

Informatique - Composition 2

PCSI 2 - PTSI

2 heures

Consignes générales :

- ◇ les copies doivent être scannées et déposées sur le OneDrive PTSI dans le dossier Informatique. Les scans doivent être au format pdf avec un nom de la forme **X-Prenom.pdf** où X correspond au numéro de l'élève dans le classement alphabétique (liste de classe disponible en fin de sujet).
- ◇ les documents, calculatrices et autres appareils électroniques sont interdits ;
- ◇ les programmes seront rédigés en langage python ;
- ◇ dans l'écriture des fonctions, on ne prévoira pas d'exception ou de message d'erreur lorsque les arguments ne vérifient pas les conditions d'utilisation de la fonction ;
- ◇ les réponses recevront les justifications appropriées, soit sous forme de commentaires dans le code, soit rédigées séparément ;
- ◇ les indentations pourront être matérialisées par des traits verticaux, marquant précisément le début et la fin du bloc indenté ;
- ◇ on s'interdira l'utilisation des fonctions `max`, `min`, etc., ainsi que l'utilisation éventuelle de méthodes sur les listes autres que la méthode `u.append(e)` qui ajoute l'élément `e` à la fin de la liste `u` (similaire à `u = u + [e]`).

Dans tout le sujet on suppose que la bibliothèque `numpy` a été importée grâce à l'instruction

```
1 | import numpy as np
```

On pourra éventuellement utiliser les fonctions suivantes :

- ◇ `round(n)` arrondit le nombre `n` à l'entier le plus proche ;
- ◇ `np.ceil(x)` renvoie le plus petit entier supérieur ou égal à `x` ;
- ◇ `np.floor(x)` renvoie le plus grand entier inférieur ou égal à `x` ;
- ◇ `np.array(u)` crée un nouveau tableau contenant les éléments de la liste `u` ;
- ◇ `np.zeros(n, dtype)`, `np.zeros((n, m), dtype)` crée respectivement un vecteur à `n` éléments ou un tableau numpy à `n` lignes et `m` colonnes dont les éléments sont initialisés à zéro pour les types numériques ou `False` pour les types booléens ;
- ◇ `a.shape` est un tuple donnant la taille du tableau `a` pour chacune de ses dimensions.

Les 3 parties sont largement indépendantes.

Élasticité d'un brin d'ADN

La capacité des molécules d'ADN à participer à des mécanismes de réplication et de transcription ainsi qu'à s'organiser en chromosomes doit beaucoup à leur élasticité. Ainsi, l'étude de la réaction d'une molécule d'ADN aux contraintes mécaniques permet d'éclairer les processus biologiques mis en œuvre dans une cellule vivante.

À l'aide d'une expérience et de deux modèles mécaniques d'un brin d'ADN, ce sujet propose de caractériser l'élasticité de l'ADN. Pour cela, on suppose qu'on exerce une traction sur un brin d'ADN et on cherche à établir une relation entre la force utilisée et l'allongement de la molécule.

Pour répondre à une question, il est possible de faire appel aux fonctions définies dans les questions précédentes. Dans ce sujet, le terme « liste » appliqué à un objet Python signifie qu'il s'agit d'une variable de type `list`. Les termes « vecteur » et « tableau » désignent des objets numpy de type `np.ndarray`, respectivement à une dimension ou de dimension quelconque.

Une attention particulière sera portée à la lisibilité et à la simplicité du code proposé. En particulier, l'utilisation d'identifiants significatifs, l'emploi judicieux de commentaires et la description du principe de chaque programme seront appréciés.

1. Fonctions utiles

Cette partie définit une fonction qui pourra être utilisée dans la suite du sujet.

1. Écrire une fonction `variance(ls)` qui calcule la variance d'une liste de nombres `ls`, sans la modifier.

Pour rappel, la variance des n nombres x_1, x_2, \dots, x_n est la moyenne des carrés des écarts à la moyenne, c'est-à-dire :

$$\text{var}(x) = \frac{1}{n} \sum_{i=1}^n (x_i - \bar{x})^2 = \frac{1}{n} \sum_{i=1}^n x_i^2 - \bar{x}^2 \quad \text{avec} \quad \bar{x} = \frac{1}{n} \sum_{i=1}^n x_i.$$

Par exemple : `variance([1, 2, 3, 4])` renvoie `1.25`.

2. Écrire les instructions permettant de définir une liste `L1` contenant les éléments `{1,10,20,50}` puis d'en calculer la variance et de stocker le résultat dans `V1`.

2. Mesures expérimentales

Une molécule d'ADN est attachée à une de ses extrémités sur un support transparent, une microbille magnétique de diamètre $2,5 \mu\text{m}$ est greffée à son autre extrémité. À l'aide d'aimants, la molécule d'ADN est soumise à une force de traction notée \vec{F} .

Afin de caractériser l'élasticité du brin d'ADN, on cherche à mesurer son allongement pour différentes intensités de la force de traction.

L'intensité de la force de traction n'est pas accessible directement, nous allons l'évaluer indirectement.

Une fois le brin d'ADN mis en tension, son extrémité matérialisée par la bille ne reste pas immobile, elle est animée d'un mouvement aléatoire, dit mouvement brownien, dû à l'agitation des molécules du liquide qui l'entourent. En assimilant la molécule à un ressort, on montre que l'intensité de la force de traction est inversement proportionnelle aux fluctuations quadratiques moyennes de la position de la bille.

Une caméra CCD reliée à un ordinateur permet de photographier l'image de la bille (figure 1). Compte tenu de la taille de cette bille, on obtient une image de diffraction que nous allons analyser pour déterminer la position de la bille.

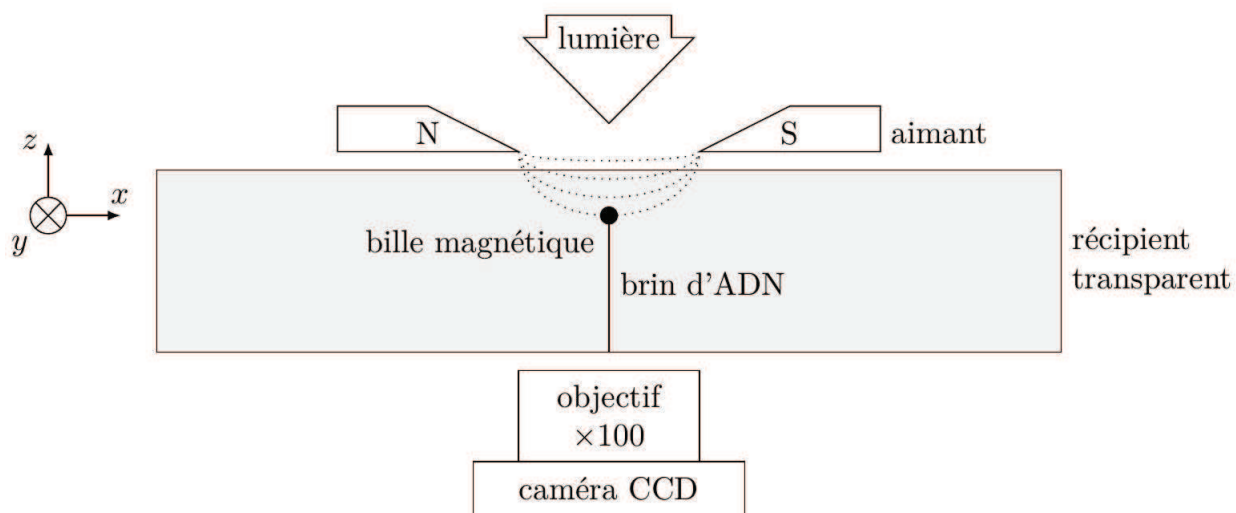


Figure 1 – Schéma du dispositif expérimental (échelle non respectée)

2.1. Position de la bille

La figure 2 donne, à gauche, un exemple d'image obtenue par la caméra CCD. Cette caméra est pilotée par un programme Python qui récupère chaque image sous la forme d'un tableau d'entiers à deux dimensions. Les images obtenues sont en niveau de gris, chaque pixel est codé sur 8 bits, soit une valeur comprise entre 0 (noir) et 255 (blanc).

3. Calculer l'espace mémoire occupé par une telle image de dimension 600×400 pixels. On donnera le résultat en ko. Quelle est la différence entre les kilo-octets (ko) et les kibi-octets (Kio).

Afin de repérer le centre de la figure de diffraction, l'image est convertie en noir et blanc inversé suivant une valeur seuil du niveau de gris : les pixels au-dessus du seuil deviennent noirs, ceux en dessous deviennent blancs.

Une fois ce seuillage effectué, on calcule le barycentre des pixels blancs de l'image seuillée pour obtenir la position de la bille.

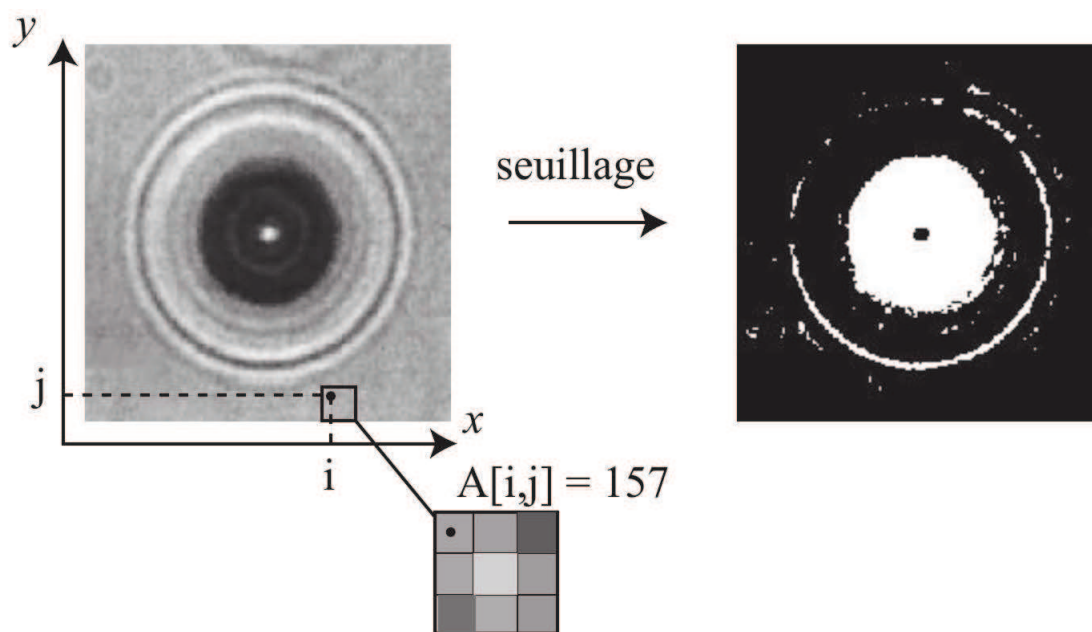


Figure 2 – Figure de diffraction d'une bille et opération de seuillage

4. Écrire une fonction `seuillage(A, seuil)` qui prend en argument un tableau d'entiers à deux dimensions `A` représentant un cliché de la caméra CCD et un entier `seuil`. La fonction construit un tableau de mêmes dimensions contenant la valeur 1 là où la valeur des pixels de l'image originale est strictement inférieure au seuil et la valeur 0 ailleurs (pixels supérieurs ou égaux au seuil).

On définit l'abscisse du barycentre d'un ensemble de points de même poids comme la moyenne des abscisses des points considérés. De même, on définit l'ordonnée du barycentre d'un ensemble de points de même poids comme la moyenne des ordonnées des points considérés.

5. On considère une petite image seuillée décrite par le tableau ci-dessous :

0	0	0	0	0
0	1	1	1	0
0	0	1	0	0
0	0	1	0	0

Reproduire ce tableau sur votre copie, et entourer le pixel associé au barycentre des pixels à 1 de cette image. Indiquer l'indice de la ligne ainsi que l'indice de la colonne de ce barycentre.

6. Écrire une fonction `pixel_centre_bille(A)` qui prend en paramètre une image seuillée telle que produite par la fonction `seuillage` et renvoie la liste des deux indices (ligne et colonne) du pixel le plus proche du centre de la bille correspondant au barycentre des pixels à 1.

On dispose de la fonction `prendre_photo()` qui déclenche la prise d'un cliché par la caméra CCD et renvoie l'image prise sous la forme d'un tableau à deux dimensions tel que décrit dans l'introduction.

7. Écrire une fonction `positions(n, seuil)` qui prend `n` photographies de la bille et renvoie la liste de ses positions dans chaque photographie en seuillant les images à la valeur `seuil`. Le résultat de cette fonction est donc une liste de `n` listes de deux entiers correspondant à l'indice de ligne et de colonne des positions successives du centre de la bille au cours de son mouvement brownien.

Par exemple, pour une bille repérée successivement aux positions (98,50), (100, 52), (102, 48), la fonction `positions` renverra la liste `[[98, 50], [100, 52], [102, 48]]`.

Le capteur CCD est positionné parallèlement au plan (xOy) et ses pixels sont carrés. La caméra a été calibrée dans les conditions de l'expérience : un pixel correspond à un carré du plan (xOy) de côté d .

8. Définir une fonction `fluctuations(lsP, d)` qui prend en paramètre une liste `lsP` de positions successives de la bille (comme celle produite par la fonction `positions`) et la longueur `d` correspondant à la longueur d'un pixel et calcule la valeur moyenne des déplacements quadratiques de la bille c'est-à-dire la moyenne des distances au carré entre chaque position et la position d'équilibre de la bille (correspondant au barycentre ou encore à la position moyenne des différentes positions observées).

Indication - Cette distance au carré peut s'exprimer de la manière suivante (en notant (x_i, y_i) les coordonnées de la position i) :

$$D^2 = d^2 \sum_{i=1}^n (x_i - x_G)^2 + (y_i - y_G)^2 = d^2 (\text{var}(x) + \text{var}(y))$$

2.2. Allongement du brin d'ADN

La position de la bille étant déterminée dans le plan (xOy), nous allons maintenant nous intéresser à sa cote, c'est-à-dire sa position dans la direction perpendiculaire à la caméra.

Pour déterminer la position de la bille suivant z , nous utilisons une méthode basée sur la répartition des cercles de la figure de diffraction. Pour cela, nous construisons un profil de cette figure en découpant l'image seuillée en anneaux concentriques équidistants centrés sur la position de la bille. Le décompte de la proportion de pixels blancs dans chaque anneau fournit un profil de la figure de diffraction qui permet de calculer la cote z de la bille en tenant compte des paramètres de calibration de la caméra.

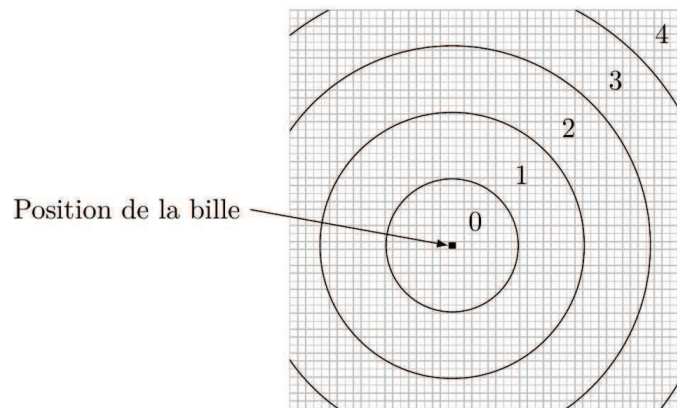


Figure 3 – Exemple de découpage d'une image en cinq anneaux concentriques. Les numéros correspondent à l'indice des anneaux.

On souhaite écrire une fonction `profil(A,n)` qui construit le profil d'une figure de diffraction seuillée A en la découpant en n anneaux concentriques.

Cette fonction devra renvoyer une liste de n nombres, compris entre 0 et 1, qui donnera la proportion de pixels blancs compris dans chaque anneau. Un pixel sera considéré comme contenu dans un anneau si son centre s'y trouve. L'élément d'indice 0 du résultat correspond à la bande la plus proche du centre de la figure (position de la bille). Cette fonction que vous mettrez en œuvre à la question 12 fera appel à trois fonctions que nous allons définir maintenant (questions 9 à 11).

9. Écrire une fonction `distance_maximale(ic,jc,P,Q)` qui détermine la distance maximale qu'il existe entre la position de la bille (ic,jc) et le pixel le plus éloigné de l'image de dimension $P \times Q$. Le pixel le plus éloigné étant forcément à l'un des 4 coins de l'image.

10. Écrire une fonction `largeur_anneau(ic,jc,P,Q,n)` renvoyant le rayon r de l'anneau d'indice 0. Les n anneaux sont centrés sur la bille de coordonnées (ic,jc) d'une image de dimension $P \times Q$ et régulièrement répartis entre le centre et le coin le plus éloigné. Ainsi, l'anneau d'indice 0 aura pour rayon r , celui d'indice 1, aura pour rayon $2r$, et celui d'indice $n-1$, nr .

11. Écrire une fonction `indice_anneau(ic,jc,P,Q,i,j,n)` qui renvoie l'indice de l'anneau dans lequel le pixel (i,j) se trouve dans une image de dimension $P \times Q$.

12. À l'aide de toutes les fonction précédentes, écrire la fonction `profil(A,n)`.

13. Si on travaille sur une image carrée de dimension $p \times p$ pixels, quelle est la complexité de la fonction `profil` ?

Pour une configuration expérimentale donnée, il est ainsi possible en prenant une série de clichés de déterminer l'amplitude du mouvement brownien de l'extrémité du brin d'ADN ainsi que sa position en trois dimensions. Ces éléments permettent alors de déterminer l'intensité de la force de traction appliquée au brin d'ADN ainsi que son

allongement. En modifiant les aimants, on peut faire varier l'intensité du champ magnétique et donc la force appliquée au brin d'ADN. En renouvelant l'expérience, on obtient ainsi une série de points expérimentaux correspondant à diverses valeurs de force et d'allongement.

3. Modèle du ver

Le « modèle du ver » est un modèle souvent utilisé pour décrire le comportement mécanique de certains polymères. Dans ce modèle, la molécule étudiée est représentée par une succession de segments semi-rigides orientés grossièrement dans la même direction. Il permet d'obtenir une expression simplifiée de F , l'intensité de la force de traction \vec{F} , en fonction de z , l'allongement de la molécule :

$$F(z) = \frac{k_B T}{L_p} \left(\frac{1}{4(1 - z/L_0)^2} - \frac{1}{4} + \frac{z}{L_0} \right) \quad (1)$$

où k_B est la constante de Boltzman et T la température. Ce modèle est paramétré par deux longueurs :

- ◊ L_p , longueur de persistance représentant la longueur typique sur laquelle le polymère maintient sa forme malgré les déformations dues à l'agitation thermique ;
- ◊ L_0 , extension maximale du polymère.

3.1. Étude de la fonction $F(z)$

14. Écrire une fonction `force(z)` qui calcule la force donnée par la formule (1) pour chaque élément du vecteur z . Cette fonction utilise les grandeurs préalablement définies `Lp`, `L0` et `T` renvoie un vecteur de même taille que z contenant le résultat du calcul pour chaque composante de z . La variable `K_B` fournit la valeur de la constante de Boltzman et est elle-aussi prédéfinie.

Le tracé de la force F en fonction de la position verticale de la bille magnétique z est proposé sur la figure 4.

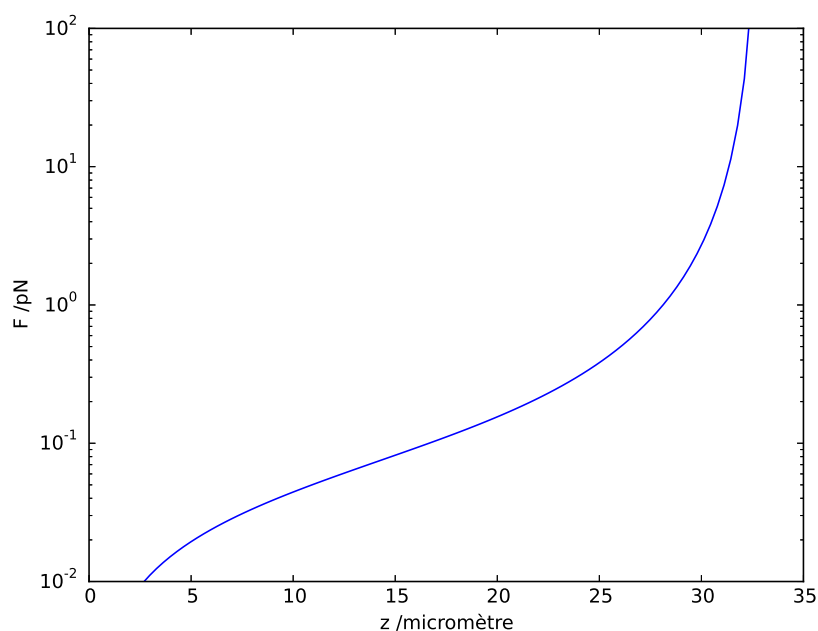


Figure 4 – Graphe de la force F en fonction de la position verticale de la bille magnétique z .

15. En vous aidant de la figure 4, écrire un script permettant de déterminer la valeur de z à 0,01 pm près pour laquelle la force F vaut 0,1 pN. Ce script utilisera une fonction `dichotomie` qui devra être clairement écrite.

On cherche à déterminer le travail W de la force $F(z)$ lors du déplacement longitudinal de la bille magnétique sur une distance $D = z_2 - z_1$ selon z . On rappelle que :

$$W = \int_{z_1}^{z_2} F(z) dz$$

16. En se basant sur la méthode des rectangles au centre, écrire une fonction `travail(F,z_1,z_2)` permettant de calculer ce travail.

La bille magnétique, de masse m , est soumise à la force F mais subit aussi un transfert de quantité de mouvement dû au mouvement Brownien modélisable par une force $F_B = F_0 \cos\left(2\pi \frac{t}{T}\right)$. Son mouvement longitudinal selon z est ainsi décrit par l'équation différentielle :

$$m \frac{d^2 z}{dt^2} = -F(z) + F_0 \cos\left(2\pi \frac{t}{T}\right) \quad (2)$$

On suppose que la bille est initialement à la position z_0 , animée d'une vitesse initiale nulle.

17. Écrire l'équation différentielle précédente sous la forme d'un problème de Cauchy.

18. Rappeler la relation de récurrence appliquée dans la méthode d'Euler. Écrire ensuite une fonction `Euler` dont on précisera les arguments, permettant de renvoyer un vecteur associé à la solution d'un problème de Cauchy donné.

19. Proposer un script permettant de calculer le vecteur $z(t)$ solution de l'équation différentielle (2) pour des temps compris en 0 et $10T$. Le pas de la méthode sera de $10^{-2}T$. On suppose que les paramètres z_0 , m , F_0 et T sont affectés aux variables `z0`, `m`, `F0` et `T`.

3.2. Calcul des paramètres L_p et L_0

Les valeurs de L_p et L_0 ne sont pas accessibles directement pour une molécule d'ADN. L'objectif de cette partie est de jeter les bases d'un algorithme permettant d'obtenir les valeurs optimales de L_p et L_0 .

Soit ϕ une fonction de classe \mathcal{C}^2 sur \mathcal{R} présentant un minimum local. On rappelle que

$$\frac{\phi(x(1+h)) - \phi(x(1-h))}{2xh} \quad (3)$$

est une expression approchée d'ordre 2 de la dérivée de ϕ en x (notée $\phi'(x)$).

On suppose que l'ordinateur utilisé représente les nombres flottants sur 64 bits avec un bit de signe, 11 bits d'exposant et 52 bits de mantisse.

20. Exprimer puis donner le nombre de chiffres significatifs décimaux donnés par ce codage.

21. Justifier que les valeurs $h = 1$ et $h = 10^{-16}$ ne permettent pas obtenir une bonne approximation du nombre dérivé $\phi'(x)$. Proposer alors une valeur adaptée de h .

22. Écrire une fonction `derive(phi, x, h)` qui calcule une valeur approchée de la dérivée au point `x` de `phi`, fonction réelle d'une variable réelle, où `h` correspond au h de la formule 3.

23. Écrire une fonction `derive_seconde(phi, x, h)` permettant d'obtenir une approximation de la dérivée seconde de la fonction `phi` au point `x`.

24. Rappeler à l'aide d'un graphe le principe de la méthode de Newton permettant de résoudre une équation à une inconnue. Écrire une fonction `min_local(phi, x0, h)` basée sur la méthode de Newton permettant de trouver l'abscisse d'un minimum local de la fonction `phi`. La valeur approchée de cette abscisse vérifiera $|\phi'(x)| < 10^{-7}$.

Liste des noms de scan

01-Rania.pdf
02-Timothee.pdf
03-Arnaud.pdf
04-Eliott.pdf
05-Guillaume.pdf
06-Gaetan.pdf
07-Romane.pdf
08-Erwan.pdf
09-Ombeline.pdf
10-Gregoire.pdf
11-Emile.pdf
12-Aymeric.pdf
13-Augustin.pdf
14-ClementD.pdf
15-Vincent.pdf
16-ClementG.pdf
17-Matthieu.pdf
18-Baptistin.pdf
19-Gautier.pdf
20-Claire.pdf
21-Jorian.pdf
22-Joseph.pdf
23-Francois.pdf
24-PaulL.pdf
25-Hugo.pdf
26-Lorenzo.pdf
27-Blanche.pdf
28-Amaury.pdf
29-Kilian.pdf
30-Hortense.pdf
31-PaulT.pdf
32-Leo.pdf
33-Philippine.pdf