

Planche 1 : On considère la matrice $A = \begin{pmatrix} 7 & 5 & 1 \\ 6 & -1 & 2 \\ 6 & 1 & 3 \end{pmatrix}$.

① a) Montrer que λ est une valeur propre de A si, et seulement si, elle est solution de l'équation :

$$\lambda^3 - 9\lambda^2 - 27\lambda + 53 = 0$$

b) On pose $f(x) = x^3 - 9x^2 - 27x + 53$. Calculer $f(0)$ et $f(3)$ puis dresser le tableau de variation de f en précisant les limites en $\pm\infty$.

c) En déduire que A possède 3 valeurs propres que l'on notera $\lambda_1 < \lambda_2 < \lambda_3$ (✎ On ne cherchera pas à les calculer).

d) A l'aide de l'algorithme de dichotomie en Python, déterminer une valeur approchée à 10^{-2} près de ces trois valeurs (✎ On pourra calculer $f(-4)$ et $f(12)$ à l'aide de Python).

e) Montrer qu'il existe une matrice P inversible telle que $A = PDP^{-1}$ où $D = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$.

② Maintenant on va s'intéresser aux matrices qui commutent avec A (c'est-à-dire telles que $AM = MA$). On note donc $\mathcal{C}(A)$ l'ensemble des matrices de $\mathcal{M}_3(\mathbb{R})$ qui commutent avec A , et $\mathcal{C}(D)$ l'ensemble de celles qui commutent avec D .

a) Montrer que $\mathcal{C}(A)$ est un sous-espace vectoriel de $\mathcal{M}_3(\mathbb{R})$.

b) Montrer que $M \in \mathcal{C}(D)$ si et seulement si M est diagonale.

c) Montrer l'équivalence suivante : $M \in \mathcal{C}(A) \Leftrightarrow P^{-1}MP \in \mathcal{C}(D)$.

d) En déduire une base de $\mathcal{C}(A)$ (en fonction de P et P^{-1}) ainsi que sa dimension.

e) Soit $Q \in \mathbb{R}[X]$ non nul tel que $Q(A) = 0_{\mathcal{M}_3(\mathbb{R})}$. Montrer que $\deg(Q) \geq 3$ (✎ On pourra montrer que les valeurs propres de A sont racines de Q).

f) En déduire que (I_3, A, A^2) est une base de $\mathcal{C}(A)$.

Rappel. Algorithme de dichotomie On considère une fonction g continue sur un segment $[a, b]$.

On suppose que g s'annule exactement une fois sur $[a, b]$ en un point que l'on note α

On définit les suites $(a_k)_{k \geq 0}$ et $(b_k)_{k \geq 0}$ de la façon suivante :

— $a_0 = a$ et $b_0 = b$

— Pour tout entier naturel k on note $c_k = \frac{a_k + b_k}{2}$.

si $g(a_k)g(c_k) \leq 0$, alors $a_{k+1} = a_k$ et $b_{k+1} = c_k$
sinon $a_{k+1} = c_k$ et $b_{k+1} = b_k$

On sait alors que les suites (a_k) et (b_k) convergent toutes les deux vers α .

Exercice sans préparation :

① Créer avec Python une matrice d'ordre 3 (de type `array`) dont les coefficients sont des entiers au hasard pris dans $\llbracket 0, 1 \rrbracket$.

② Ecrire une fonction `testSymetrie(A)` qui renvoie `True` ou `False` selon que la matrice A (de type `array`) est symétrique ou non.

③ Estimer la proportion de matrices d'ordre 3, formée comme en 1. qui sont symétriques.

Planche 2 :

Un magasin possède une caisse, toutes les personnes qui sortent du magasin doivent passer par cette caisse y compris si le client n'achète pas d'article. Pour tout $k \in \mathbb{N}^*$, on note X_k la variable aléatoire réelle égale au nombre d'articles achetés par le k -ième client. On suppose que ces variables sont mutuellement indépendantes et suivent une loi de Poisson de paramètre μ . On note N le nombre aléatoire de clients qui passent par la caisse au cours de la première heure, N suit une loi de Poisson de paramètre λ et est indépendante des X_k . On souhaite étudier le nombre X d'articles passant par cette caisse durant la première heure.

1. Exprimer X en fonction de N et des X_i .
2. Écrire une fonction Python `X` qui prend en entrée 1 pour λ et `m` pour μ et qui simule la variable aléatoire X .

On pourra pour cela utiliser la fonction `poisson` du module `numpy.random`.

3. Soit $n \in \mathbb{N}$. Déterminer $\mathbb{P}_{(N=0)}(X = n)$.
4. Montrer par récurrence sur $k \in \mathbb{N}^*$ que $X_1 \cdots + X_k$ suit une loi de Poisson de paramètre $k\mu$.
5. En déduire que pour tout $n \in \mathbb{N}^*$, $\mathbb{P}(X = n) = \sum_{k=1}^{+\infty} \frac{\lambda^k (k\mu)^n}{k!n!} e^{-(\lambda+k\mu)}$.
6. Montrer que $\mathbb{E}(X) = \lambda\mu$. On admettra l'existence de $\mathbb{E}(X)$ et on pourra procéder sans justification à une permutation de deux sommes infinies.
7. Montrer que pour tout $n \in \mathbb{N}^*$:

$$\mathbb{P}(X = 0) = \exp(\lambda(e^{-\mu} - 1)) \text{ et } \mathbb{P}(X = n) = \frac{\mu^n e^{-\lambda}}{n!} f_n(\lambda e^{-\mu})$$

où $f_n(x) = \sum_{k=1}^{+\infty} k^n \frac{x^k}{k!}$. On pose aussi $f_0(x) = \sum_{k=1}^{+\infty} \frac{x^k}{k!}$.

8. Exprimer f_0 et f_1 à l'aide de fonctions usuelles.
9. Montrer que pour tout $n \in \mathbb{N}$ et tout $x \in \mathbb{R}$:

$$f_{n+1}(x) = x + \sum_{i=0}^n x \binom{n}{i} f_i(x).$$

Indication : On pourra utiliser le changement de variables $\ell = k - 1$.

10. En déduire une expression de la fonction f_2 .
11. Écrire une fonction `f` qui prend en entrée un entier `n` et un réel `x` et qui renvoie la valeur $f_n(x)$. On pourra utiliser la fonction `binom` du module `scipy.special`.
12. Utiliser la fonction précédente pour écrire une fonction Python qui prend en entrée deux réels λ et μ et un entier n et qui renvoie $\mathbb{P}(X = n)$. On pourra utiliser la fonction `factorial` du module `scipy.special`.

Exercice sans préparation :

Écrire en Python une fonction qui retourne la position de départ et la longueur de la plus longue suite de valeurs identiques dans une séquence (chaîne de caractères ou liste).

Exemple : si `L = [3,3,2,2,3,3,3,3,1]` alors cette fonction renvoie le couple (4, 5).

Planche 3 :**Rappel. Algorithme de dichotomie**

On considère une fonction g continue sur un segment $[a, b]$.

On suppose que g s'annule exactement une fois sur $[a, b]$ en un point que l'on note α

On définit les suites $(a_k)_{k \geq 0}$ et $(b_k)_{k \geq 0}$ de la façon suivante :

- $a_0 = a$ et $b_0 = b$
- Pour tout entier naturel k on note $c_k = \frac{a_k + b_k}{2}$.
 si $g(a_k)g(c_k) \leq 0$, alors $a_{k+1} = a_k$ et $b_{k+1} = c_k$
 sinon $a_{k+1} = c_k$ et $b_{k+1} = b_k$

On sait alors que les suites (a_k) et (b_k) convergent toutes les deux vers α .

On étudie dans cet exercice, pour tout entier naturel n non nul, les solutions sur $\mathbb{R}^{+*}$ de l'équation

$$(E_n) : \ln x + x = n.$$

À cet effet, on introduit la fonction f définie sur $\mathbb{R}_+^*$ par l'expression

$$f(x) = \ln x + x.$$

- ① a) Montrer que pour tout entier naturel non nul n , l'équation (E_n) admet une unique solution, notée x_n .
 b) En utilisant l'algorithme de dichotomie, déterminer des valeurs approchées à 10^{-3} près des termes x_n pour n allant de 1 à 10, et les représenter graphiquement.
 c) Étudier les variations de la suite $(x_n)_{n \in \mathbb{N}}$ (on pourra comparer $f(x_n)$ et $f(x_{n+1})$).
- ② a) Montrer que $\forall x \in \mathbb{R}^{+*}, \ln x < x$.
 b) Prouver que $\forall n \in \mathbb{N}^*, \frac{n}{2} \leq x_n \leq n$.
 c) Déterminer $\lim_{n \rightarrow +\infty} x_n$.
- ③ a) Montrer que $\lim_{n \rightarrow +\infty} \frac{\ln(x_n)}{n} = 0$. En déduire un équivalent de x_n .
 b) Déterminer $\lim_{n \rightarrow +\infty} x_{n+1} - x_n$.
- ④ On note : $\forall n \geq 2, u_n = \frac{n - x_n}{\ln n}$.
 a) Exprimer $u_n - 1$ en fonction de x_n et n . En déduire $\lim_{n \rightarrow +\infty} u_n$.
 b) En déduire que $1 - u_n \sim \frac{1}{n}$.
 c) En déduire que

$$x_n = n - \ln n + \frac{\ln n}{n} + o\left(\frac{\ln n}{n}\right)$$

Exercice sans préparation :

Écrire en Python une fonction qui détermine si tous les `True` d'une liste de booléens sont côte à côte.

Planche 4 :

On considère une pièce qui fait Pile avec probabilité $p \in]0, 1[$ et Face avec probabilité $q = 1 - p$. Soit n un entier non nul fixé.

On considère n joueurs qui lancent chacun la pièce jusqu'à obtenir Pile. Le(s) gagnant(s) sont désignés comme ceux qui ont fait le moins de lancers.

Pour $i \in \llbracket 1, n \rrbracket$, on pose X_i le nombre de lancers du i -ième joueur, et on note N le nombre de gagnants.

- ① Quelle est la loi de X_i , pour $i \in \llbracket 1, n \rrbracket$?

Rappeler son espérance et sa variance.

- ② Calculer pour tout $i \in \llbracket 1, n \rrbracket$ et tout $j \in \mathbb{N}$, $\mathbb{P}(X_i > j)$.

- ③ On note $Y = \min(X_1, X_2, \dots, X_n)$.

Calculer $\mathbb{P}(Y > j)$ pour tout $j \in \mathbb{N}$.

En déduire la loi de Y , son espérance et sa variance.

- ④ Écrire une fonction Python `NbMin(L)` prenant en argument une liste `L` et renvoyant le nombre de fois où la valeur minimale apparaît dans la liste `L`.

- ⑤ En déduire une fonction Python `N(n,p)` qui, prenant en argument la valeur de n et p , simule l'expérience aléatoire décrite et renvoie la valeur de N .

- ⑥ Calculer $\mathbb{P}(N = n)$.

- ⑦ Montrer que :

$$\forall k \in \llbracket 1, n \rrbracket, \mathbb{P}(N = k) = \frac{\binom{n}{k} p^k q^{n-k}}{1 - q^n}.$$

- ⑧ En déduire l'espérance de N et la variance de N .

- ⑨ Vérifier la valeur de $\mathbb{E}(N)$ à l'aide d'estimations construites grâce à la fonction `N` de la question 5.

Exercice sans préparation :

On considère une liste L de longueur n formée d'une permutation des entiers de 0 à $n - 1$. On appelle « point fixe » une valeur de cette liste qui coïncide avec son indice.

Si $L = [2, 4, 1, 3, 0]$, 3 est un point fixe.

Écrire une fonction `compter` qui prend en entrée une telle liste et qui renvoie le nombre de points fixes (La fonction renvoie 0 s'il n'y a aucun point fixe).

Planche 5 :

On munit $E = \mathbb{R}^3$ du produit scalaire usuel que l'on notera $\langle \cdot, \cdot \rangle$. On considère l'endomorphisme f de $\mathbb{R}^3$ dont la matrice dans la base canonique $\mathcal{B} = (e_1, e_2, e_3)$ de $\mathbb{R}^3$ est :

$$A = \begin{pmatrix} \frac{5}{2} & 1 & \frac{1}{2} \\ 1 & 2 & 1 \\ \frac{1}{2} & 1 & \frac{5}{2} \end{pmatrix}.$$

- ① a) Montrer que A est diagonalisable.
b) Que permet le programme suivant ?

```
1      import numpy as np
2      import numpy.linalg as al
3
4      A = np.array([[5/2, 1, 1/2], [1, 2, 1], [1/2, 1, 5/2]])
5      I = np.eye(3)
6
7      r = al . matrix_rank (A - I)
8      s = al . matrix_rank (A -2* I)
9      t = al . matrix_rank (A -4* I)
```

- c) A l'aide de Python, déterminer une base de $\mathcal{M}_{3,1}(\mathbb{R})$ formée de vecteurs propres de A .
d) Pourquoi cette base est-elle orthogonale ?
e) Proposer une base orthonormale de E formée de vecteurs propres de f . On notera $\mathcal{B}' = (e'_1, e'_2, e'_3)$ cette base.
② On considère l'application ϕ définie sur E^2 par :

$$\forall (u, v) \in E^2, \quad \phi(u, v) = \langle u, f(v) \rangle.$$

- a) Soit u (resp. v) un vecteur de E de coordonnées X (resp. Y) dans $\mathcal{B}$.
i. Exprimer $\phi(u, v)$ en fonction de A , X et Y .
ii. Exprimer $\phi(v, u)$ en fonction de A , X et Y .
iii. Montrer que $\phi(u, v) = \phi(v, u)$.
b) On note pour tout $u \in E$, F_u l'ensemble des vecteurs orthogonaux à u et :

$$F'_u = \{v \in E \text{ tel que } \phi(u, v) = 0\}.$$

- i. Montrer que si u est un vecteur propre de f , alors $F_u = F'_u$.
ii. A-t-on toujours $F_u = F'_u$?

Exercice sans préparation :

Écrire en Python une fonction prenant en entrée une liste de nombres entiers entre et qui retourne le nombre le plus fréquent. Le comportement est au choix en cas d'égalité.

Planche 6 :

Soit $(\Omega, \mathcal{A}, \mathcal{P})$ un espace probabilisé et X une variable aléatoire sur cet espace suivant la loi normale centrée réduite. On note φ la fonction densité continue de X , Φ sa fonction de répartition, f la fonction $x \mapsto P(X > x)$ et (E) l'équation différentielle :

$$\forall x \in \mathbb{R}, y'(x) + xy(x) = 0.$$

On considère la fonction M définie par : $\forall x \in \mathbb{R}, M(x) = \frac{f(x)}{\varphi(x)}$.

- ① Résoudre l'équation différentielle (E) . Montrer que φ est solution de (E) .
- ② Montrer que pour tout $x > 0$, on a : $f(x) = - \int_x^{+\infty} \frac{\varphi'(t)}{t} dt$.
- ③ Montrer que pour tout $x > 0$, on a : $f(x) \leq \frac{\varphi(x)}{x}$. Déterminer la limite de M en $+\infty$.
- ④ a) Montrer que f est dérivable et exprimer sa fonction dérivée f' en fonction de φ .
b) Montrer que la fonction M est dérivable. Calculer sa dérivée M' à l'aide des fonctions f et φ .
En déduire le sens de variation de M sur $\mathbb{R}^+$.
- ⑤ a) Écrire un script Python permettant d'afficher sur un même graphique les courbes des fonctions $x \mapsto M(x)$, $x \mapsto \frac{1}{x}$ et $x \mapsto \frac{1}{x} - \frac{1}{x^3}$ sur l'intervalle $]0, 5]$.
Pour calculer $f(x)$, on pourra utiliser la fonction `norm.sf()` du module `scipy.stats`.
b) Montrer, à l'aide d'une intégration par parties que :

$$\forall x > 0, f(x) = \frac{\varphi(x)}{x} + \int_x^{+\infty} \frac{\varphi'(t)}{t^3} dt.$$

- c) Montrer que pour $x > 0$, on a :

$$\frac{1}{x} - \frac{1}{x^3} \leq M(x) \leq \frac{1}{x}$$

(on pourra intégrer une deuxième fois par parties).

En déduire un équivalent de $M(x)$ lorsque x tend vers $+\infty$.

Exercice sans préparation :

Écrire une fonction d'argument une liste d'entiers et renvoyant le nombre de minimums de cette liste.

Planche 7 : Pour tout entier naturel n , on pose : $I_n = \int_0^1 ((1-x)e^x)^n dx$

① Justifier que la suite (I_n) est ainsi bien définie, et calculer I_0 et I_1 .

② a) Justifier que pour tout entier naturel n ,

$$I_n = \lim_{N \rightarrow +\infty} \left(\frac{1}{N} \sum_{k=0}^{N-1} \left(\left(1 - \frac{k}{N}\right) e^{k/N} \right)^n \right).$$

b) Écrire alors une fonction en Python qui prend en argument un entier n et renvoie une valeur approchée de I_n .

c) Utiliser la fonction précédente pour donner une conjecture sur la limite éventuelle de $\sqrt{n}I_n$ lorsque n tend vers $+\infty$.

③ a) Démontrer que pour tout réel x , $e^{-x} \geq 1 - x$.

b) Étudier les variations de la suite (I_n) .

④ Pour tout réel $x \in]0, 1[$, on pose $H(x) = -\frac{2}{x^2}(x + \ln(1-x))$.

a) Démontrer que la fonction H est prolongeable par continuité en 0.

On notera encore H son prolongement, et on admet que H réalise une bijection croissante de $[0, 1[$ vers $[1, +\infty[$.

b) Démontrer que :

$$\forall n \in \mathbb{N}^*, I_n = \int_0^1 \exp\left(-n \frac{x^2}{2} H(x)\right) dx = \frac{1}{\sqrt{n}} \int_0^{\sqrt{n}} \exp\left(-\frac{u^2}{2} H\left(\frac{u}{\sqrt{n}}\right)\right) du.$$

c) Donner la valeur de $\int_0^{+\infty} \exp\left(-\frac{u^2}{2}\right) du$, et en déduire que pour tout entier naturel non nul n , $I_n \leq \sqrt{\frac{\pi}{2n}}$.

⑤ Soit (u_n) une suite de réels appartenant à $[0, 1[$ qui converge vers 0, et telle que la suite $(\sqrt{n}u_n)$ diverge vers $+\infty$.

a) Donner l'exemple d'une telle suite.

b) On pose $\forall n \in \mathbb{N}$, $v_n = H(u_n)$.

$$\text{Démontrer que : } \forall n \in \mathbb{N}^*, I_n \geq \int_0^{u_n} \exp\left(-\frac{nx^2}{2} H(u_n)\right) dx \geq \frac{1}{\sqrt{nv_n}} \int_0^{\sqrt{nv_n}u_n} e^{-u^2/2} du.$$

c) En déduire que $I_n \underset{n \rightarrow \infty}{\sim} \sqrt{\frac{\pi}{2n}}$.

Exercice sans préparation :

Soit $n \in \mathbb{N}^*$. On parle de diviseur strict de n pour désigner tout diviseur k de n tel que $1 \leq k < n$. On note s_n la somme des diviseurs stricts de n . On dit que l'entier n est parfait si $s_n = n$. En Python, la commande `n%k` renvoie le reste de la division de n par k (ce reste est nul si, et seulement si, k divise n)

① Écrire une fonction `s(n)` qui renvoie la somme des diviseurs stricts d'un entier naturel non nul.

② Écrire une fonction `prop_parf(N)` qui renvoie la proportion des nombres compris entre 1 et N qui sont parfaits.

③ Écrire une fonction `trouve_parf(k)` qui renvoie le k -ième nombre entier parfait.

Planche 8 :

Soient a_1, a_2 et a_3 trois réels distincts.

Soit A la matrice définie par :

$$\begin{pmatrix} 1 & a_1 & a_1^2 \\ 1 & a_2 & a_2^2 \\ 1 & a_3 & a_3^2 \end{pmatrix}.$$

Pour tout $i \in \llbracket 1, 3 \rrbracket$, on pose :

$$s_i = \sum_{\substack{j=1 \\ j \neq i}}^3 a_j, \quad p_i = \prod_{\substack{j=1 \\ j \neq i}}^3 a_j \quad \text{et} \quad d_i = \prod_{\substack{j=1 \\ j \neq i}}^3 (a_i - a_j).$$

- ① Écrire une fonction Python qui prend en argument une liste contenant les valeurs de a_1, a_2 et a_3 et renvoie la matrice A , sous forme de liste de listes.
- ② Écrire une fonction Python qui prend en argument un entier i et une liste contenant les valeurs de a_1, a_2 et a_3 , et renvoie une liste contenant les valeurs de s_i, p_i et d_i .
- ③ Soit φ l'application définie par :

$$\forall P \in \mathbb{R}_2[X], \quad \varphi(P) = (P(a_1), P(a_2), P(a_3)).$$

- a) Démontrer que φ est une application linéaire de $\mathbb{R}_2[X]$ dans $\mathbb{R}^3$.
 - b) On note $\mathcal{B} = (1, X, X^2)$ la base canonique $\mathbb{R}_2[X]$, et $\mathcal{B}' = (e_1, e_2, e_3)$ la base canonique de $\mathbb{R}^3$.
Déterminer la matrice représentative de φ dans les bases $\mathcal{B}$ et $\mathcal{B}'$.
 - c) Déterminer le noyau de φ et en déduire que φ admet une réciproque, notée φ^{-1} .
- ④ Pour tout entier $i \in \llbracket 1, 3 \rrbracket$, on pose : $L_i(X) = \frac{1}{d_i} \prod_{\substack{j=1 \\ j \neq i}}^3 (X - a_j)$.
- a) Démontrer que : $\forall i \in \llbracket 1, 3 \rrbracket, L_i = \varphi^{-1}(e_i)$.
 - b) À l'aide des questions précédentes, démontrer que la matrice A est inversible, et déterminer A^{-1} .
(On exprimera les coefficients de A^{-1} en fonction des réels s_i, p_i et d_i .)
 - c) Écrire une fonction Python qui prend en argument une liste contenant les valeurs de a_1, a_2 et a_3 et renvoie la matrice A^{-1} , sous forme de liste de listes. Appliquer cette fonction avec $a_1 = 2, a_2 = 3$ et $a_3 = 4$.

Exercice sans préparation :

Écrire en Python une fonction qui prend en entrée une liste croissante L de nombres et qui retourne l'indice i tel que l'écart entre $L[i]$ et $L[i+1]$ soit le plus petit écart entre deux nombres de L .