur en CPGE

Émilie Liboz
Professeur en CPGE

Matthias Moreno
ENS Lyon

Tristan Poullaouec
Professeur en CPGE

Pauline Tan
ENS Cachan

Sommaire

Centrale-Supélec

Mines-Ponts

Formulaires

Sommaire thématique de mathématiques

2015

	Structures algébriques et arithmétique	Polynômes	Algèbre linéaire générale	Réduction des endomorphismes	Produit scalaire et espaces euclidiens	Topologie des espaces vectoriels normés	Suites et séries numériques	Suites et séries de fonctions	Séries entières	Analyse réelle	Intégration	Équations différentielles	Fonctions de plusieurs variables	Dénombrement et probabilités
e3a PSI Maths A				•		•					•			•
e3a PSI Maths B			•		•									
CCP MP Maths 1								•			•			•
CCP MP Maths 2				•	•									
CCP PC Maths			•								•			•
CCP PSI Maths				•					•			•		
Centrale MP Maths 1											•			
Centrale MP Maths 2								•			•			
Centrale PC Maths 1			•	•										
Centrale PC Maths 2								•		•	•			
Centrale PSI Maths 1							•							•
Centrale PSI Maths 2		•	•								•		•	
Mines MP Maths 1					•			•				•		•
Mines MP Maths 2						•								•
Mines PC Maths 1							•							•
Mines PC Maths 2							•							
Mines PSI Maths 1							•							•
Mines PSI Maths 2			•											
X/ENS MP Maths A		•		•										
X MP Maths B								•		•	•		•	
X/ENS PC Maths				•										

e3a

CONCOURS ARTS ET MÉTIERS ParisTech - ESTP - POLYTECH

Épreuve de Mathématiques 1 PSI

Durée 4 h

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, d'une part il le signale au chef de salle, d'autre part il le signale sur sa copie et poursuit sa composition en indiquant les raisons des initiatives qu'il est amené à prendre.

L'usage de calculatrices est interdit.

AVERTISSEMENT

La **présentation**, la lisibilité, l'orthographe, la qualité de la **rédaction, la clarté et la précision** des raisonnements entreront pour une **part importante** dans **l'appréciation des copies**. En particulier, les résultats non justifiés ne seront pas pris en compte. Les candidats sont invités à encadrer les résultats de leurs calculs.

Tournez la page S.V.P

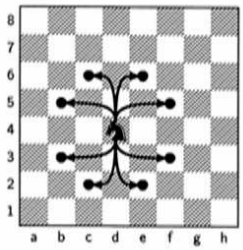
Il est interdit aux candidats de signer leur composition ou d'y mettre un signe quelconque pouvant indiquer sa provenance.

Le sujet est constitué de quatre exercices qui testent plusieurs compétences complémentaires des programmes de mathématiques et d'« informatique pour tous » de la filière PSI.
Il est donc demandé au candidat de répartir équitablement son travail sur les quatre exercices proposés. Il en sera tenu compte dans l'évaluation de l'épreuve.

Exercice 1.

But de l'exercice

Le jeu d'échec se joue sur un échiquier, c'est à dire sur un plateau de 8×8 cases. Ces cases sont référencées de a1 à h8 (cf. figure).
Une pièce, appelée le cavalier, se déplace suivant un "L" imaginaire d'une longueur de deux cases et d'une largeur d'une case.
Exemple : un cavalier situé sur la case d4 atteint, en un seul déplacement, une des huit cases b5, c6, e6, f5, f3, e2, c2 et b3 (voir figure ci-contre).



Dans toute la suite de l'exercice, on appellera **case permise** toute case que le cavalier peut atteindre en un déplacement à partir de sa position.

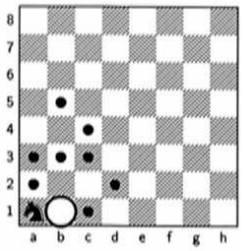
Le but de cet exercice est d'écrire un programme faisant parcourir l'ensemble de l'échiquier à un cavalier **en ne passant sur chaque case qu'une et une seule fois**.

Motivation et méthode retenue

Une première idée est de faire parcourir toutes les cases possibles à un cavalier en listant à chaque déplacement les cases parcourues. Lorsque celui-ci ne peut plus avancer on consulte le nombre de cases parcourues.

- Si ce nombre est égal à $64 = 8 \times 8$, alors le problème est résolu.
- Sinon, il faut revenir en arrière et tester d'autres chemins.

0. Exemple : on considère le parcours suivant d'un cavalier démarrant en a1 (figure ci contre) :



a1, b3, c1, a2, c3, b5, a3, c4, d2.

Avec ce début de parcours, au déplacement suivant :

a. le cavalier va en b1. Peut-il accomplir sa mission ?

b. le cavalier ne va pas en b1. Peut-il accomplir sa mission ?

Il convient donc dans la résolution du problème proposé d'éviter de se retrouver dans la situation repérée à la question **0.**

Dans tout ce qui suit, nous nommerons **coordonnées** d'une case la liste d'entiers $[i, j]$ où i représente le numéro de ligne et j le numéro de colonne (tous deux compris entre 0 et 7). Par exemple, la case b3 a pour **coordonnées** $[2, 1]$.

D'autre part, les cases sont numérotées de 0 à 63 en partant du coin gauche comme indiqué par la figure ci-contre.

8	56	57	58	59	60	61	62	63
7	48	49	50	51	52	53	54	55
6	40	41	42	43	44	45	46	47
5	32	33	34	35	36	37	38	39
4	24	25	26	27	28	29	30	31
3	16	17	18	19	20	21	22	23
2	8	9	10	11	12	13	14	15
1	0	1	2	3	4	5	6	7
	a	b	c	d	e	f	g	h

Nous appellerons **indice** d'une case le numéro n compris entre 0 et 63 ainsi déterminé. Ainsi la case b3 a pour **indice** 17.

Les questions

1. Écrire une fonction `Indice` qui aux **coordonnées** $[i, j]$ d'une case renvoie son **indice**. Ainsi, `Indice` appliquée à $[2, 1]$ doit renvoyer 17.

e3a Maths 1 PSI 2015 — Corrigé

Ce corrigé est proposé par Tristan Poullaouec (Professeur en CPGE) ; il a été relu par Benjamin Monmege (Enseignant-chercheur à l'université), Guillaume Batog (Professeur en CPGE) et Céline Chevalier (Enseignant-chercheur à l'université).

Ce sujet est constitué de quatre exercices complètement indépendants, permettant d'aborder les grands thèmes du programme.

- Il commence avec un exercice d'informatique, dans lequel on s'intéresse au déplacement d'un cavalier sur un échiquier. Au fil de questions bien détaillées et de difficulté progressive, on écrit un programme lui faisant parcourir l'ensemble de l'échiquier en passant une fois et une seule sur chaque case.
- L'exercice 2 porte sur les probabilités dénombrables : on étudie des variables aléatoires suivant une loi géométrique sur $\mathbb{N}$ et l'on utilise leurs fonctions génératrices.
- Le troisième exercice, le plus long, se penche sur une fonction définie par une intégrale à paramètre. Après avoir déterminé son sens de variation, on calcule son expression sur $\mathbb{N}$ puis, après quelques manipulations astucieuses, on en trouve des équivalents en 0 et en $+\infty$.
- Dans le quatrième et dernier exercice, on commence par établir une caractérisation originale des polynômes unitaires scindés sur $\mathbb{R}$. On l'utilise ensuite pour montrer assez élégamment que, pour tout $q \in \mathbb{N}^*$, l'ensemble des matrices trigonalisables est une partie fermée de $\mathscr{M}_q(\mathbb{R})$, contrairement à l'ensemble des matrices diagonalisables.

Ce sujet ne contient pas de réelles difficultés : les programmes de l'exercice 1 sont assez faciles à écrire vu la tournure des questions, l'exercice 2 est surtout constitué d'applications directes du cours, et les exercices 3 et 4 font appel à des techniques et des idées classiques. Par contre, c'est bien long pour une épreuve de 4 heures : il faut du temps pour rédiger proprement l'exercice 3, et aussi du temps pour bien comprendre les idées et les fonctions élaborées dans l'exercice 1. Au moins, tous les candidats auront eu de quoi s'occuper pendant les 4 heures.

Il est quand même regrettable que l'exercice de probabilités se ramène à des manipulations de séries et ne fasse aucun lien avec une situation concrète.

Indications

Exercice 1

A.1 Pour trouver l'expression de l'application qui, aux coordonnées d'une case, associe son indice, commencer par observer les variations de l'indice lorsque l'on incrémente le numéro de colonne ou le numéro de ligne.

A.2 La division euclidienne de n par 8 fait apparaître les coordonnées recherchées.

A.4 Penser à utiliser la fonction `CasA`.

A.7.2 Comme les listes A et B sont déjà triées, il suffit de comparer leurs premiers éléments et de déplacer celui de plus petite valuation dans la liste fusionnée.

A.7.3 Créer une fonction récursive, qui coupe la liste en deux et fusionne les deux nouvelles listes après les avoir triées.

Exercice 2

B.1 Il faut s'intéresser au nombre de racines du polynôme caractéristique.

B.2 Décrire E_2 et exhiber une bijection entre $\mathbb{N}$ et E_2.

B.4 On voit apparaître une loi géométrique sur $\mathbb{N}$ (et non sur $\mathbb{N}^*$, attention).

B.6 L'espérance se retrouve en dérivant la fonction génératrice.

B.7 Les dérivées successives de cette fonction donnent la loi de probabilité.

B.8 Utiliser les résultats des questions B.1 et B.7.

Exercice 3

C.1 Appliquer le critère de Riemann en 1 et en $+\infty$ à la fonction $t \longmapsto \dfrac{1}{t^x\sqrt{t^2-1}}$.

C.2 On pourra effectuer le changement de variables $u = \mathrm{e}^{\,x}$.

C.6 Partir de la définition du sens de variation d'une fonction.

C.7 Attention à bien vérifier toutes les hypothèses du théorème de dérivation des intégrales à paramètres.

C.8 Commencer par écrire $f(x) = \displaystyle\int_0^{+\infty} \frac{\mathrm{d}u}{\mathrm{ch}^{\,x}(u)} = \int_0^{+\infty} \mathrm{ch}\,(u)\,\mathrm{ch}^{\,-x-1}(u)\,\mathrm{d}u$.

C.10 Utiliser le résultat de la question C.8.

C.11 Exploiter la continuité de ϕ en 1.

C.12 Combiner les résultats des questions C.6 et C.10.

C.13 La décroissance de f permet d'encadrer $f(x)$ entre les images de deux entiers.

C.15 Commencer par prouver que ϕ admet une limite finie en $+\infty$, puis montrer grâce à la question C.10 que la suite de terme général $\phi(x+n)$ est constante.

Exercice 4

D.1.2 Utiliser le résultat de la question précédente.

D.1.3.b Penser aux racines complexes de P.

D.2.2 Pour $z \in \mathbb{C}$ fixé, montrer que l'application ψ telle que $\mathrm{P}_n(z) = \psi(\mathrm{A}_n)$ est continue.

D.2.3 Utiliser les résultats des questions D.1.2, D.2.1 puis D.1.4.

D.3.2 La diagonalisabilité dépend des dimensions des sous-espaces propres réels, elles-mêmes reliées au nombre de racines du polynôme caractéristique.

Exercice 1

A.0 Si le cavalier va en b1, les cases permises au coup suivant sont a3, c3 et d2. Mais comme elles ont déjà été occupées, il ne peut plus avancer et ne peut donc accomplir sa mission.

Si le cavalier ne va pas en b1, il ne pourra jamais plus y aller car les seules cases permettant de l'atteindre, à savoir a3, c3 et d2, ont déjà été occupées. Il ne peut pas non plus accomplir sa mission.

Ainsi, ce début de parcours empêche le cavalier d'accomplir sa mission.

A.1 Vue la numérotation imposée, l'indice d'une case augmente :

- de 1 lorsque l'on incrémente son numéro de colonne de 1 ;
- de 8 lorsque l'on incrémente son numéro de ligne de 1.

En outre, `Indice([0,0])`$= 0$ donc

$$\forall\,(i,j)\in[\![\,0\,;\,7\,]\!]^2 \qquad \texttt{Indice([i,j])} = 8i+j$$

Le code Python de cette fonction s'écrit ainsi

```
def Indice(coordonnees):
    return 8*coordonnees[0]+coordonnees[1]
```

A.2 Soit $n\in[\![\,0\,;\,63\,]\!]$. Pour tout couple $(i,j)\in[\![\,0\,;\,7\,]\!]^2$, on a

$$\texttt{Coord(n)} = [i,j] \quad\Longleftrightarrow\quad n = \texttt{Indice([i,j])} = 8i+j$$

Autrement dit, les entiers i et j sont le quotient et le reste de la division euclidienne de n par 8, puisque $0\leqslant j\leqslant 7$. En code, cela donne

```
def Coord(indice):
    return [indice/8,indice%8]
```

A.3.1 Représentons le déroulement de `CasA` à l'aide d'un tableau dans lequel figure, pour chaque valeur de d, les coordonnées $[u,v]$ résultantes ainsi que l'indice associé, lorsque ce sont les coordonnées d'une case de l'échiquier.

- Exécution de `CasA(0)` : on part de $[i,j]=$ `Coord(0)` $=[0,0]$.

d	$[1,-2]$	$[2,-1]$	$[2,1]$	$[1,2]$	$[-1,2]$	$[-2,1]$	$[-2,-1]$	$[-1,-2]$
$[u,v]$	$[1,-2]$	$[2,-1]$	$[2,1]$	$[1,2]$	$[-1,2]$	$[-2,1]$	$[-2,-1]$	$[-1,-2]$
`Indice`			17	10				

De ce fait, `CasA(0)` renvoie $[17,10]$.

- Exécution de `CasA(39)` : on part de $[i,j]=$ `Coord(39)` $=[4,7]$.

d	$[1,-2]$	$[2,-1]$	$[2,1]$	$[1,2]$	$[-1,2]$	$[-2,1]$	$[-2,-1]$	$[-1,-2]$
$[u,v]$	$[5,5]$	$[6,6]$	$[6,8]$	$[5,9]$	$[3,9]$	$[2,8]$	$[2,6]$	$[3,5]$
`Indice`	45	54					22	29

Par conséquent, `CasA(39)` renvoie $[45,54,22,29]$.

A.3.2 Cette fonction détermine les indices de toutes les cases que le cavalier peut atteindre en un coup à partir de la case d'indice n.

> En fait, en partant de la case occupée, on effectue les huit déplacements possibles (ce sont les déplacements donnés sur la première figure, décrits dans le sens horaire en partant de celui qui est en haut et à gauche) et l'on ne conserve que les cases atteintes qui appartiennent effectivement à l'échiquier.

A.4 Pour tout entier n tel que $0 \leqslant n \leqslant 63$, l'élément `ListeCA[n]` est par définition la liste `CasA(n)`, d'après ce qui précède. Voici alors la fonction demandée :

```
def Init():
    global ListeCA
    global ListeCoups
    ListeCoups=[]
    ListeCA=[]
    for n in range(64):
        ListeCA.append(CasA(n))
```

> Il ne faut surtout pas oublier l'instruction `global`, afin de préciser que les variables manipulées sont globales.
>
> On peut également construire la liste `ListeCA` de deux autres façons, par remplissage :
>
> ```
> ListeCA = [0]*64
> for n in range(64):
>     ListeCA[n] = CasA(n)
> ```
>
> ou bien en compréhension :
>
> ```
> ListeCA = [ CasA(n) for n in range(64) ]
> ```

A.5 La commande de l'énoncé renvoie la liste `CasA(0)`, c'est-à-dire $[17, 10]$. Ainsi,

La réponse correcte est la f.

A.6.1 Quand on applique la fonction `OccupePosition` à un entier n au cours de la recherche, la case d'indice n n'a jamais été occupée par le cavalier. Il est donc certain que n appartient encore à `ListeCA[k]` pour toutes les cases d'indice k dans `ListeCA[n]` (ce sont les cases permises depuis la case d'indice n). De ce fait, l'instruction `ListeCA[k].remove(n)` ne provoque pas d'erreur dans la boucle ci-dessous.

```
def OccupePosition(n):
    ListeCoups.append(n)
    SituationCritique=False
    for k in ListeCA[n]:
        ListeCA[k].remove(n)
        if ListeCA[k]==[]:
            SituationCritique=True
    return SituationCritique
```

e3a Maths 2 PSI 2015 — Corrigé

Ce corrigé est proposé par Céline Chevalier (Enseignant-chercheur à l'université) ; il a été relu par Matthias Moreno (ENS Lyon) et Sophie Rainero (Professeur en CPGE).

Ce problème, divisé en quatre parties très liées, traite d'algèbre bilinéaire euclidienne. Il s'intéresse en particulier à l'ensemble T(E) des endomorphismes u symétriques de rang inférieur ou égal à 1 vérifiant

$$\forall x \in \mathrm{E} \qquad (u(x) \mid x) \geqslant 0$$

- La partie préliminaire démontre quelques résultats préalables, certains n'ayant rien à voir avec la suite (question 3) mais permettant simplement d'identifier les candidats à l'aise avec le sujet ; d'autres résultats sont largement réutilisés dans la suite, comme la question 4 dans laquelle on montre que l'application

$$\begin{cases} \mathscr{S}(\mathrm{E})^2 \longrightarrow \mathbb{R} \\ (f,g) \longmapsto < f,g >= \mathbf{tr}(f \circ g) \end{cases}$$

  est un produit scalaire. À part la question 1 qui requiert un peu d'intuition, c'est une partie proche du cours et facile si l'on a bien compris l'algèbre linéaire.

- La partie 1 propose une caractérisation des endomorphismes de T(E). Tout endomorphisme v de T(E) peut s'écrire $v = u_a$ avec $a \in \mathrm{E}$, où u_a est l'endomorphisme de E défini par

$$\forall x \in \mathrm{E} \qquad u_a(x) = (x \mid a)a$$

  Malgré quelques difficultés techniques, cette partie reste abordable.

- La partie 2 définit, pour un endomorphisme symétrique f de E fixé, l'application $\Phi : x \in \mathrm{E} \longmapsto [\mathrm{N}(f - u_x)]^2$. Elle étudie alors la valeur de $m(f) = \inf_{x \in \mathrm{E}} \Phi(x)$ en utilisant la fonction intermédiaire, définie pour tout vecteur x de E et tout vecteur y de E de norme 1, $h_x : t \in \mathbb{R} \longmapsto \Phi(x + ty)$. Notons que $m(f)$ est la distance de f à l'espace T(E). C'est sans doute la partie la plus difficile du problème, où la multiplicité des fonctions définies peut dérouter.

- Enfin, la partie 3 applique les résultats de la partie 2 dans certains cas particuliers : tout d'abord les matrices stochastiques (c'est-à-dire celles dont les coefficients sont positifs et telles que la somme des coefficients de chaque ligne est égale à 1) et ensuite deux exemples de matrices simples. Il faut avoir bien compris les résultats obtenus dans la partie 2 avant de pouvoir les appliquer dans cette partie.

Ce problème est d'un niveau élevé pour le concours E3A, en grande partie parce que certaines questions (par exemple I.2.2 , II.3 , II.5 , II.6) sont ouvertes et réutilisées dans la suite, ce qui les rend bloquantes pour tout candidat ne les ayant pas résolues. Il offre une occasion de faire le point sur l'algèbre euclidienne et de s'entraîner à suivre le déroulement d'un énoncé complexe.

Indications

1 Montrer que l'ensemble T(E) n'est pas stable par combinaisons linéaires en considérant par exemple l'application u définie par $u(e_1) = e_1$ et $u(e_i) = 0$ pour tout $i \neq 1$, où $(e_1, \dots, e_p)$ est une base orthonormale pour le produit scalaire défini dans l'énoncé.

3 Pour l'assertion (3), se ramener à la précédente. Pour les assertions (4) et (5), utiliser le théorème du rang.

4 Un produit scalaire est une application bilinéaire symétrique définie positive.

5 Comme la somme de chaque ligne de A vaut -3, on sait que -3 est valeur propre de A, associée au vecteur propre ${}^t\begin{pmatrix}1 & 1 & 1\end{pmatrix}$. On peut déterminer les deux autres valeurs propres en utilisant les relations entre leur somme et la trace de A, ainsi que leur produit et le déterminant de A.

I.2.3 Décomposer $f(a)$ dans la base $\mathscr{B}$: $f(a) = \dfrac{(f(a) \mid a)}{\|a\|^2} a + b$, avec $b \in (\text{Vect}\,(a))^\perp$.

I.3.1 Exploiter le fait que u est de rang inférieur ou égal à 1 pour en déduire que $\text{Im}\,(u) = \text{Vect}\,(b)$.

I.3.2 Pour tout $x \in \text{E}$, il existe $\alpha \in \mathbb{R}$ tel que $u(x) = \alpha\, b$. Conclure en prenant le produit scalaire des deux membres de cette égalité avec b.

I.3.4 D'après la question I.3.2 , le vecteur a cherché est un multiple de b.

I.4 Montrer que $u_{-a} = u_a$ pour tout $a \in \text{E}$.

II.3 Utiliser l'égalité de la question II.2 : développer chaque terme en utilisant l'égalité $\|y\| = 1$ et les (bi)linéarités et symétries des applications en jeu.

II.5 D'après la question 2.3 des préliminaires, la trace de $f \circ f$ est égale à la trace de la matrice représentative de $f \circ f$ dans une base bien choisie.

II.6 Si $z \in \text{E}$ est de norme 1, il existe $(z_1, \dots, z_p) \in \mathbb{R}^p$ tel que $z = z_1 e_1 + \cdots + z_p e_p$ avec ${z_1}^2 + \cdots + {z_p}^2 = 1$. Développer alors le produit $(z \mid f(z))$.

II.7.1 La fonction h_a est minimale en 0.

II.7.2 Calculer $h_a'(0)$ avec l'expression trouvée à la question II.3 .

II.7.3 Utiliser à nouveau l'expression de la question II.3 .

II.9.1 Montrer que les conditions de la question II.7.4 sont satisfaites pour le vecteur $a = \sqrt{\lambda_p} e_p$. Exploiter ensuite les résultats des questions II.7.3 et II.5 .

II.9.2 Pour le sens direct, la question II.7.2 montre que x est un vecteur propre de f. La valeur propre associée est λ_p par maximalité.

III.1.3 Utiliser les conditions de la question II.2.2 .

III.2 Comme B est de rang 1, 0 est valeur propre de B de multiplicité $p-1$. L'autre valeur propre est déterminée comme dans la question 5 des préliminaires. La valeur de $m(f_{\text{B}})$ est donnée dans la question II.9.1 .

III.3.1 Remarquer que $\text{C} = \text{B} - \text{I}_p$.

III.3.3 Chercher un vecteur satisfaisant aux conditions de la question II.9.2 .

III.3.4 Si $w \in \text{T(E)}$ est un autre endomorphisme vérifiant l'égalité, utiliser la surjectivité de l'application φ puis les conditions de la question II.9.2 .

Préliminaires

1 Soit $(e_1, \dots, e_p)$ une base orthonormale pour le produit scalaire donné dans l'énoncé, et u l'endomorphisme défini par

$$\begin{cases} u(e_1) = e_1 \\ u(e_i) = 0 \quad \text{si } i > 1 \end{cases}$$

Puisque $(e_1, \dots, e_p)$ est une base, on a $\operatorname{rg}(u) = \operatorname{rg}(u(e_1), \dots, u(e_p)) = \operatorname{rg}((e_1)) = 1$. En outre, si $x \in \mathrm{E}$, il existe $(x_1, \dots, x_p) \in \mathbb{R}^p$ tel que $x = x_1 e_1 + \cdots + x_p e_p$. Ainsi,

$$\begin{aligned} (u(x) \mid x) &= \left(\sum_{i=1}^{p} x_i u(e_i) \middle| \sum_{j=1}^{p} x_j e_j \right) \\ &= \left(x_1 e_1 \middle| \sum_{j=1}^{p} x_j e_j \right) \\ &= \sum_{j=1}^{p} x_1 x_j \, (e_1 \,|e_j\,) \\ (u(x) \mid x) &= {x_1}^2 \geqslant 0 \end{aligned}$$

On en déduit l'appartenance de u à $\mathrm{T(E)}$.

Notons $v = -u$. Alors $v \in \mathscr{S}(\mathrm{E})$ et

$$(v(e_1) \mid e_1) = -(u(e_1) \mid e_1) = -\|e_1\|^2 < 0$$

d'où l'on déduit que $v \notin \mathrm{T(E)}$ puis que $\mathrm{T(E)}$ n'est pas stable par combinaisons linéaires. Finalement,

L'ensemble $\mathrm{T(E)}$ n'est pas un sous-espace vectoriel de $\mathscr{L}(\mathrm{E})$.

2.1 Si $(i, j) \in \{1, \dots, p\}^2$,

$$(\mathrm{AB})_{ij} = \sum_{k=1}^{p} \mathrm{A}_{ik}\mathrm{B}_{kj} \qquad \text{et} \qquad (\mathrm{BA})_{ij} = \sum_{k=1}^{p} \mathrm{B}_{ik}\mathrm{A}_{kj}$$

d'où $$\mathbf{tr}(\mathrm{AB}) = \sum_{i=1}^{p} (\mathrm{AB})_{ii} = \sum_{i=1}^{p}\sum_{k=1}^{p} \mathrm{A}_{ik}\mathrm{B}_{ki}$$

et $$\mathbf{tr}(\mathrm{BA}) = \sum_{i=1}^{p} (\mathrm{BA})_{ii} = \sum_{i=1}^{p}\sum_{k=1}^{p} \mathrm{B}_{ik}\mathrm{A}_{ki} = \sum_{k=1}^{p}\sum_{i=1}^{p} \mathrm{A}_{ki}\mathrm{B}_{ik}$$

en inversant les deux dernières sommes (ce qui est autorisé puisque les sommes sont finies). En remarquant que les rôles de i et k sont inversés,

$$\boxed{\mathbf{tr}(\mathrm{AB}) = \mathbf{tr}(\mathrm{BA})}$$

2.2 Si B est semblable à A, il existe une matrice P inversible telle que $\mathrm{B} = \mathrm{P}^{-1}\mathrm{AP}$. Ainsi,

$$\mathbf{tr}(\mathrm{B}) = \mathbf{tr}(\mathrm{P}^{-1}\mathrm{AP}) = \mathbf{tr}[\mathrm{P}^{-1}(\mathrm{AP})] = \mathbf{tr}[(\mathrm{AP})\mathrm{P}^{-1}] = \mathbf{tr}(\mathrm{A})$$

d'après la question précédente. En conclusion,

$$\boxed{\mathbf{tr}(\mathrm{B}) = \mathbf{tr}(\mathrm{A})}$$

2.3 Soit u un endomorphisme de E. Si A et B sont deux matrices représentant u dans deux bases différentes, alors A et B sont semblables (théorème du changement de base). On en déduit d'après la question précédente que $\mathbf{tr}(A) = \mathbf{tr}(B)$. On peut donc poser $\mathbf{tr}(u) = \mathbf{tr}(A)$.

La trace d'un endomorphisme de E est égale à la trace de la matrice représentant u dans une base quelconque de E.

3 Par définition,

Un hyperplan de l'espace vectoriel E de dimension p est un sous-espace vectoriel de E de dimension $p-1$.

Le complémentaire d'un espace vectoriel n'est jamais un espace vectoriel puisqu'il ne contient pas le vecteur nul. A fortiori, l'espace G ne peut donc pas être le supplémentaire de H.

L'assertion (1) est fausse.

Soient $a \in G$ et $x \in H \cap \text{Vect}(a)$. Il existe alors un scalaire $k \in \mathbb{R}$ tel que $x = ka$. Si $k \neq 0$, il vient $a = x/k \in H$, ce qui est impossible puisque $a \in G$ et que G et H sont complémentaires. Par suite, $k = 0$, puis x est le vecteur nul. Ainsi, H et $\text{Vect}(a)$ sont en somme directe. Comme $\dim(H) + \dim(\text{Vect}(a)) = p$, on en déduit qu'ils sont supplémentaires. Par conséquent,

L'assertion (2) est vraie.

Si a est un vecteur non nul et orthogonal à H, alors il n'appartient pas à H. Il appartient donc à son complémentaire, c'est-à-dire G. Ainsi, on est ramené à l'assertion précédente, ce qui signifie que

L'assertion (3) est vraie.

L'application $\mathbf{tr}$ est une application non nulle de $\mathscr{M}_p(\mathbb{R})$ dans $\mathbb{R}$, son image est donc de dimension 1, ce qui signifie que son rang vaut 1. D'après le théorème du rang, en notant $\text{Ker}(\mathbf{tr})$ le noyau de cette application, on a l'égalité

$$\dim(\text{Ker}(\mathbf{tr})) = \dim(\mathscr{M}_p(\mathbb{R})) - \text{rg}(\mathbf{tr}) = \dim(\mathscr{M}_p(\mathbb{R})) - 1$$

c'est-à-dire que le noyau de l'application $\mathbf{tr}$ est un hyperplan de $\mathscr{M}_p(\mathbb{R})$.

L'assertion (4) est vraie.

La preuve de l'assertion précédente est vraie quelle que soit l'application considérée : en utilisant le théorème du rang, un endomorphisme de E est de rang 1 si, et seulement si, son noyau est de dimension $p-1$, c'est-à-dire si et, seulement si, c'est un hyperplan.

L'assertion (5) est vraie.

4 Montrons que l'application proposée est un produit scalaire, c'est-à-dire une forme bilinéaire symétrique définie positive de $\mathscr{S}(E)^2$ dans $\mathbb{R}$.

- Tout d'abord, elle est bien à valeurs dans $\mathbb{R}$ car la trace l'est aussi.
- Elle est symétrique d'après les questions 2.1 et 2.3 .

CCP Maths PSI 2015 — Corrigé

Ce corrigé est proposé par Pauline Tan (ENS Cachan) ; il a été relu par Mathilde Perrin (Docteur en mathématiques) et Sophie Rainero (Professeur en CPGE).

Ce sujet est consacré à l'étude d'un système différentiel linéaire homogène de taille $n \in \mathbb{N}^*$ défini sur un intervalle I par

$$\forall\, t \in \mathrm{I} \qquad \mathrm{X}'(t) = \mathrm{A}(t)\,\mathrm{X}(t) \tag{E}$$

avec $\mathrm{X} : \mathrm{I} \to \mathbb{C}^n$ dérivable et $\mathrm{A} : \mathrm{I} \to \mathcal{M}_n(\mathbb{C})$ une fonction continue.

- Dans la première partie, on commence par établir deux résultats qui permettront de trouver des bases de solutions de l'équation (E). Ensuite, pour $n = 2$, on étudie l'équation (E) dans le cas où A est diagonalisable, d'abord lorsqu'elle est constante puis dans le cas général.
- La deuxième partie aborde le cas où A est constante, mais de taille n quelconque (puis égale à 4). Cette restriction permet de développer les solutions en séries entières dont les sommes s'écrivent en fonction des A^k. On montre alors que ces puissances de A sont combinaisons linéaires de A et de $\mathrm{A}(\mathrm{A} - \mathrm{I}_n)$, ce qui fournit une formule explicite pour les solutions X.
- Enfin, la troisième partie considère les fonctions u et v définies sur $\mathbb{R}$ par

  $$u : t \longmapsto \int_0^{+\infty} \frac{\mathrm{e}^{-x}\cos(tx)}{\sqrt{x}}\,\mathrm{d}x \quad \text{et} \quad v : t \longmapsto \int_0^{+\infty} \frac{\mathrm{e}^{-x}\sin(tx)}{\sqrt{x}}\,\mathrm{d}x$$

  dont on montre qu'elles sont solutions d'une équation de la forme (E). Résoudre cette équation permet de donner une formule explicite pour u et v.

Mis à part une question hors-programme dans la deuxième partie, ce sujet est relativement facile. Il couvre l'essentiel du programme sur les équations différentielles linéaires homogènes et permet également de travailler la diagonalisation. Les trois parties du sujet sont pratiquement indépendantes, seules les questions I.1 et I.2 étant utiles dans les autres parties.

Indications

Partie I

I.3 Calculer le polynôme caractéristique de A pour déterminer ses valeurs propres, puis trouver une base de vecteurs propres.

I.4.1 Si $\mathrm{X} = \begin{pmatrix} x \\ y \end{pmatrix}$, montrer que $x+y$ et $x-y$ sont solutions d'équations différentielles linéaires d'ordre 1.

I.4.2 Fixer t et déterminer valeurs propres et vecteurs propres de $\mathrm{A}(t)$.

I.4.4 Utiliser la question I.2 avec les couples $(\lambda_1, \mathrm{V}_1)$ et $(\lambda_2, \mathrm{V}_2)$ déterminés à la question I.4.2.

Partie II

II.1.1 Commencer par montrer que $\mathrm{N}(\mathrm{A})$ est bien définie pour toute matrice A, puis établir les trois propriétés qui caractérisent les normes.

II.2.1 Raisonner par récurrence.

II.2.2 Appliquer la formule de Taylor avec reste intégrale à chacune des coordonnées de X.

II.2.3 Cette question est hors-programme. Admettre que, pour $t \in \mathrm{I}$, tel que $t \geqslant 0$,

$$\left\| \int_0^t \frac{(t-u)^p}{p!} \mathrm{A}^{p+1}\mathrm{X}(u)\,\mathrm{d}u \right\| \leqslant \int_0^t \left\| \frac{(t-u)^p}{p!} \mathrm{A}^{p+1}\mathrm{X}(u) \right\| \mathrm{d}u$$

puis utiliser la question II.2.2 pour majorer $\|\mathrm{A}^{p+1}\mathrm{X}(u)\|$. Enfin, les croissances comparées permettent de prouver que le membre de droite dans l'inégalité ci-dessus tend vers 0 quand p tend vers $+\infty$.

II.3.2 Écrire la division euclidienne, en exprimant le reste dans la base proposée, puis évaluer en 0 et 1. Penser également à simplifier par $\mathrm{X}(\mathrm{X}-1)$ et à dériver, avant d'évaluer à nouveau en 1.

II.3.3 Penser à utiliser le théorème de Cayley-Hamilton.

II.3.5 Commencer par travailler avec les sommes partielles.

II.3.6 Utiliser les questions II.2.3, II.3.3, II.3.4 et II.3.5.

Partie III

III.1.2 Utiliser le théorème fondamental de l'analyse pour F et le théorème de dérivation des intégrales à paramètre pour G.

III.1.3 Effectuer le changement de variable $u = x\,t$ dans l'une des intégrales, puis calculer $\mathrm{F}(0) + \mathrm{G}(0)$.

III.1.4 Commencer par prouver que $\mathrm{G}(x)$ tend vers 0 quand x tend vers $+\infty$.

III.2.1 Utiliser le théorème de dérivation des intégrales à paramètre.

III.2.2 Calculer w et w' en les écrivant en fonction de u, v, u' et v' pour déterminer l'équation différentielle demandée. Ensuite, identifier les parties réelle et imaginaire.

III.2.4 Appliquer les questions I.2 et I.1 pour trouver une base de solutions, que l'on calcule explicitement.

III.2.5 Établir l'expression générale de u et de v grâce à la question III.2.4, puis calculer de deux manières différentes $u(0)$ et $v(0)$.

I. Quelques exemples d'étude d'un système différentiel

I.1 Le théorème de Cauchy-Lipschitz linéaire assure que

> L'ensemble des solutions de l'équation (E) forme un $\mathbb{C}$-espace vectoriel de dimension n.

I.2 La fonction $\mathrm{X} : t \mapsto \alpha(t)\,\mathrm{V}$ est solution de l'équation (E) si et seulement si elle est de classe $\mathscr{C}^1$ et vérifie

$$\forall\, t \in \mathrm{I} \qquad \mathrm{X}'(t) = \mathrm{A}(t)\,\mathrm{X}(t)$$

Or, la fonction X est de classe $\mathscr{C}^1$ si et seulement si la fonction α est elle-même de classe $\mathscr{C}^1$, avec dans ce cas

$$\forall\, t \in \mathrm{I} \qquad \mathrm{X}'(t) = \alpha'(t)\,\mathrm{V}$$

On en déduit que X est solution de (E) si et seulement si α est $\mathscr{C}^1$ et

$$\forall\, t \in \mathrm{I} \qquad \alpha'(t)\,\mathrm{V} = \mathrm{A}(t)\,\big(\alpha(t)\,\mathrm{V}\big) = \alpha(t)\,\mathrm{A}(t)\,\mathrm{V} \qquad\qquad (\alpha(t) \in \mathbb{C})$$

Par hypothèse, on a $\mathrm{A}(t)\,\mathrm{V} = \lambda(t)\,\mathrm{V}$ pour tout $t \in \mathrm{I}$. Il s'ensuit que la fonction X est solution de (E) si et seulement si α est $\mathscr{C}^1$ et

$$\forall\, t \in \mathrm{I} \qquad \alpha'(t)\,\mathrm{V} = \lambda(t)\,\alpha(t)\,\mathrm{V}$$

soit, puisque V est supposé non nul, si et seulement si

$$\forall\, t \in \mathrm{I} \qquad \alpha'(t) = \lambda(t)\,\alpha(t)$$

Par conséquent,

> La fonction $\mathrm{X} : t \longmapsto \alpha(t)\,\mathrm{V}$ est solution de (E) si et seulement si la fonction α est $\mathscr{C}^1$ et α est solution de l'équation différentielle $\alpha'(t) = \lambda(t)\,\alpha(t)$. Pour tout $t_0 \in \mathrm{I}$ fixé, les solutions de cette équation différentielle sont données par
>
> $$t \longmapsto \mathrm{C}\,\exp\left(\int_{t_0}^{t} \lambda(u)\,\mathrm{d}u\right) \qquad \text{avec } \mathrm{C} \in \mathbb{C}.$$

I.3 L'équation différentielle étudiée correspond à un système différentiel linéaire à coefficients constants. Il faut donc commencer par calculer les valeurs propres de la matrice A. Pour cela, considérons son polynôme caractéristique :

$$\mathrm{P_A}(\mathrm{X}) = \det(\mathrm{X}\,\mathrm{I}_2 - \mathrm{A}) = \begin{vmatrix} \mathrm{X} - a & -1 + a \\ -b & \mathrm{X} - 1 + b \end{vmatrix} = \mathrm{X}^2 - (a - b + 1)\,\mathrm{X} + a - b$$

Déterminons les racines de ce polynôme : son discriminant vaut

$$\Delta = (a - b + 1)^2 - 4\,(a - b) = (a - b)^2 + 2\,(a - b) + 1 - 4\,(a - b) = (a - b - 1)^2$$

D'après les hypothèses sur a et b qui assurent que $a - b - 1$ est non nul, le discriminant Δ est également non nul. Par conséquent, la matrice A est diagonalisable, de valeurs propres distinctes

$$\lambda_1 = \frac{(a - b + 1) + (a - b - 1)}{2} = a - b$$

et

$$\lambda_2 = \frac{(a - b + 1) - (a - b - 1)}{2} = 1$$

Déterminons à présent les sous-espaces propres associés à ces deux valeurs propres. Commençons par le sous-espace propre $\mathrm{E}_{\lambda_1}(\mathrm{A})$ associé à λ_1. Le vecteur $\mathrm{V} = \begin{pmatrix} x \\ y \end{pmatrix}$ appartient à $\mathrm{E}_{\lambda_1}(\mathrm{A})$ si et seulement s'il vérifie

$$\mathrm{A}\,\mathrm{V} = \lambda_1\,\mathrm{V} \qquad \text{soit} \qquad \begin{cases} a\,x + (1-a)\,y = (a-b)\,x \\ b\,x + (1-b)\,y = (a-b)\,y \end{cases}$$

En simplifiant, on obtient

$$\begin{cases} (1-a)\,y = -b\,x \\ b\,x + y = a\,y \end{cases} \qquad \Longleftrightarrow \qquad b\,x + (1-a)\,y = 0$$

En particulier, on en déduit que le sous-espace propre $\mathrm{E}_{\lambda_1}(\mathrm{A})$ est généré par le vecteur non nul $\mathrm{V}_1 = \begin{pmatrix} a-1 \\ b \end{pmatrix}$. En résolvant ensuite le système

$$\mathrm{A}\,\mathrm{V} = \lambda_2\,\mathrm{V} \qquad \text{soit} \qquad \begin{cases} a\,x + (1-a)\,y = x \\ b\,x + (1-b)\,y = y \end{cases}$$

on démontre de la même façon que le sous-espace propre $\mathrm{E}_{\lambda_2}(\mathrm{A})$ associé à la valeur propre λ_2 est généré par le vecteur non nul $\mathrm{V}_2 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$. L'ensemble des solutions est de dimension 2 et les deux fonctions $t \mapsto \mathrm{e}^{\lambda_1 t}\mathrm{V}_1$ et $t \mapsto \mathrm{e}^{\lambda_2 t}\mathrm{V}_2$ sont linéairement indépendantes. Ainsi, les solutions de l'équation (E) sont de la forme

$$t \mapsto \mathrm{C}_1\,\mathrm{e}^{\lambda_1 t}\mathrm{V}_1 + \mathrm{C}_2\,\mathrm{e}^{\lambda_2 t}\mathrm{V}_2 \qquad \text{avec } (\mathrm{C}_1, \mathrm{C}_2) \in \mathbb{C}^2.$$

Les fonctions

$$t \longmapsto \mathrm{e}^{(a-b)t}\begin{pmatrix} a-1 \\ b \end{pmatrix} \quad \text{et} \quad t \longmapsto \mathrm{e}^{t}\begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

forment une base des solutions de l'équation (E).

I.4.1 Si $\mu = 1$, alors $\qquad \forall\, t \in \mathrm{I} \qquad \mathrm{A}(t) = \begin{pmatrix} a(t) & b(t) \\ b(t) & a(t) \end{pmatrix}$

Soit $\mathrm{X} = \begin{pmatrix} x \\ y \end{pmatrix}$ avec x et y deux fonctions $\mathscr{C}^1$ de I dans $\mathbb{C}$. La fonction X est solution de l'équation (E) si et seulement si

$$\forall\, t \in \mathrm{I} \qquad \begin{cases} x'(t) = a(t)\,x(t) + b(t)\,y(t) \\ y'(t) = b(t)\,x(t) + a(t)\,y(t) \end{cases}$$

En considérant la somme puis la différence des deux équations de ce système linéaire, on en déduit qu'il est équivalent au système

$$\forall\, t \in \mathrm{I} \qquad \begin{cases} x'(t) + y'(t) = \big(a(t) + b(t)\big)\,\big(x(t) + y(t)\big) \\ x'(t) - y'(t) = \big(a(t) - b(t)\big)\,\big(x(t) - y(t)\big) \end{cases}$$

soit, en posant $z = x + y$ et $v = x - y$,

$$\forall\, t \in \mathrm{I} \qquad \begin{cases} z'(t) = \big(a(t) + b(t)\big)\,z(t) \\ v'(t) = \big(a(t) - b(t)\big)\,v(t) \end{cases}$$

CCP Informatique PSI 2015 — Corrigé

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

De nouvelles techniques chirurgicales, dites mini-invasives, permettent d'opérer un patient en faisant passer de longs instruments chirurgicaux et une caméra au travers de fines canules cylindriques traversant la peau. Afin d'aider le chirurgien au cours de l'opération, un robot pilote le déplacement de la caméra pour suivre le mouvement des instruments chirurgicaux. Le sujet propose de réaliser une partie du programme pour ce suivi automatique, fondé sur un traitement mathématique des images extraites du flux vidéo.

- Les questions 1 à 6 sont une petite mise en jambe où l'on demande d'écrire de courts programmes qui ne sont pas utiles pour la suite. Les notions abordées sont très variées : expressions booléennes, requêtes SQL, tableaux à 3 dimensions, quantité de mémoire. On se familiarise avec la représentation d'une fenêtre rectangulaire d'une image et le codage en niveaux de gris.
- Le cœur de l'algorithme est programmé dans les questions 7 à 13. Étant donné une fenêtre entourant un objet dans l'image $\mathrm{I}(t)$ à l'instant t, on souhaite calculer une nouvelle fenêtre entourant ce même objet dans l'image $\mathrm{I}(t + \mathrm{d}t)$. Pour cela, on extrait des pixels d'intérêt dans $\mathrm{I}(t)$. On calcule le déplacement de chacun d'eux dans l'image $\mathrm{I}(t + \mathrm{d}t)$ à partir des variations d'intensité spatiales et temporelle autour des pixels. Cette phase de modélisation est difficile à comprendre à cause de quelques sous-entendus dans l'énoncé. Toutefois, il est possible de traiter indépendamment chacune des questions si la démarche générale est comprise.
- Le calcul des nouveaux pixels d'intérêt dans l'image $\mathrm{I}(t + \mathrm{d}t)$ est heuristique. Les questions 14 à 19 proposent trois tests pour éliminer les nouveaux pixels qui ne conviennent pas. La dernière question utilise une fonction très générale dont la documentation est donnée en anglais en annexe. Réussir à faire le lien entre les structures manipulées dans le sujet et les spécifications de cette fonction suppose une compréhension en profondeur du sujet et du cours.
- Les questions 20 et 21 étudient un code servant à calculer les dimensions d'une nouvelle fenêtre entourant la plupart des pixels d'intérêt dans la nouvelle image $\mathrm{I}(t + \mathrm{d}t)$. On demande de modifier et commenter ce code pour pouvoir l'utiliser dans le programme principal.

Cette épreuve évalue un large panel de compétences du programme d'informatique commune, dont certaines sont difficiles à mettre en œuvre en temps limité. Elle comporte aussi des applications directes du cours : algorithme de tri, calculs de produits matriciels, méthode du pivot de Gauss. C'est pourquoi elle constitue un excellent support d'entraînement, notamment pour réviser le programme de première année.

Indications

2 Il suffit de vérifier que les dimensions sont positives et que les pixels supérieur gauche et inférieur droit de la fenêtre se trouvent bien dans l'image.

6 Utiliser deux boucles `for` imbriquées pour traiter chacun des pixels de l'image.

7 Les points à extraire ont pour coordonnées $(\mathrm{P}x + i\,\Delta\mathrm{L}, \mathrm{P}y + j\,\Delta\mathrm{H})$ avec $\Delta\mathrm{L}$ (resp. $\Delta\mathrm{H}$) le nombre de pixels pour passer d'une colonne (resp. ligne) des points à extraire à la suivante

8 Écrire les approximations au premier ordre de $\mathrm{I}(x + \mathrm{d}x, y, t)$ et $\mathrm{I}(x - \mathrm{d}x, y, t)$. Pour le calcul de `Ix`, prendre $\mathrm{d}x = 1$.

9 Déterminer entre quelles valeurs se trouvent les abscisses x et les ordonnées y des pixels du patch. Utiliser deux boucles `for` imbriquées en x et y pour construire la liste `patch`.

10 La matrice A possède autant de lignes qu'il y a d'équations $I_x\mathrm{d}x + I_y\mathrm{d}y = -I_t\mathrm{d}t$, c'est-à-dire une par pixel du patch. Elle possède 2 colonnes correspondant aux variables $\mathrm{d}x$ et $\mathrm{d}y$. Attention : le pixel numéro k de la liste `patch` a pour coordonnées `patch[2*k]` et `patch[2*k+1]` (où le premier pixel est numéroté 0).

11 Pour calculer $\mathrm{A}^{\mathrm{T}}\mathrm{A}$ et $\mathrm{A}^{\mathrm{T}}b$, programmer des fonctions qui calculent respectivement la transposée et le produit de matrices générales. Résoudre le système 2×2 obtenu à l'aide de l'algorithme du pivot de Gauss. Il est inutile de programmer l'algorithme général vu en cours.

12 Pour chaque pixel d'intérêt de `pts`, appliquer successivement les fonctions des questions 9, 10 et 11 avec un patch de taille 5. On obtient le déplacement $(\mathrm{d}x, \mathrm{d}y)$ du pixel d'intérêt.

15 Pour un tableau `t`, la notation `t[0::2]` (resp. `t[1::2]`) consiste à extraire toutes les cases d'indices pairs (resp. impairs) de `t`.

16 Utiliser n'importe quel algorithme de tri vu en cours.

19 Programmer une fonction qui construit une liste `ssimg` des valeurs `img[x][y]` correspondant à un patch d'une image `img`. Retourner plus précisément la sous-image `[ssimg]` de taille $1 \times \ell^2$, où ℓ est la taille du patch. Utiliser ensuite la fonction `match_template` avec les sous-images correspondant au patch de `imgI` autour de `P` et à celui de `imgJ` autour de `Pn`, où `Pn` est la nouvelle position de `P`.

20 Regarder de plus près la structure de `pt0` et `pt1`.

21 Expliquer ce que calcule chaque morceau de code, sans entrer dans le détail de la façon d'obtenir le résultat.

Robot Evolap – Suivi d'instrument chirurgical

1 On demande successivement 4 valeurs qu'on renvoie sous forme d'un tableau.

```
def demande_fenetre():
    Px = demande_valeur("Valeur de Px = ")
    Py = demande_valeur("Valeur de Py = ")
    L  = demande_valeur("Valeur de L  = ")
    H  = demande_valeur("Valeur de H  = ")
    return [Px,Py,L,H]
```

2 La fenêtre est définie correctement si et seulement si elle est non vide et ses coins en haut à gauche et en bas à droite se trouvent bien dans l'image. On exprime chacune de ces conditions à l'aide des variables booléennes `nonvide`, `coinHG` et `coinBD` respectivement. On retourne le résultat de la conjonction (`and`) de ces 3 variables.

```
def verification_fenetre(image_width,image_height,fenetre):
    [Px,Py,L,H] = fenetre
    iw,ih   = image_width, image_height  # quelques raccourcis
    nonvide = ( L>0 and H>0 )
    coinHG  = ( Px  >=0 and Py  >=0 and Px  <iw and Py  <ih )
    coinBD  = ( Px+L>=0 and Py+H>=0 and Px+L<iw and Py+H<ih )
    return ( nonvide and coinHG and coinBD )
```

> Voici une solution avec moins de tests à effectuer :
>
> ```
> def verification_fenetre(image_width,image_height,fenetre):
>     [Px,Py,L,H] = fenetre
>     nonvide = ( L>0 and H>0 )
>     coinHGbis = ( Px>=0 and Py>=0 )
>     coinBDbis = ( Px+L < image_width and Py+H < image_height )
>     return ( nonvide and coinHGbis and coinBDbis )
> ```
>
> En reprenant le nom des variables du corrigé, remarquons que `nonvide` et `coinHGbis` impliquent que $\mathrm{P}x + \mathrm{L} \geqslant 0$ et $\mathrm{P}y + \mathrm{H} \geqslant 0$ donc il n'est pas nécessaire de réaliser ces tests dans `coinBDbis`. De même, `nonvide` et `coinBDbis` impliquent que $\mathrm{P}x <$ `iw` et $\mathrm{P}y <$ `ih`.

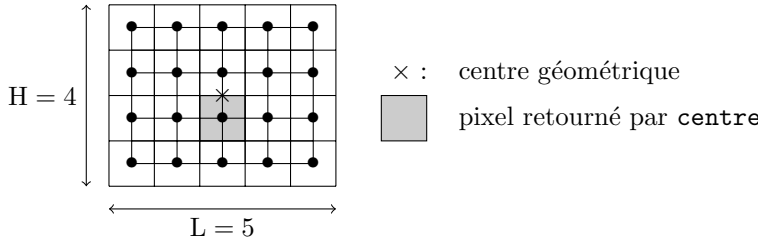
3 Grâce à l'égalité (via `ON`) des identifiants d'instruments id dans la table INSTRUMENTS et mid dans la table DEFINITIONS, on effectue une jointure (via `JOIN`) pour mettre en correspondance les enregistrements de ces deux tables pour un même instrument. Dans cette jointure, on sélectionne (via `WHERE`) les enregistrements correspondant aux pinces puis on extrait (via `SELECT`) la colonne d'attribut des noms de fichiers images.

```
SELECT filename
   FROM DEFINITIONS
   JOIN INSTRUMENTS ON INSTRUMENTS.id = mid
   WHERE name = "pince";
```

> Puisque l'attribut id apparaît dans les deux tables, on distingue les deux attributs id en les faisant précéder du nom de la table (`INSTRUMENTS.id` pour celui de la table INSTRUMENTS).

4 Le centre de la fenêtre se situe au milieu de la largeur et de la hauteur d'où

```
def centre(fenetre):
    [Px,Py,L,H] = fenetre
    return ( Px+L//2, Py+H//2 )
```

> Si les dimensions $\mathrm{L} = 2\ell + 1$ et $\mathrm{H} = 2h + 1$ sont impaires, alors le centre de la fenêtre se trouve en position $(\mathrm{P}x + \ell, \mathrm{P}y + h)$. On obtient ℓ et h par quotient d'une division euclidienne par 2, ce que permet l'opérateur `//` sur les variables de type `int`. Si l'une des dimensions H ou L est paire, le centre géométrique de la fenêtre ne correspond plus au centre d'un pixel de l'image. Toutefois, il se trouve sur le bord du pixel retourné par la fonction `centre`.
>
> 
>
>
> L'énoncé n'indique pas clairement si l'on doit renvoyer les coordonnées du centre géométrique, de type `float`, ou les coordonnées d'un pixel, de type `int`, comme proposé dans le corrigé.

5 Le tableau possède $3 \times m \times n = 3 \times 800 \times 600 = 1\,440\,000$ valeurs. Chaque valeur est un entier compris entre 0 et 255. La représentation binaire d'une valeur possède au plus 8 chiffres 0 ou 1 car $2^8 = 256$ donc on peut la stocker dans un octet. Finalement,

Le tableau représentant l'image nécessite 1,44 millions d'octets de mémoire.

> Rappelons que 1 Ko = 1 024 octets et 1 Mo = 1 024 Ko. Ainsi, 1,44 millions d'octets correspond à environ 1,37 Mo.

6 Après avoir initialisé une image avec des valeurs nulles, deux boucles `for` imbriquées permettent de traiter successivement chacun des pixels de l'image.

```
def grayscale(imagecolor):
    [c,l] = imagecolor.shape # récupération des dimensions
    image = zeros((c,l))     # initialisation de l'image
    # remplissage des valeurs de l'image, case par case
    for i in range(c):
        for j in range(l):
            # traitement du pixel de coordonnees (i,j)
            r = imagecolor[0][i][j]
            g = imagecolor[1][i][j]
            b = imagecolor[2][i][j]
            mini = min(min(r,g),b)  # version où min et max ne
            maxi = max(max(r,g),b)  # travaillent que sur 2 éléments
            image[i][j] = (mini+maxi)//2
    return image
```

Centrale Maths 1 PSI 2015 — Corrigé

Ce corrigé est proposé par Céline Chevalier (Enseignant-chercheur à l'université) ; il a été relu par Matthias Moreno (ENS Lyon) et Benjamin Monmege (Enseignant-chercheur à l'université).

Le problème a pour objet le processus de Galton-Watson, qui modélise le développement d'une population. C'est un processus très célèbre car il fut le premier à quantifier la probabilité d'extinction d'une population.

- La première partie établit des résultats préliminaires qui portent uniquement sur les suites et séries de réels. Mis à part les premières questions, qui ont pu dérouter certains candidats par leur originalité, cette partie est tout à fait classique et sans grande difficulté.
- La deuxième partie définit les variables aléatoires

$$\mathrm{S}_0 = 0 \qquad \text{et} \qquad \forall n \geqslant 1 \quad \mathrm{S}_n = \sum_{k=1}^{n} \mathrm{X}_k$$

  où $(\mathrm{X}_n)_{n\in\mathbb{N}^*}$ est une suite de variables aléatoires indépendantes à valeurs dans $\mathbb{N}$ qui ont la même loi que X. À l'aide des fonctions génératrices, on démontre la formule de Wald : $\mathrm{E}(\mathrm{S_T}) = \mathrm{E(T)E(X)}$, valable lorsque X et le temps d'arrêt T, à valeurs dans $\mathbb{N}$, sont d'espérance finie. Cette partie se conclut par l'exemple concret d'une ponte d'insectes.
- La troisième partie s'intéresse à un processus de Galton-Watson, c'est-à-dire une famille $(\mathrm{Y}_n)_{n\in\mathbb{N}}$ de variables aléatoires définies par $\mathrm{Y}_0 = 1$ et

$$\mathrm{Y}_{n+1} = \begin{cases} 0 & \text{si } \mathrm{Y}_n = 0 \\ \displaystyle\sum_{i=1}^{\mathrm{Y}_n} \mathrm{X}_{n,i} & \text{sinon} \end{cases}$$

  où les variables aléatoires $(\mathrm{X}_{n,i})_{n,i\in\mathbb{N}\times\mathbb{N}^*}$ sont indépendantes et de même loi. Les variables Y_n représentent le nombre d'individus à la génération n, et les variables $\mathrm{X}_{n,i}$ le nombre de fils (d'espérance m). On montre que la probabilité d'extinction est égale à 1 lorsque $m \leqslant 1$, avant d'étudier la lignée en déterminant l'espérance du nombre total d'individus.
- La quatrième partie étudie le cas particulier où les $\mathrm{X}_{i,n}$ suivent une loi géométrique de paramètre $1/2$. Les questions portent uniquement sur des calculs ; elles peuvent être abordées sans avoir entièrement résolu la troisième partie.
- Enfin, la cinquième partie est dans le prolongement de la troisième, en proposant l'étude de la lignée dans le cas, appelé sur-critique, où $m > 1$. Dans ce cas, soit la population s'éteint, soit le nombre d'individus à chaque génération diverge vers l'infini. C'est la partie la plus difficile du problème.

C'est un très beau sujet sur les probabilités, qui étaient au programme pour la première fois, mêlant théorie et applications. Intéressant, ambitieux, il a cependant dû déstabiliser nombre de candidats en raison de sa difficulté technique et des nombreuses notations qu'il introduit. Il est parfait pour travailler ou réviser tout le programme de probabilités.

Indications

Partie I

I.A.3 Montrer que $\ell \leqslant x_f$ puis utiliser la minimalité de x_f.

I.B Trouver une autre solution de l'équation $f(x) = x$ en utilisant le théorème des valeurs intermédiaires.

I.C Pour la deuxième partie de la question, montrer que f est strictement croissante sur un voisinage de 1.

I.D.1 Appliquer la formule de Taylor-Young à l'ordre 2 à f.

I.E.1 Exploiter la règle de d'Alembert ainsi que le calcul de la question I.D.1 .

Partie II

II.A.3 La suite $(\mathrm{T} = k)_{k\in\mathbb{N}}$ forme un système complet d'événements, ce qui permet d'appliquer la formule des probabilités totales.

II.B Si Y est une variable aléatoire, alors Y est d'espérance finie si, et seulement si, $\mathrm{G_Y}$ est dérivable en 1 et dans ce cas, $\mathrm{E(Y)} = \mathrm{G_Y}'(1)$.

II.C.1 Si Y suit une loi de Poisson de paramètre $\lambda > 0$ alors, pour tout $k \in \mathbb{N}$, $\mathrm{P(Y} = k) = \mathrm{e}^{-\lambda}\lambda^k/k!$.

Partie III

III.A.2 Utiliser la question II.B .

III.A.3.a L'extinction est l'union des événements $(\mathrm{Y}_n = 0)$ pour $n \in \mathbb{N}$.

III.A.3.b Montrer tout d'abord que, pour tout $n \geqslant 0$, $u_n = f^n(0)$ où f^n est la composée n fois de f, puis que f vérifie les hypothèses de la partie I.

III.B.1 L'événement $(\mathrm{T} = n)$ est inclus dans l'événement $(\mathrm{Y}_{n-1} \neq 0)$.

III.C.2.a Prouver que la suite d'événements $((\mathrm{Z}_n \leqslant k))_{n\in\mathbb{N}^*}$ est décroissante, puis utiliser le théorème de continuité décroissante.

III.C.2.c Découper la somme dans la valeur absolue en deux, utiliser l'inégalité triangulaire, puis majorer s par 1 dans la première somme, et la différence des probabilités par 1 dans la deuxième.

III.C.2.d Commencer par fixer K pour majorer le deuxième terme de l'inégalité de la question précédente. Faire ensuite tendre n vers l'infini.

III.C.3.a Appliquer la question II.A.1 .

III.C.3.c Prouver l'existence de $\mathrm{E(Z)} = \mathrm{G_Z}'(1)$ grâce au théorème de dérivation par limite de la dérivée.

Partie IV

IV.E Développer φ_n en série entière afin de reconnaître les valeurs $\mathrm{P(Y}_n = k)$ pour tout entier k. Pour cela, effectuer la division euclidienne du numérateur par le dénominateur.

IV.F Utiliser l'égalité, valable pour $n \in \mathbb{N}^*$, $(\mathrm{T} > n) = (\mathrm{Y}_n \geqslant 1)$.

IV.G Résoudre l'équation donnée par la question III.C.3.b . Développer ensuite la fonction $\mathrm{G_Z}$ obtenue en série entière.

Partie V

V.A Montrer la convergence absolue des séries.

V.B.1 Exprimer W_1 en fonction des variables $X_{0,i}$ pour $i \in \{1, \dots, k\}$.

V.B.2 Pour le cas $p_0 = 0$, montrer tout d'abord que la population ne peut que croître. La considérer ensuite à la génération 1.

Pour le cas $p_0 > 0$, regarder ce qui se passe si les individus de la génération 0 n'ont pas de fils.

V.C.1 Partitionner selon le rang où la valeur k est atteinte pour la première fois.

V.C.2 Raisonner par récurrence et reconnaître un produit de Cauchy grâce au résultat de la question précédente.

V.D.1 Définir les événements « la suite $(W_n)_{n\in\mathbb{N}^*}$ prend la valeur k au moins r fois » pour tout $r \geqslant 1$, et les exprimer en fonction des événements apparus dans la question V.A . Utiliser ensuite les questions V.C.2 et V.B.2 pour calculer les probabilités.

V.D.2 Utiliser la formule des probabilités totales pour ramener la suite $(Y_n)_{n\in\mathbb{N}^*}$ à la suite $(W_n)_{n\in\mathbb{N}^*}$ une fois la valeur k prise à un certain rang.

V.E C'est une application du théorème de continuité croissante.

V.F S'inspirer des événements utilisés à la question V.D.2 et conclure grâce à la question précédente.

I. Étude d'une suite récurrente

I.A.1 Afin de prouver la croissance de la suite (u_n), montrons, par récurrence sur $n \in \mathbb{N}$, que, pour tout $n \in \mathbb{N}$, la propriété

$$\mathscr{P}(n) : u_n \leqslant u_{n+1}$$

est vraie.

- $\mathscr{P}(0)$: par définition, $u_0 = 0$ et, par hypothèse, f est à valeurs dans $[\,0\,;1\,]$, donc $u_1 = f(u_0) \in [\,0\,;1\,]$, d'où $u_0 \leqslant u_1$.
- $\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)$: on a $u_{n+2} = f(u_{n+1})$ et $u_{n+1} = f(u_n)$. D'après l'énoncé, f' est positive, donc f est croissante. Par hypothèse de récurrence, $u_n \leqslant u_{n+1}$ donc $u_{n+1} = f(u_n) \leqslant f(u_{n+1}) = u_{n+2}$ en composant par f.
- Conclusion : $\forall n \geqslant 0 \qquad u_n \leqslant u_{n+1}$

  Comme f est à valeurs dans $[\,0\,;1\,]$, la suite (u_n) est également à valeurs dans $[\,0\,;1\,]$. Elle est ainsi croissante et majorée, donc elle est convergente. Par suite,

$$\boxed{\text{La suite } (u_n) \text{ est croissante et convergente.}}$$

I.A.2 Notons E l'ensemble $\{x \in [\,0\,;1\,] \mid f(x) = x\}$. Cet ensemble est non vide car il contient 1 et il est minoré par 0 ; il admet donc une borne inférieure. En outre, en posant $g : x \longmapsto f(x) - x$ l'application de $[\,0\,;1\,]$ dans $\mathbb{R}$, g est continue et l'ensemble E est l'image réciproque par g de l'ensemble $\{0\}$, qui est fermé. On en déduit que E est fermé. Sa borne inférieure est donc son minimum.

$$\boxed{\text{L'équation } f(x) = x \text{ admet une plus petite solution.}}$$

> On aurait également pu résoudre cette question en raisonnant avec des suites. En effet, si l'on note α la borne inférieure de E, alors il existe une suite (x_n) d'éléments de E qui converge vers α. En outre, pour tout entier n, $f(x_n) = x_n$ et f est continue. En passant à la limite, on en déduit que $f(\alpha) = \alpha$, puis que $\alpha \in \mathrm{E}$: c'est donc bien un minimum.

I.A.3 Pour tout $n \in \mathbb{N}$, $u_{n+1} = f(u_n)$. En outre, f est continue donc, en passant à la limite quand n tend vers l'infini, $f(\ell) = \ell$. Le réel ℓ est ainsi solution de l'équation $f(x) = x$.

Montrons à présent que $\ell \leqslant x_f$. Comme f est croissante sur $[\,0\,;x_f\,]$,

$$f([\,0\,;x_f\,]) = [\,f(0)\,;f(x_f)\,] = [\,0\,;x_f\,]$$

En outre, $u_0 = 0 \in [\,0\,;x_f\,]$ donc, par récurrence immédiate, $u_n \in [\,0\,;x_f\,]$ pour tout entier n. En passant une nouvelle fois à la limite, il vient $\ell \leqslant x_f$.

Par minimalité de la solution x_f de l'équation $f(x) = x$, on obtient

$$\boxed{\ell = x_f}$$

I.B Considérons la fonction dérivable $g : x \longmapsto f(x) - x$ définie dans la question I.A.2. Pour tout $x \in [\,0\,;1\,]$, $g'(x) = f'(x) - 1$. D'après l'énoncé, $g'(1) > 0$ et $g(1) = 0$, donc il existe $x_0 \in [\,0\,;1\,[$ tel que $g(x_0) < 0$. En outre, on a l'inégalité

Centrale Maths 2 PSI 2015 — Corrigé

Ce corrigé est proposé par Michel Blockelet (ENS Cachan) et Guillaume Batog (Professeur en CPGE) ; il a été relu par Nicolas Weiss (Professeur agrégé) et Benjamin Monmege (Enseignant-chercheur à l'université).

Ce sujet porte sur les fonctions harmoniques à deux variables. Ce sont les fonctions réelles u de classe $\mathscr{C}^2$ sur un ouvert Ω de $\mathbb{R}^2$ dont le laplacien Δu est nul. Elles interviennent dans les solutions de l'équation de la chaleur avec conditions aux limites, que l'on retrouve dans de nombreuses applications physiques.

- La partie I, préliminaire, donne quelques propriétés simples des polynômes harmoniques et des polynômes de deux variables s'annulant sur un ouvert non vide Ω. Une dose de topologie et d'algèbre linéaire interviennent.
- La partie II propose quelques exemples de fonctions harmoniques où l'on applique des résultats de stabilité par dérivation et par changement homothétique de domaine. On étudie plus particulièrement le noyau de Poisson

$$\mathrm{N}(x,y,t) = \frac{1-(x^2+y^2)}{(x-\cos t)^2+(y-\sin t)^2}$$

  avec (x,y) dans le disque unité ouvert $\mathrm{D}(0,1)$ et $t \in \mathbb{R}$. Cette partie fait appel à un peu de géométrie sur les complexes, aux théorèmes généraux de dérivabilité dans $\mathbb{R}$ et dans $\mathbb{R}^2$, et à un théorème d'échange des signes somme et intégrale dans la dernière question.
- La partie III s'intéresse au problème de Dirichlet sur le disque unité fermé $\overline{\mathrm{D}}(0,1)$ où l'on cherche les fonctions continues sur $\overline{\mathrm{D}}(0,1)$, harmoniques sur $\mathrm{D}(0,1)$ et qui coïncident avec une fonction réelle f sur le cercle unité. Dans un premier temps, on montre qu'une telle solution est donnée par

$$\forall\,(x,y) \in \mathrm{D}(0,1) \qquad \mathrm{N}_f(x,y) = \frac{1}{2\pi}\int_0^{2\pi} \mathrm{N}(x,y,t)\, f(\cos t, \sin t)\,\mathrm{d}t$$

  C'est la partie la plus difficile du sujet. Elle demande une très bonne maîtrise technique lorsqu'il faut dériver sous le signe intégrale ou manipuler jusqu'à trois inégalités avec des ε, δ et η (dépendants entre eux bien sûr !) pour établir la continuité en tout point du cercle. Dans un deuxième temps, on démontre l'unicité de cette solution avec des outils simples d'analyse sur $\mathbb{R}$.
- La partie IV se focalise sur le problème de Dirichlet avec une fonction f polynomiale sur le cercle, de degré au plus $m \in \mathbb{N}^*$. À l'aide d'arguments d'algèbre linéaire, on montre que l'unique solution est polynomiale de degré au plus m et l'on trouve la dimension de l'espace $\mathcal{H}_m$ des fonctions harmoniques de degré au plus m. Le problème se termine par une extension aux polynômes à n variables où l'on obtient la dimension de $\mathcal{H}_m$ après avoir dénombré (sans indication !) les partitions de l'entier n.

Cette épreuve étant trop longue pour le temps imparti, il était judicieux de traiter sans précipitation les parties I et II, en soignant la rédaction, puis de se focaliser sur une partie de la suite, après avoir lu entièrement l'énoncé pour comprendre sa logique, qui est très claire.

Indications

Partie I

I.A.1.a Trouver un carré $\mathrm{I} \times \mathrm{J}$ à placer dans une boule ouverte incluse dans Ω.

I.A.1.b Se ramener à des polynômes d'une seule variable sachant qu'un polynôme de $\mathbb{R}[\mathrm{X}]$ possédant une infinité de racines a tous ses coefficients nuls.

I.B.1 La définition des polynômes à deux variables donne une base évidente.

I.B.3.b Le laplacien abaisse de 2 le degré ou rend le polynôme nul.

I.C.2 Pour $(x, y) \in \mathscr{C}(0, 1)$, on a $x^2 + y^2 = 1$.

Partie II

II.B.1 Utiliser le théorème de Schwarz.

II.B.2 Reconnaître la composée d'une homothétie et d'une translation. Elle envoie toute boule ouverte sur une boule ouverte de $\mathbb{R}^2$.

II.B.3 Après avoir justifié le caractère $\mathscr{C}^2$, calculer les dérivées à l'aide de la règle de la chaîne.

II.C.1 Pour h_2, utiliser la question II.B.1.

II.C.2 Exprimer la fonction de l'énoncé à l'aide de $\partial_1 h_1$ et $\partial_2 h_1$.

II.D.1 Ne faire aucun calcul de dérivée !

II.D.3 Réduire au même dénominateur le membre de droite et comparer avec l'expression complexe de $\mathrm{N}(x, y, t)$.

II.D.4 Écrire $1/(1-z\mathrm{e}^{-\mathrm{i}t})$ comme la somme d'une série puis appliquer le théorème d'échange des signes somme et intégrale.

Partie III

III.A.1.a Sortir le facteur constant $1 - x^2 - y^2$ de l'intégrale avant de dériver sous le signe intégrale. Il est judicieux d'exprimer les dérivées d'ordre 1 et 2 au cours du calcul en fonction de $\widetilde{\mathrm{N}}(x, y, t) = 1/\big[(x - \cos t)^2 + (y - \cos t)^2\big]$.

III.A.2.a L'ensemble I_0^δ représente les paramètres t des points $\mathrm{M}_t = (\cos t, \sin t)$ du cercle $\mathrm{C}(0, 1)$ situés à distance au plus δ du point M_{t_0}.

III.A.2.b Écrire la définition avec ε de la continuité de f en un point (a_0,b_0) de $\mathrm{C}(0,1)$.

III.A.2.c La racine carrée du dénominateur de $\mathrm{N}(x, y, t)$ représente une distance. La minorer à l'aide de l'inégalité triangulaire.

III.A.2.d Fixer δ afin de majorer, grâce à la question III.A.2.c, l'intégrale de l'énoncé par $\mathrm{K}(1-x^2-y^2)$ avec K constante. Déterminer ensuite $\eta' > 0$ pour rendre ce majorant inférieur à $\varepsilon/2$.

III.A.3 Démontrer la continuité de u en $(\cos t_0, \sin t_0)$ à l'aide de la définition avec ε. Utiliser la relation de Chasles pour faire apparaître les intégrales des questions III.A.2.b et III.A.2.d.

III.B.1.a Écrire la formule de Taylor-Young à l'ordre 2 au voisinage de $\widetilde{x}$ pour la fonction $x \longmapsto u_n(x, \widetilde{y})$.

III.B.1.b Calculer $\Delta u_n(x, y)$.

III.B.2 Utiliser le théorème des bornes atteintes pour u sur $\overline{\mathrm{D}}(0, 1)$.

III.B.3 Montrer que u est négative puis qu'elle est positive en considérant $-u$.

Partie IV

IV.A.2 Montrer que ϕ_{m-2} est bijective.

IV.A.3 Il s'agit du polynôme harmonique de la question IV.A.2.

IV.A.4 Chercher un polynôme T de la forme $ax + by + c$ avec $a, b, c \in \mathbb{R}$.

IV.B.1 L'existence provient de la question IV.A.2. L'unicité s'obtient en considérant deux décompositions et en utilisant l'injectivité de ϕ_{m-2}.

IV.B.2 Reprendre la preuve de la question I.B.3.c en montrant que $\operatorname{Im} \Delta_m = \mathscr{P}_{m-2}$.

IV.B.3 Utiliser des polynômes déjà rencontrés et montrer qu'ils sont indépendants entre eux.

IV.C.1 Pour le calcul du cardinal, on se ramène à une situation plus facile à dénombrer : si $i_1 + i_2 + \cdots + i_n = m$, alors on considère le mot

$$\underbrace{\mathrm{A}\cdots\mathrm{A}}_{i_1 \text{ fois}}\mathrm{B}\underbrace{\mathrm{A}\cdots\mathrm{A}}_{i_2 \text{ fois}}\mathrm{B}\cdots\underbrace{\mathrm{A}\cdots\mathrm{A}}_{i_{n-1} \text{ fois}}\mathrm{B}\underbrace{\mathrm{A}\cdots\mathrm{A}}_{i_n \text{ fois}}$$

Remarquer enfin que $\mathscr{P}_m$ est une union disjointe d'ensembles de l'énoncé.

IV.C.2 Reprendre la preuve de la question IV.B.2.

I. Résultats préliminaires

I.A.1.a Soit $(x,y) \in \Omega$. L'ensemble Ω étant un ouvert de $\mathbb{R}^2$, il existe par définition un rayon $r > 0$ tel que le disque ouvert $\mathrm{D}((x,y),r)$ soit inclus dans Ω. Notons

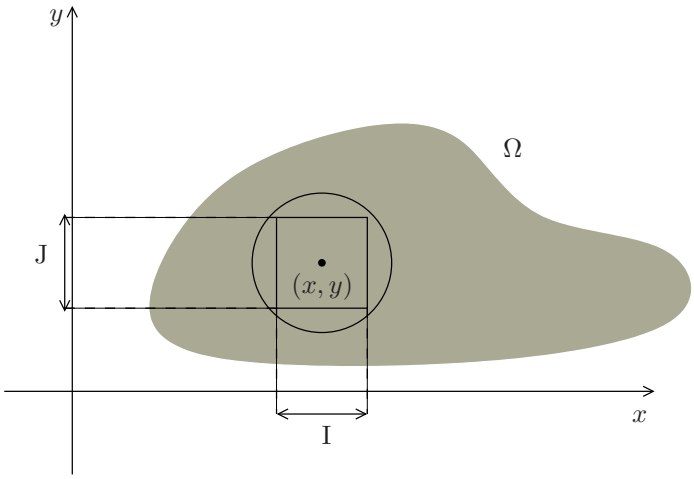
$$\mathrm{I} =]\, x - r/2\,;x + r/2\,[\qquad \text{et} \qquad \mathrm{J} =]\, y - r/2\,;y + r/2\,[$$

Pour tout $(x',y') \in \mathrm{I} \times \mathrm{J}$, la distance de (x',y') à (x,y) vaut

$$d((x,y),(x',y')) = \sqrt{(x'-x)^2 + (y'-y)^2} < \sqrt{2(r/2)^2} < r$$

donc $(x',y') \in \mathrm{D}((x,y),r)$. Finalement,

$$\boxed{\forall\,(x,y) \in \Omega \quad \exists\,\mathrm{I},\mathrm{J} \text{ intervalles ouverts de } \mathbb{R} \qquad (x,y) \in \mathrm{I} \times \mathrm{J} \subset \Omega}$$



I.A.1.b Soit $(x_0,y_0) \in \Omega$. D'après la question I.A.1.a, considérons I et J deux intervalles ouverts non vides tels que $(x_0,y_0) \in \mathrm{I} \times \mathrm{J} \subset \Omega$. Puisque P s'annule sur Ω, il s'annule sur $\mathrm{I} \times \mathrm{J}$ donc, pour tout $y \in \mathrm{J}$, le polynôme $\mathrm{P}_y = \mathrm{P}(\cdot,y)$ en x s'annule sur l'intervalle I :

$$\forall\, y \in \mathrm{J} \quad \forall\, x \in \mathrm{I} \qquad \mathrm{P}_y(x) = \mathrm{P}(x,y) = \sum_{k=0}^{n} \left(\sum_{\ell=0}^{n-\ell} \alpha_{k,\ell}\, y^\ell \right) x^k$$

Ainsi, P_y possède une infinité de racines donc c'est le polynôme nul. Par conséquent, tous ses coefficients sont nuls :

$$\forall\, y \in \mathrm{J} \quad \forall\, k \in [\![\, 0\,;\, n\,]\!] \qquad \sum_{\ell=0}^{n-\ell} \alpha_{k\ell}\, y^\ell = 0$$

Ce sont des polynômes en y s'annulant sur J donc ils sont identiquement nuls sur J. Ainsi, $\alpha_{k,\ell} = 0$ pour tous $1 \leqslant k,\ell \leqslant n$ donc P est le polynôme nul.

Si un polynôme P s'annule sur un ouvert non vide de $\mathbb{R}^2$, alors il est nul.

I.A.2 Soient $\mathrm{P}(x,y) = xy$ et $\Omega = \{0\} \times \mathbb{R}$. L'ensemble Ω est fermé par produit de fermés, il admet une infinité d'éléments et P s'annule sur Ω sans être le polynôme nul.

Le résultat de la question I.A.1 ne subsiste pas si Ω n'est pas supposé ouvert.

Centrale Informatique PSI 2015 — Corrigé

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

Ce sujet s'intéresse à une application en astronomie (effectivement utilisée par les astronomes) d'un algorithme d'intégration numérique, appelé schéma d'intégration de Verlet.

La première partie part du constat qu'en Python les listes ne sont pas directement adaptées pour faire du calcul vectoriel puisque leur addition à l'aide du symbole + conduit à la concaténation des deux listes (qui du coup peuvent être de tailles quelconques) plutôt qu'à l'addition terme à terme comme on pourrait s'y attendre si l'on représente des vecteurs par des listes. On construit donc quelques fonctions qui permettent de pallier cet inconvénient, comme l'addition ou la soustraction terme à terme ou encore la multiplication par un scalaire. Ces questions servent essentiellement à vérifier que le candidat maîtrise les boucles sur les listes.

La deuxième partie est une étude plutôt théorique des avantages comparés des schémas d'intégration d'Euler et de Verlet pour résoudre numériquement une équation différentielle[1]. L'accent est mis sur l'aspect mathématique, mais on fait aussi des liens avec le cours de mécanique sur les oscillateurs à un degré de liberté puisque l'on étudie principalement l'effet du schéma d'intégration sur un oscillateur harmonique et sa représentation graphique en terme de portrait de phase. Les implémentations du schéma d'Euler (prolongement naturel et direct du cours) et du schéma de Verlet (variation sur le thème d'Euler) constituent les applications informatiques de cette partie.

La troisième partie se concentre sur l'implémentation du schéma de Verlet dans le cadre classique du problème gravitationnel à N corps. Il s'agit dans un premier temps de coder le calcul de la force gravitationnelle d'un objet sur un autre, puis de tous les autres objets sur l'objet d'étude pour finalement intégrer les équations du mouvement à l'aide du schéma de Verlet. On termine sur une estimation expérimentale et théorique de la complexité de l'algorithme proposé.

La quatrième partie aborde des questions sur les bases de données dans l'idée de collecter une série de corps issus du système solaire pour démarrer une simulation à grande échelle. Cela commence par des requêtes SQL très simples, puis la difficulté augmente progressivement. Le problème se termine par l'écriture de la fonction simulant le système solaire, en partant des conditions initiales collectées via la base de données. Elle utilise les fonctions de la partie III.

L'épreuve est progressive, même si elle démarre avec des questions légèrement hors programme. Elle construit presque complètement un intégrateur à N corps qui pourra servir de base, par exemple, à de futurs TIPE sur ce thème. La longueur est raisonnable et le sujet semble parfaitement correspondre à ce qu'on est en droit d'attendre sur le programme d'informatique commune.

[1] Sans surprise, c'est Verlet qui va gagner.

Indications

Partie I

I.A.1 Attention, les listes Python ne se comportent pas naturellement comme des vecteurs en mathématiques. Dans le doute, tester le code dans une session interactive de Python.

Partie II

II.A.1 Par définition, on a $y' = z$, il reste donc à trouver à quoi relier z'.

II.A.2 Il suffit de partir de la constatation que $\displaystyle\int_{t_i}^{t_{i+1}} y'(t)\,\mathrm{d}t = y(t_{i+1}) - y(t_i)$.

II.B.2 Utiliser la fonction `f` définissant l'équation différentielle, les conditions initiales `y0` et `z0` ainsi que le pas de temps `h` et le nombre `n` de points.

II.B.3.a Penser au cours de physique : multiplier par la vitesse y' pour faire apparaître des dérivées de fonctions composées connues.

II.B.3.b Ne pas oublier qu'on se restreint ici au cas de l'oscillateur harmonique.

II.B.3.c Un schéma satisfait la conservation de l'énergie si... l'énergie se conserve.

II.B.3.d et e Là encore, il faut s'inspirer du portrait de phase de l'oscillateur harmonique vu dans le cours de physique.

II.C.2.a C'est la question la plus calculatoire du problème. Commencer par exprimer y_{i+1} et z_{i+1} en fonction uniquement de y_i et z_i sans oublier qu'on s'intéresse toujours uniquement à l'oscillateur harmonique. Par la suite, utiliser l'identité remarquable $a^2 - b^2 = (a-b)(a+b)$ et faire un développement en $\mathrm{O}(h^3)$ pour ne pas s'encombrer de termes inutiles.

Partie III

III.A.2 Définir une fonction auxiliaire `norme(vecteur)` pour simplifier l'implémentation.

III.B.2 Utiliser la méthode de Verlet pour chaque coordonnée en position de chaque corps après avoir fait apparaître l'équation différentielle par application de la relation fondamentale de la dynamique. Attention, la fonction f dépend de la position de tous les corps et pas seulement de la coordonnée considérée.

III.B.3 Il reste les vitesses à faire évoluer. Utiliser à nouveau la méthode de Verlet sur chaque composante de chaque vitesse à l'aide des positions présentes et à venir. Noter qu'il n'est pas nécessaire de calculer les positions suivantes pour chaque étape de la boucle sur j. Un seul calcul en dehors de la boucle est suffisant.

III.B.5.a Il est possible de trouver une complexité cubique dans le cas où vous n'avez pas tenu compte de l'indication précédente.

Partie IV

IV.B.1 Le mot-clé `DISTINCT` utilisé en conjonction avec `COUNT` permet d'éviter les doublons.

IV.B.2 Pour un corps donné, la date du dernier état antérieur à `tmin()` est le `MAX` de toutes les dates des états antérieurs à `tmin()`.

IV.B.3 Il y a des informations à prendre sur trois tables donc deux jointures à effectuer. La distance attendue pour l'ordonnancement est euclidienne.

I. Quelques fonctions utilitaires

I.A.1 Le + appliqué sur deux listes a pour action de les concaténer l'une à l'autre en créant une troisième liste. Il ne faut donc pas confondre avec l'addition terme à terme comme on peut l'imaginer lors de la sommation de deux vecteurs.

```
>>> [1,2,3] + [4,5,6]
[1, 2, 3, 4, 5, 6]
```

L'idée de cette partie est de montrer que les listes Python ne se comportent pas naturellement comme des vecteurs mathématiques, c'est-à-dire que la « somme » de deux listes ne donne pas une somme composante par composante des deux listes (ici on attendrait `[5, 7, 9]`) mais les concatène. Il existe bien sûr des objets Python qui permettent de faire exactement cela, ce sont les `numpy.array` qui se comportent « comme attendu » vis-à-vis de l'addition :

```
>>> import numpy
>>> a = numpy.array([1,2,3])
>>> b = numpy.array([4,5,6])
>>> a+b
array([5, 7, 9])
```

Attention, il ne faut *jamais* utiliser la concaténation pour construire une liste élément par élément car celle-ci, en Python, renvoie une copie des deux listes. Par conséquent, la construction suivante d'une liste contenant les n premiers entiers pairs est de complexité *quadratique* en n.

```
L= []
for i in range(n):
    L= L + [2*i] # Ne JAMAIS faire cela, préférer L.append(2*i)
```

En effet, chaque concaténation est linéaire en i, le nombre d'éléments de la liste `L` à l'étape i, de sorte que le nombre total d'opérations est de l'ordre de $\sum_{i=1}^{N} i \approx \frac{N^2}{2}$. Alors certes, cela ne sera pas trop handicapant quand on fabrique une liste d'une dizaine d'éléments, mais déjà pour mille, cela se sentira et ce sera encore bien pire pour un million !

I.A.2 Pour les entiers naturels, $n \times x$ vaut $x + \cdots + x$ où x apparaît n fois. Suivant la même sémantique, la multiplication d'un entier avec une liste revient à concaténer la liste avec elle-même autant de fois que demandé.

```
>>> 2*[1,2,3]
[1, 2, 3, 1, 2, 3]
```

Le résultat « naturel » attendu (soit `[2, 4, 6]`) serait donné par l'usage d'un tableau Numpy (essayez `2*numpy.array([1,2,3])`).

Ni la concaténation de deux listes, ni la multiplication d'une liste par un entier ne sont à proprement parler au programme d'informatique pour tous, mais il est probable que vous croisiez cette syntaxe pour définir une liste initialisée à zéro sous la forme `L = [0]*n`.

> Attention, il ne faut pas utiliser cette construction pour initialiser une matrice de zéros car Python ne fait pas une « copie profonde » des objets concernés. Observez plutôt :
>
> ```
> >>> a = [[0]*3]*2     # Création de la "matrice"
> >>> a[0][1] = 1       # Modification de la première ligne
> >>> a                 # Hein ?!?
> [[0, 1, 0], [0, 1, 0]]
> ```

I.B Présentons trois versions possibles, l'une en créant une liste remplie de zéros, l'autre en itérant sur les éléments et en construisant la liste au fur et à mesure par ajouts successifs, la dernière en utilisant la Pythonnerie de construction de liste en compréhension.

```
def smul(nombre,liste):
    """ Multiplication terme à terme d'une liste par un nombre."""
    L = [0]*len(liste)           # Initialisation à une liste de 0
    for i in range(len(liste)):  # Autant de fois que d'éléments
        L[i] = nombre * liste[i] # On remplit avec la valeur idoine
    return L                     # On n'oublie pas de renvoyer L

def smul(nombre,liste):
    L = []                       # Initialisation à une liste vide
    for element in liste:        # On itère sur les éléments
        L.append(nombre*element) # On rajoute la valeur idoine
    return L                     # On n'oublie pas de renvoyer L

def smul(nombre,liste):
    return [nombre*element for element in liste] # One-liner !
```

> Bien sûr, le jour du concours, une seule version est nécessaire (et suffisante !), mais il est intéressant de comparer les diverses manières permettant d'arriver au résultat. La version la plus proche de ce que tout le monde doit pouvoir faire au vu du programme officiel est la seconde. Néanmoins, les correcteurs comprendront les différents dialectes. Le rapport de concours précise même que « De nombreux candidats résolvent cette partie à l'aide de liste en compréhension, qui produisent du code concis et lisible. »

I.C.1 L'addition terme à terme sur les listes se définit par

```
def vsom(L1,L2):
    """ Fait l'addition vectorielle L1+L2 de deux listes.
    Les deux listes doivent avoir la même taille. """
    L = [0] * len(L1)        # Inialisation à une liste de 0
    for i in range(len(L1)): # On regarde toutes les positions
        L[i] = L1[i] + L2[i] # Addition à la position i
    return L                 # Renvoi du résultat
```

> À noter qu'ici on ne peut pas itérer sur les éléments car on a besoin de ceux de chaque liste, ce qui impose de passer par les indices des positions, communes aux deux listes.

Mines Maths 1 PSI 2015 — Corrigé

Ce corrigé est proposé par Émilie Liboz (Professeur en CPGE) ; il a été relu par Florence Monna (Docteur en mathématiques) et Benjamin Monmege (Enseignant-chercheur à l'université).

Le sujet introduit une distance entre deux lois de probabilités à valeurs dans $\mathbb{N}$, connue sous le nom de « distance en variation totale » dans les livres. L'objectif du problème est de majorer la distance entre une loi de Poisson et la loi d'une somme S de variables aléatoires indépendantes de Bernoulli. On suit pour cela la méthode dite de Stein, qui permet par ailleurs de généraliser l'interprétation de la loi de Poisson comme une loi des événements rares.

- La première partie établit des résultats préliminaires sur des séries convergentes et calcule la distance entre deux lois de Bernoulli.
- La deuxième partie caractérise l'ensemble des suites de réels $(p_n^{(\lambda)})_{n\in\mathbb{N}}$ définissant une loi de Poisson $\mathscr{P}(\lambda)$.
- La troisième partie est consacrée à la résolution de l'équation de Stein

$$\forall\, n \in \mathbb{N} \qquad \lambda\, f(n+1) - n\, f(n) = \widetilde{h}(n)$$

  où le second membre fait intervenir les $p_n^{(\lambda)}$. On démontre que les suites f solutions sont bornées.
- La quatrième partie démontre la propriété de Lipschitz pour les fonctions solutions f précédentes : elle donne un majorant de l'ensemble des variations $|f(n+1) - f(n)|$ de f pour $n \in \mathbb{N}$. Les notions de borne inférieure et de borne supérieure interviennent à plusieurs reprises.
- Enfin, la cinquième partie applique les résultats précédents pour majorer la distance entre la loi de Poisson $\mathscr{P}(\lambda)$ et la loi de S. Elle fait appel à des outils de base sur les variables aléatoires finies : espérance, indépendance, théorème de transfert.

La majorité des questions portent sur les séries numériques : étude de la convergence, calculs sur les sommes de séries convergentes. Les probabilités n'apparaissent que dans la dernière partie. L'ensemble est très calculatoire et assez long. À la difficulté technique s'ajoute la forte dépendance des questions et des parties entre elles. Ce sujet permet à la fois de faire le point sur les séries numériques et de s'entraîner à suivre le fil d'un énoncé, c'est-à-dire à comprendre où il nous emmène et par quel chemin.

Indications

Partie I

3 Majorer f afin de comparer le terme général de cette série à celui d'une série convergente.

Partie II

6 Considérer $f = \mathbf{1}_{\{n_0\}}$ pour $n_0 \in \mathbb{N}$.

Partie III

7 Raisonner par analyse-synthèse. Utiliser la formule (**3**) de l'énoncé pour construire une infinité d'éléments de $\mathcal{S}_h$, en remarquant que la condition ne concerne que les $n > 0$.

8 Calculer $\displaystyle\sum_{k=0}^{+\infty} \widetilde{h}(k)\frac{\lambda^k}{k!}$.

9 Utiliser la formule de la question 8 en majorant $\widetilde{h}$.

Partie IV

10 Utiliser la formule de la question 7.

11 Exploiter le résultat de la question 8.

12 Utiliser les formules des questions 10 et 11.

13 Attention, la formule donnée pour $\Delta f_0(0)$ est fausse.

14 Distinguer les cas $n \neq m$ et $n = m$ pour utiliser les questions 12 et 13.

15 Comparer $\widetilde{h}$ et $\widetilde{h_+}$ (les notations utilisées sont celles définies dans l'énoncé au début de la partie III).

16 Majorer h_+ afin de comparer le terme général de cette série à celui d'une série convergente en utilisant la formule de la question 10.

17 Utiliser le fait que $f_m \in \mathcal{S}_{\mathbf{1}_{\{m\}}}$.

18 Remarquer que, ici, f est une fonction quelconque de $\mathcal{S}_h$ et chercher à la relier à la fonction de la question 17.

Partie V

19 Distinguer les cas $\mathrm{X}_k = 0$ et $\mathrm{X}_k = 1$ pour la première égalité. Utiliser l'indépendance des X_k pour la deuxième.

20 Remplacer λ par $\displaystyle\sum_{k=1}^{n} r_k$ et S par $\displaystyle\sum_{k=1}^{n} \mathrm{X}_k$ dans le membre de gauche et utiliser les formules obtenues à la question 19.

21 Appliquer le théorème du transfert et utiliser le fait que $f_{\mathrm{A}} \in \mathcal{S}_{\mathbf{1}_{\mathrm{A}}}$ pour calculer $\mathrm{E}\left(\lambda f_{\mathrm{A}}(\mathrm{S}+1) - \mathrm{S} f_{\mathrm{A}}(\mathrm{S})\right)$ puis prendre la borne supérieure sur $\mathrm{A} \subset \mathbb{N}$.

22 Appliquer les résultats des questions 20 et 21 pour exprimer $\mathrm{dist}(\mathrm{loi}(\mathrm{S}), \mathrm{P}_\lambda)$ en fonction de f_{A} puis utiliser l'inégalité (**5**).

I. Préliminaires

1 La série numérique $\sum \dfrac{\lambda^n}{n!}$ est convergente de somme e^{λ}. La suite $\left(\mathrm{e}^{-\lambda}\dfrac{\lambda^n}{n!}\right)_{n\geqslant 0}$ appartient donc à $\mathcal{P}$ car pour tout $n \geqslant 0$, $\mathrm{e}^{-\lambda}\dfrac{\lambda^n}{n!} \geqslant 0$. Par conséquent,

$$\boxed{\text{Pour } c = \mathrm{e}^{-\lambda}, \text{ la suite } \left(c\frac{\lambda^n}{n!}\right)_{n\geqslant 0} \text{ appartient à } \mathcal{P}.}$$

2 Pour p et $q \in [\,0\,;1\,]$, posons $p_0 = 1-p$, $p_1 = p$, $q_0 = 1-q$, $q_1 = q$ et, pour tout $n \geqslant 2$, $p_n = q_n = 0$. Alors, pour $\mathrm{A} \subset \mathbb{N}$:

$$\sum_{n\in\mathrm{A}} p_n - \sum_{n\in\mathrm{A}} q_n = \begin{cases} 0 & \text{si} \quad \{0,1\} \subset {}^{\mathrm{c}}\mathrm{A} \\ q-p & \text{si} \quad 0\in\mathrm{A},\ 1\notin\mathrm{A} \\ p-q & \text{si} \quad 1\in\mathrm{A},\ 0\notin\mathrm{A} \\ 0 & \text{si} \quad \{0,1\} \subset \mathrm{A} \end{cases}$$

Ainsi, $$\sup_{\mathrm{A}\subset\mathbb{N}} \left| \sum_{n\in\mathrm{A}} p_n - \sum_{n\in\mathrm{A}} q_n \right| = |p-q|$$

soit $$\boxed{\operatorname{dist}\Big((1-p,p,0,\dots),(1-q,q,0,\dots)\Big) = |p-q|}$$

3 Puisque $f \in \mathcal{F}$, il existe $\mathrm{M} > 0$ tel que pour tout $n \in \mathbb{N}$, $|f(n)| \leqslant \mathrm{M}$. De plus, si $\mathrm{P} \in \mathcal{P}$, alors pour tout entier naturel n, on a $p_n \geqslant 0$ de sorte que $|f(n)p_n| \leqslant \mathrm{M}p_n$. Or $\mathrm{M}p_n$ est le terme général d'une série convergente. Par comparaison de séries à termes positifs, on en déduit que la série numérique $\sum f(n)p_n$ est absolument convergente et par conséquent

$$\boxed{\text{La série numérique } \sum f(n)p_n \text{ est convergente.}}$$

II. Caractérisation

4 Soit $\mathrm{M} > 0$ tel que pour tout $n \in \mathbb{N}$, $|f(n)| \leqslant \mathrm{M}$. Ainsi, pour tout entier $n \geqslant 1$:

$$|nf(n)p_n^{(\lambda)}| \leqslant \mathrm{M}\mathrm{e}^{-\lambda} n\frac{\lambda^n}{n!} = \mathrm{M}\lambda\mathrm{e}^{-\lambda}\frac{\lambda^{n-1}}{(n-1)!}$$

Or le membre de droite est le terme d'une série convergente (on reconnaît le terme général de la série exponentielle). Par comparaison de séries à termes positifs, on en déduit que la série numérique $\sum nf(n)p_n^{(\lambda)}$ est absolument convergente, d'où

$$\boxed{\text{La série numérique } \sum nf(n)p_n^{(\lambda)} \text{ est convergente.}}$$

5 Tout d'abord, d'après la question 4, la série $\sum nf(n)p_n^{(\lambda)}$ est convergente. Ensuite, la suite P_λ est un élément de $\mathcal{P}$ d'après la question 1 et donc, la fonction f étant bornée, par le même raisonnement que celui de la question 3, on montre que la série

$\sum f(n+1)p_n^{(\lambda)}$ est convergente. Les sommes $\sum_{n=0}^{+\infty} f(n+1)p_n^{(\lambda)}$ et $\sum_{n=0}^{+\infty} nf(n)p_n^{(\lambda)}$ sont alors bien définies. De plus, à l'aide du changement d'indice $n = k+1$, on obtient

$$\sum_{n=0}^{+\infty} nf(n)p_n^{(\lambda)} = \sum_{n=1}^{+\infty} nf(n)\mathrm{e}^{-\lambda}\frac{\lambda^n}{n!} = \sum_{k=0}^{+\infty}(k+1)f(k+1)\mathrm{e}^{-\lambda}\frac{\lambda^{k+1}}{(k+1)!}$$

soit
$$\sum_{n=0}^{+\infty} nf(n)p_n^{(\lambda)} = \lambda\sum_{k=0}^{+\infty} f(k+1)\mathrm{e}^{-\lambda}\frac{\lambda^k}{k!}$$

En réindexant la deuxième somme par n au lieu de k, on peut conclure que

$$\boxed{\lambda\sum_{n=0}^{+\infty} f(n+1)p_n^{(\lambda)} = \sum_{n=0}^{+\infty} nf(n)p_n^{(\lambda)}}$$

6 Soit $\mathrm{Q} = (q_n)_{n\geqslant 0} \in \mathcal{P}$ tel que, pour tout $f \in \mathcal{F}$, on ait

$$\lambda\sum_{n=0}^{+\infty} f(n+1)q_n = \sum_{n=0}^{+\infty} nf(n)q_n$$

En sélectionnant $f = 1_{\{n_0\}}$ pour $n_0 \in \mathbb{N}^*$ (la fonction indicatrice définie dans l'introduction du sujet), l'identité vérifiée par Q devient :

$$\lambda\sum_{n=0}^{+\infty} \mathbf{1}_{\{n_0\}}(n+1)q_n = \sum_{n=0}^{+\infty} n\mathbf{1}_{\{n_0\}}(n)q_n$$

Or, $\mathbf{1}_{\{n_0\}}(n+1) \neq 0$ si et seulement si $n = n_0 - 1$ donc cette égalité est équivalente à $\lambda q_{n_0-1} = n_0 q_{n_0}$. La suite $(q_n)_{n\geqslant 0}$ vérifie alors

$$\forall\, n \geqslant 0 \quad q_{n+1} = \frac{\lambda}{n+1}q_n$$

Montrons par récurrence sur n que la propriété

$$\mathscr{P}(n) : \quad q_n = q_0\frac{\lambda^n}{n!}$$

est vraie pour tout entier n.

- $\underline{\mathscr{P}(0)}$ est évidemment vraie.
- $\underline{\mathscr{P}(n) \Longrightarrow \mathscr{P}(n+1)}$: si $q_n = q_0\lambda^n/n!$ pour un n donné, alors

  $$q_{n+1} = \frac{\lambda}{n+1}q_n = \frac{\lambda}{n+1}\frac{\lambda^n}{n!}q_0 = \frac{\lambda^{n+1}}{(n+1)!}q_0$$

  ce qui montre que la propriété est vraie au rang $n+1$.
- Conclusion : $\quad \forall n \in \mathbb{N} \quad q_n = q_0\dfrac{\lambda^n}{n!}$

De plus, la suite Q est un élément de $\mathcal{P}$ donc, d'après la question 1, $q_0 = \mathrm{e}^{-\lambda}$ ce qui implique que

$$\boxed{\text{Les suites Q et } \mathrm{P}_\lambda \text{ sont égales.}}$$

Mines Maths 2 PSI 2015 — Corrigé

Ce corrigé est proposé par Matthias Moreno (ENS Lyon) ; il a été relu par Céline Chevalier (Enseignant-chercheur à l'université) et Guillaume Batog (Professeur en CPGE).

Le groupe orthogonal réel O_n est l'ensemble des matrices carrées M de taille n vérifiant ${}^t\mathrm{M}\,\mathrm{M} = \mathrm{I}_n$. En remplaçant cette condition par ${}^t\mathrm{M}\cdot\mathrm{J}\cdot\mathrm{M} = \mathrm{J}$ où (lorsque n est pair)

$$\mathrm{J} = \begin{pmatrix} 0_{n/2} & -\mathrm{I}_{n/2} \\ \mathrm{I}_{n/2} & 0_{n/2} \end{pmatrix}$$

les matrices M forment le groupe symplectique réel, noté $\mathcal{S}p_n$. L'étude de ce groupe est l'objet de cette épreuve.

Les parties II et III sont indépendantes, mais chacune utilise l'ensemble des résultats de la partie I, notamment celui de la question 8.

- La partie I montre que le produit de deux matrices de $\mathcal{S}p_{2n}$, l'inverse, et la transposée d'une matrice de $\mathcal{S}p_{2n}$ sont aussi des matrices de $\mathcal{S}p_{2n}$. On examine quelques éléments de ce groupe et on termine par une description générale par blocs des matrices de $\mathcal{S}p_{2n}$. Hormis la question 7, où l'on peut tourner en rond, cette partie ne présente pas de difficultés particulières.
- La partie II montre que l'ensemble des matrices de $\mathcal{S}p_{2n}$ qui commutent avec toutes les autres est égal à $\{-\mathrm{I}_{2n}, \mathrm{I}_{2n}\}$. Si l'on pense à réutiliser les exemples vus précédemment, cette partie est encore relativement facile, mis à part la dernière question où l'on démontre le résultat classique suivant : une matrice commute avec toutes les matrices de $\mathrm{GL}_n(\mathbb{R})$ si, et seulement si, elle est de la forme $\lambda\mathrm{I}_n$ avec $\lambda \in \mathbb{R}$.
- La partie III montre que toutes les matrices de $\mathcal{S}p_{2n}$ ont pour déterminant 1. On étudie d'abord un cas particulier, en montrant que la matrice s'écrit comme le produit d'une matrice triangulaire supérieure par une matrice triangulaire inférieure (par blocs). Ensuite, l'examen des vecteurs propres d'une matrice symétrique permet de ramener le cas général au cas particulier précédent.

Cette épreuve est élémentaire, la plupart des questions se résolvent par un simple calcul. Elle permet de travailler sur les différentes opérations matricielles (produit par blocs, transposée, inverse, déterminant). Dans la dernière partie, on aborde quelques questions d'algèbre bilinéaire (produit scalaire, matrice symétrique, vecteurs propres orthogonaux).

Indications

Partie I

1 Utiliser la formule de calcul du produit de matrices par blocs rappelée dans l'énoncé. Déduire directement l'inverse de J du calcul de J^2.

2 Utiliser les résultats de la question 1 pour J.

4 Prendre le déterminant dans l'équation qui définit l'appartenance à $\mathcal{S}p_{2n}$.

6 La question 4 donne directement l'inversibilité.

7 Commencer par prendre l'inverse dans l'égalité définissant $\mathcal{S}p_{2n}$.

8 Faire les calculs par blocs, puis identifier les blocs.

Partie II

10 Reconnaître la matrice de la question 2 pour $\alpha = -1$. Écrire que M commute avec L et ${}^tL$. Une fois M de la forme voulue, l'examen de son déterminant montre que A est inversible.

11 Se reporter à la question 3, puis écrire que M commute avec L_U.

12 L'indication de l'énoncé permet de montrer que A est de la forme λI_n avec $\lambda \in \mathbb{R}$. On conclut avec la question 4.

Partie III

13 Raisonner par analyse-synthèse, en effectuant des calculs par blocs. Penser à utiliser l'hypothèse que D est inversible.

14 La question 8 permet de montrer que BD^{-1} est symétrique. Appliquer ensuite le déterminant à la formule de la question 13 et utiliser à nouveau la question 8 pour simplifier le résultat.

15 Le produit scalaire se calcule matriciellement. En utilisant sa symétrie, on fait apparaître une première fois s_1, puis s_2.

16 La question 8 permet de conclure directement.

17 Pour la liberté, montrer que les vecteurs sont deux à deux orthogonaux.

18 Si $D - \alpha B$ n'est pas inversible, alors son noyau possède un vecteur non nul.

19 Multiplier M par $K(\alpha)$ permet de se ramener au cas de la question 14. Conclure grâce à l'égalité $\det K(\alpha) = 1$.

I. Le groupe symplectique

1 Tout d'abord, utilisons la formule de calcul par blocs donnée dans l'énoncé :

$$\mathrm{J}^2 = \begin{pmatrix} 0_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix}\begin{pmatrix} 0_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix} = \begin{pmatrix} 0_n\cdot 0_n - \mathrm{I}_n\cdot \mathrm{I}_n & 0_n\cdot(-\mathrm{I}_n) - \mathrm{I}_n\cdot 0_n \\ \mathrm{I}_n\cdot 0_n + 0_n\cdot \mathrm{I}_n & \mathrm{I}_n\cdot(-\mathrm{I}_n) + 0_n\cdot 0_n \end{pmatrix}$$

Ainsi, $\boxed{\mathrm{J}^2 = \begin{pmatrix} -\mathrm{I}_n & 0_n \\ 0_n & -\mathrm{I}_n \end{pmatrix} = -\mathrm{I}_{2n}}$

Ensuite, $${}^t\mathrm{J} = {}^t\begin{pmatrix} 0_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix} = \begin{pmatrix} {}^t 0_n & {}^t\mathrm{I}_n \\ {}^t(-\mathrm{I}_n) & {}^t 0_n \end{pmatrix} = \begin{pmatrix} 0_n & \mathrm{I}_n \\ -\mathrm{I}_n & 0_n \end{pmatrix}$$

d'où $\boxed{{}^t\mathrm{J} = -\mathrm{J}}$

> Attention, lorsque l'on écrit la transposée d'une matrice par blocs, il faut penser à échanger les blocs anti-diagonaux !

Enfin, l'égalité $\mathrm{J}^2 = -\mathrm{I}_{2n}$ se réécrit $\mathrm{J}\cdot(-\mathrm{J}) = \mathrm{I}_{2n}$ donc

$\boxed{\mathrm{J} \text{ est inversible et } \mathrm{J}^{-1} = -\mathrm{J}.}$

2 On a $\mathrm{J} \in \mathscr{M}_{2n}$. D'après la question 1, ${}^t\mathrm{J}\cdot\mathrm{J}\cdot\mathrm{J} = -\mathrm{J}\cdot\mathrm{J}^2 = -\mathrm{J}\times(-\mathrm{I}_{2n}) = \mathrm{J}$ d'où

$\boxed{\mathrm{J} \in \mathcal{S}p_{2n}}$

Soit $\alpha \in \mathbb{R}$, on a ${}^t\mathrm{K}(\alpha) = \begin{pmatrix} \mathrm{I}_n & -\alpha\mathrm{I}_n \\ 0_n & \mathrm{I}_n \end{pmatrix}$. Calculons

$$\underbrace{\begin{pmatrix} \mathrm{I}_n & -\alpha\mathrm{I}_n \\ 0_n & \mathrm{I}_n \end{pmatrix}}_{{}^t\mathrm{K}(\alpha)} \underbrace{\begin{pmatrix} -\alpha\mathrm{I}_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix}}_{{}^t\mathrm{K}(\alpha)\cdot\mathrm{J}} \overbrace{\begin{pmatrix} 0_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix}}^{\mathrm{J}} \overbrace{\begin{pmatrix} \mathrm{I}_n & 0_n \\ -\alpha\mathrm{I}_n & \mathrm{I}_n \end{pmatrix}}^{\mathrm{K}(\alpha)} = {}^t\mathrm{K}(\alpha)\cdot\mathrm{J}\cdot\mathrm{K}(\alpha)$$

donc $\boxed{\forall \alpha \in \mathbb{R} \qquad \mathrm{K}(\alpha) \in \mathcal{S}p_{2n}}$

3 Soit $\mathrm{U} \in \mathcal{G}_n$, on a bien $\mathrm{L}_\mathrm{U} \in \mathscr{M}_{2n}$ puis ${}^t\mathrm{L}_\mathrm{U} = \begin{pmatrix} {}^t\mathrm{U} & 0_n \\ 0_n & \mathrm{U}^{-1} \end{pmatrix}$ d'où

$$\underbrace{\begin{pmatrix} {}^t\mathrm{U} & 0_n \\ 0_n & \mathrm{U}^{-1} \end{pmatrix}}_{{}^t\mathrm{L}_\mathrm{U}} \underbrace{\begin{pmatrix} 0_n & -{}^t\mathrm{U} \\ \mathrm{U}^{-1} & 0_n \end{pmatrix}}_{{}^t\mathrm{L}_\mathrm{U}\cdot\mathrm{J}} \overbrace{\begin{pmatrix} 0_n & -\mathrm{I}_n \\ \mathrm{I}_n & 0_n \end{pmatrix}}^{\mathrm{J}} \overbrace{\begin{pmatrix} \mathrm{U} & 0_n \\ 0_n & {}^t\mathrm{U}^{-1} \end{pmatrix}}^{\mathrm{L}_\mathrm{U}} = {}^t\mathrm{L}_\mathrm{U}\cdot\mathrm{J}\cdot\mathrm{L}_\mathrm{U}$$

car ${}^t\mathrm{U}\,{}^t\mathrm{U}^{-1} = {}^t(\mathrm{U}^{-1}\mathrm{U}) = {}^t\mathrm{I}_n = \mathrm{I}_n$. Finalement,

$\boxed{\forall\, \mathrm{U} \in \mathcal{G}_n \qquad \mathrm{L}_\mathrm{U} \in \mathcal{S}p_{2n}}$

> La notation ${}^t\mathrm{U}^{-1}$ n'est pas ambiguë car pour toute matrice U inversible, la matrice ${}^t\mathrm{U}$ est inversible et ${}^t(\mathrm{U}^{-1}) = ({}^t\mathrm{U})^{-1}$ d'après le cours.

4 Soit $M \in \mathcal{S}p_{2n}$. Alors ${}^tM\,JM = J$. D'après les propriétés du déterminant, on a

$$\det J = \det({}^tM\,JM) = \underbrace{\det({}^tM)}_{=\det M} \times \det J \times \det M = \det J \times (\det M)^2$$

Puisque J est inversible (question 1), $\det J$ est non nul d'où $(\det M)^2 = 1$. Ainsi,

$$\boxed{\text{Si } M \in \mathcal{S}p_{2n} \text{ alors } \det M = \pm 1.}$$

5 Soit $(M, N) \in (\mathcal{S}p_{2n})^2$. Alors $MN \in \mathcal{S}p_{2n}$. Calculons

$$\begin{aligned} {}^t(MN)\,J(MN) &= {}^tN({}^tM\,JM)N \\ &= {}^tN\,JN && (\text{car } M \in \mathcal{S}p_{2n}) \\ {}^t(MN)\,J(MN) &= J && (\text{car } N \in \mathcal{S}p_{2n}) \end{aligned}$$

donc $$\boxed{\forall (M, N) \in (\mathcal{S}p_{2n})^2 \qquad MN \in \mathcal{S}p_{2n}}$$

6 Soit $M \in \mathcal{S}p_{2n}$. D'après la question 4, M est inversible car de déterminant non nul. Multiplions l'égalité ${}^tM\,JM = J$ à gauche par $({}^tM)^{-1}$ et à droite par M^{-1}. Il vient

$$J = ({}^tM)^{-1}JM^{-1} = {}^t(M^{-1})\,JM^{-1}$$

d'où $$\boxed{\forall M \in \mathcal{S}p_{2n} \qquad M \in \mathcal{G}_{2n} \quad \text{et} \quad M^{-1} \in \mathcal{S}p_{2n}}$$

7 Soit $M \in \mathcal{S}p_{2n}$. D'après la question 6, $M^{-1} \in \mathcal{S}p_{2n}$ d'où ${}^t(M^{-1})\,JM^{-1} = J$. Prenons l'inverse :

$$({}^tM^{-1} \times J \times M^{-1})^{-1} = J^{-1}$$

Comme $(AB)^{-1} = B^{-1}A^{-1}$ pour toutes matrices $A, B \in \mathcal{G}_{2n}$, on obtient

$$\begin{aligned} (M^{-1})^{-1} \times J^{-1} \times ({}^tM^{-1})^{-1} &= J^{-1} \\ M \times (-J) \times {}^tM &= -J \quad (J^{-1} = -J, \text{ question 1}) \\ MJ\,{}^tM &= J \end{aligned}$$

soit $$ {}^t({}^tM)\,J\,{}^tM = J$$

Ainsi, $$\boxed{\forall M \in \mathcal{S}p_{2n} \qquad {}^tM \in \mathcal{S}p_{2n}}$$

> Si l'on prend la transposée dans ${}^tM\,JM = J$, on retombe sur la même égalité. En effet, la transposée échange l'ordre des matrices et elle transforme également M en ${}^tM$! En prenant l'inverse, on échange l'ordre des matrices sans transformer M en ${}^tM$, ce qui fait passer ${}^tM$ du bon côté.

8 On a $$ {}^tM = \begin{pmatrix} {}^tA & {}^tC \\ {}^tB & {}^tD \end{pmatrix}$$

d'où $$ {}^tM\,J = \begin{pmatrix} {}^tA & {}^tC \\ {}^tB & {}^tD \end{pmatrix} \begin{pmatrix} 0_n & -I_n \\ I_n & 0_n \end{pmatrix} = \begin{pmatrix} {}^tC & -{}^tA \\ {}^tD & -{}^tB \end{pmatrix}$$

puis $$ {}^tM\,JM = \begin{pmatrix} {}^tC & -{}^tA \\ {}^tD & -{}^tB \end{pmatrix} \begin{pmatrix} A & B \\ C & D \end{pmatrix} = \begin{pmatrix} {}^tC\,A - {}^tA\,C & {}^tC\,B - {}^tA\,D \\ {}^tD\,A - {}^tB\,C & {}^tD\,B - {}^tB\,D \end{pmatrix}$$

Mines Informatique PSI 2015 — Corrigé

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

Le sujet porte sur différents aspects de tests de validation d'une imprimante. Les parties sont indépendantes.

- La première partie porte sur la réception des données issues de la carte d'acquisition, en se concentrant plus particulièrement sur la trame contenant les informations.
- La deuxième, très courte et proche du cours, met en place des programmes de calcul approché d'intégrales, qui sont utilisés pour obtenir la moyenne et l'écart-type des données physiques reçues dans les trames.
- La troisième partie s'intéresse à la gestion d'un parc d'imprimantes en cours de validation à l'aide de bases de données.
- La quatrième aborde le thème de la compression de données, principalement en étudiant un long programme fourni en annexe.
- Enfin, la cinquième partie examine un schéma numérique de résolution d'équations différentielles afin de définir un test de validation des moteurs utilisés dans l'imprimante.

Ce sujet est intéressant sur le principe mais il est décevant : beaucoup de notions sont hors programme et ne sont pas définies ; le programme proposé dans la quatrième partie ne marche pas, ce qui conduit à des questions n'ayant pas vraiment de sens pour un candidat rigoureux ; les formulations des questions sont parfois très imprécises et on ne sait pas réellement ce qui est demandé... Bref, cet énoncé a tout pour désarçonner un candidat ayant sérieusement préparé le concours. Cependant, en vue des révisions, c'est un sujet qui balaie de nombreux domaines et, si on lit les indications du corrigé pour savoir ce qu'il est possible de faire ou pas, ce problème varié mérite que l'on s'y attarde.

Indications

> La mention (HP) dans une indication signale une question hors programme, pour laquelle une explication détaillée est fournie dans le corrigé.

Q1 (HP)

Q3 Penser à regarder soigneusement les structures demandées en retour de fonction.

Q4 Ne pas oublier le modulo.

Q5 (HP)

Q6 Cette question et la suivante se rapprochent des algorithmes de première année.

Q8 (HP) Faire comme si l'on pouvait définir une constante en SQL comme `Imin` ou `Imax`.

Q9 Utiliser des requêtes imbriquées.

Q10 Écrire une requête portant sur une table dans laquelle la clause `WHERE` dépend d'une requête portant sur une autre table.

Q11 (HP)

Q12 On suppose que le programme proposé marche.

Q14 (HP) On peut deviner la réponse en s'inspirant de l'énoncé, sans plus de connaissances sur les dictionnaires.

Q16 (HP) Un invariant d'itération est l'équivalent d'un invariant de boucle pour les fonctions récursives.

Q22 (HP)

Q23 Il faut construire ici une fonction permettant d'évaluer si un moteur est défectueux ou non. Par exemple, on peut proposer un test comparant quelques solutions simulées et la sortie mesurée.

Tests de validation d'une imprimante

Q1 Le complément à 2 est une méthode de représentation des entiers relatifs sur un nombre fini de bits. Un des bits est utilisé pour coder le signe, les suivants permettent de représenter la valeur absolue du nombre. La façon de représenter cette valeur absolue n'est pas nécessaire pour répondre à cette question. On donne ici une réponse générale indépendante de la convention, l'emploi de la terminologie « complément à 2 » imposant en fait l'intervalle demandé.

En complément à 2, puisqu'un bit sert à coder le signe, cela signifie qu'il en reste 9 pour coder la valeur absolue, soit $2^9 = 512$ valeurs possibles. Traditionnellement, **on considère que l'on code ici les entiers compris entre -512 et $+511$.** Mais on peut représenter de façon générale tout intervalle de nombres entiers contenant $2^{10} = 1024$ valeurs, si l'on décide arbitrairement d'une convention.

Le résultat de la conversion peut prendre toute plage de valeurs entières de 1024 valeurs.

Cependant, rien n'interdit de coder plusieurs fois une même valeur. Ainsi, une autre méthode de représentation de nombres qui consiste à coder le signe par le premier bit et la valeur absolue par les suivants en tant qu'entier naturel conduit à coder deux fois le zéro. Par exemple, sur 4 bits, les binaires 0000 et 1000 représentent « respectivement » $+0$ et -0, codant deux fois la même chose. L'intérêt des différentes méthodes de représentations de nombres est précisément de pallier ce genre de défaut.

Q2 La résolution de la mesure se déduit de la possibilité de représenter 1024 éléments sur $N = 10$ bits. 2^N éléments permettent de définir $2^N - 1$ intervalles. La plage de tension P étant de 10 V, la résolution σ vaut

$$\sigma = \frac{P}{2^N - 1} = 1.10^{-2} \text{ V}$$

Q3 Le principe du programme est le suivant. On initialise une variable `nonstop` à la valeur booléenne `True`. On lit alors un à un les caractères reçus jusqu'à tomber sur un caractère d'en-tête. On change alors la valeur de `nonstop` pour sortir de la boucle. On lit par la suite le nombre N de données envoyées. On sait que la longueur totale de la trame est exactement $8 + 4N$: un caractère correspond à l'en-tête, trois au nombre de données envoyées, 4N permettent de détailler ces dernières et quatre caractères donnent le checksum. On utilise également à répétition la conversion du type chaîne de caractères vers le type entier grâce à `int`.

```
def lect_mesures():
    '''Fonction qui renvoie une trame à partir du premier en-tête'''
    nonstop=True
    resultat=[ ] #Liste que l'on renverra à la fin
    ###Recherche de l'en-tête
    while nonstop:
        test=com.read(1)
        if test in ['U','I','P']:#A-t-on trouvé l'en-tête ?
            resultat=[test]      #Si oui, on le stocke
            nonstop=False    #Et on fait en sorte de sortir du while
```

```
    ###Lecture du nombre de données à venir
    N=int(com.read(3)) #On stocke le nombre de données à lire
    ###Lecture des données
    donnees=[ ] #Liste vide contenant les données
    for inc in range(N):
        #On lit 4 caractères que l'on convertit
        #en entier pour les stocker
        donnees.append(int(com.read(4)))
    resultat.append(donnees) #On stocke donnees
    ##Ajout du checksum
    resultat.append(int(com.read(4)))
    return resultat
```

Q4 Notons une confusion de l'énoncé entre `mesure` et `mesures` qui peut déstabiliser à la lecture du sujet, surtout lors de l'emploi de la notation `mesures[]`, utilisée par exemple en Java... Ce qui est demandé est toutefois relativement compréhensible entre les lignes. Enfin, ne pas oublier le modulo 10000.

```
def check(mesure,CheckSum):
    '''Vérifie la validité d'un checksum'''
    #On calcule la somme des données reçues
    somme=0
    for x in mesure:
        somme += abs(x)
    #On teste la validité du checksum
    return somme%10000==CheckSum
```

Q5 On suppose ici que la bibliothèque `matplotlib.pyplot` a été importée sous l'alias `plt`.

```
def affichage(mesure):
    ###Création du vecteur des temps
    Temps=[ ]
    #On connaît exactement la plage nécessaire : on part de 0ms
    #et on va à 400ms par pas de 2ms. On met donc 401 pour atteindre
    #400 inclus mais sans dépasser cette valeur.
    for t in range(0,401,2):
        Temps.append(t)
    ###Création du vecteur des mesures
    mesures=[ ] #Initialisation des mesures
    for m in mesure: #On balaie les données fournies
        mesures+=[m*4e-3] #Conversion des données en intensités
    ###Représentation graphique proprement dite
    plt.plot(Temps,mesures)
    ###Compléments : axes et titre
    plt.xlabel('Temps (ms)')
    plt.ylabel('Intensité (A)')
    plt.title('Courant moteur')
```

Ce programme suivant est enrichi des commandes qui auraient permis d'obtenir tout le graphique proposé. Selon l'interface de développement, on peut être amené à ajouter la fonction `show` pour que le graphique créé s'affiche effectivement. On peut également le sauver à l'aide de la fonction `savefig`.