Sommaire

CHAPITRE 1

Banque Agro & Véto — Mathématiques & Informatique

Résumé/Plan

Vous trouverez dans cette partie du polycopié des anciens sujets d'oraux de tous types : des planches hybrides Mathématiques & Informatique pour Agro-Véto, et des planches individuelles par domaine quant à G2E. Je remercie par avance tous les collègues & étudiants ayant bien voulu les partager (que ce soit au lycée ou *via* l'UPA) !

Organisation : 24 heures sont prévues dans l'emploi du temps : 20 heures de passages au tableau sur des planches Agro-Véto, 4 heures seront consacrées aux planches G2E (les étudiants souhaitant en préparer d'avantage peuvent le faire en autonomie, des corrections seront proposées en fin de session)

Il n'y a pas de réussite facile, ni d'échecs définitifs

Marcel Proust

✜ ***À propos de l'organisation*** **1 heure** sera allouée à chaque planche Agro-Véto, structurée de la manière suivante :

❶ réflexion commune de la classe sur la planche (15 minutes),

❷ passage du binôme ayant préparé la planche, muni d'un ordinateur (20 minutes — durée au concours),

❸ commentaires sur la prestation, commentaires divers.

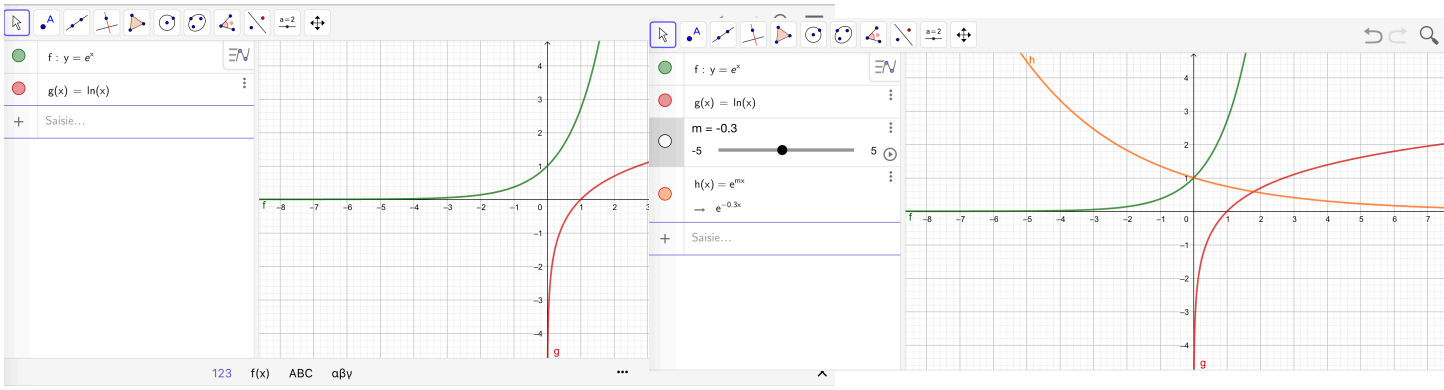
1 Notes sur le fonctionnement de Geogebra

Geogebra, en complément de Spyder et Pyzo, est installé sur les postes à disposition le jour du concours. Ce logiciel, très (très très) simple d'utilisation, vous fera gagner une part non négligeable pour les usages suivants : tracer une courbe, tracer des termes de suites, *etc.* (bien plus rapide que l'utilisation de `matplotlib.pyplot`).

✓ Tracer une courbe. On tape simplement l'équation de l'objet.

✓ Faire varier un curseur pour étudier des courbes dépendant d'un paramètre.

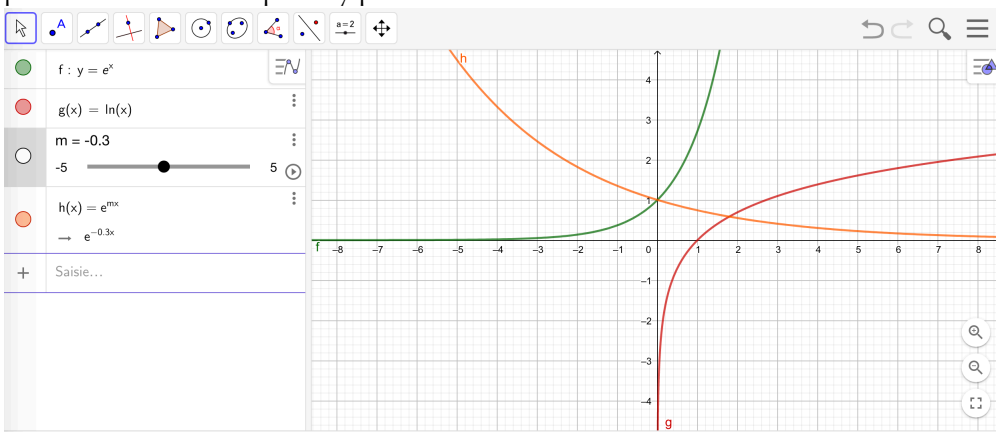
Pour le premier usage, on peut taper par exemple simplement `exp(x)`, ou encore `exp` ou encore `f(x)=exp(x)`, ... Geogebra comprendra que vous lui demandez de tracer la courbe associée. Il est aussi possible de créer des courbes dépendant d'un paramètre (voir capture de droite). Geogebra comprend qu'il s'agit d'un paramètre (les éléments différents du symbole `x`), il vous est possible de faire varier ensuite le paramètre en question.



■ **Exemple 1 —** On souhaite savoir si l'équation

$$\frac{\ln^2(x)}{x} = \frac{1}{n}$$

pour tout entier n. On peut s'y prendre ainsi :



■

2 Algèbre & Géométrie

python La plupart des commandes d'algèbre linéaire numérique (calcul du rang, calcul des valeurs propres/espaces propres) ne sont pas explicitement mentionnées dans les programmes. Les sujets vous les rappelleront toujours, mais il est bon déjà de les connaître avant de se présenter le jour de l'oral !

■ **Remarque 2.1 —** La plupart des planches proposent le type `np.array` pour la gestion des matrices. Mais il existe aussi le type `np.matrix`. Pricipale différence : les opérations sont codées autrement avec ce type.

```
>>> import numpy as np
>>> A=np.matrix([[1,2],[2,3]])
>>> B=A*A #plutôt que np.dot() pour le type array de numpy
>>> B
matrix([[ 5,  8],
        [ 8, 13]])
>>> C=np.array([[1,0],[0,2]])
>>> np.linalg.eig(C)
(array([ 1.,  2.]), array([[ 1.,  0.],
       [ 0.,  1.]]))
>>> #renvoie les éléments propres de C
>>> # d'abord les vp ensuite les vecp.

>>> #Tout est compté avec multiplicité, ce sont des listes ayant autant d'éléments que de colonnes (ou
```

```
>>> D=np.array([[1,2],[2,1]])
>>> np.linalg.eigh(D) #renvoie les éléments propres de D SI ELLE EST SYMETRIQUE =>
(array([-1.,  3.]), array([[-0.70710678,  0.70710678],
       [ 0.70710678,  0.70710678]]))
>>> #algorithme optimisé pour ce cas et uniquement celui-ci !
>>> # h comme hermitian en Anglais = symétrique lorsque la matrice est réelle
```

■

{ALG}{CCAgroVéto}{1}

[Sol 3.1]

■ Exercice 1.1 *Agro-Véto, 2018*

Soient $n \in \mathbf{N}^*$ et Φ définie sur $\mathbf{R}_n[X]$ par : $\Phi(P) = \sum_{k=0}^{n} \binom{n}{k} P^{(k)}$.

1 — 1.1. Déterminer le lien entre degré de P et degré de $P^{(k)}$.

1.2. Montrer que Φ est un endomorphisme de $\mathbf{R}_n[X]$.

2 — Soit $\Delta \in \mathcal{L}(\mathbf{R}_n[X]) : P \in \mathbf{R}_n[X] \longmapsto P'$. On note $A = (a_{i,j})_{1 \leqslant i,j \leqslant n+1}$ la matrice canoniquement associée à Φ.

2.1. Écrire la matrice D de Δ dans la base canonique de $\mathbf{R}_n[X]$.

2.2. python On rappelle que dans le module `numpy` on peut coder des matrices sous forme de tableau de listes en utilisant la commande `np.array`. *Exemple :* `np.array([[1,2,3],[7,8,9]])` code la matrice $\begin{pmatrix} 1 & 2 & 3 \\ 7 & 8 & 9 \end{pmatrix}$. La commande `np.zeros((2,3))` code la matrice $\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$. La commande `np.eye(3)` code la matrice identité I_3. La commande `np.dot(A,B)` effectue le produit de la matrice A par B sous réserve qu'elles soient compatibles et codées comme des `array`.
python Coder en langage `Python` la matrice D.

2.3. Montrer que : $A = (I_{n+1} + D)^n$.

2.4. python Écrire une fonction `Phi(coordonneesP)` qui retourne les coordonnées de $\Phi(P)$ sous forme de liste en prenant en argument les coordonnées de P sous forme de liste.

3 — 3.1. Montrer que pour tout polynôme P non nul de $\mathbf{R}_n[X]$ le degré de $\Phi(P)$ est égal à celui de P.

3.2. En déduire que Φ est injective.

3.3. Montrer que Φ est bijective.

4 — 4.1. Montrer que 1 est l'unique valeur propre de Φ et déterminer l'espace propre associé.

4.2. L'endomorphisme Φ est-il diagonalisable ? Justifier avec deux arguments différents.

{ALG}{CCAgroVéto}{2}

[Sol 3.2]

■ Exercice 1.2 *Agro-Véto, 2018, Sujet 2*

On rappelle que dans le package `numpy`*, la commande* `numpy.transpose(A)` *donne la transposée de* A*, la commande* `numpy.linalg.eig(A)` *donne les valeurs propres et les vecteurs propres éventuels de* A *et la commande* `numpy.eye(n,n)` *crée la matrice* I_n *et la commande* `numpy.ones((n,n))` [1] *crée la matrice de taille* $n \times n$ *ne comportant que des* 1.

Soit $(n,k) \in \mathbf{N}^2$ tels que $n \geq 1$ et $1 \leq k \leq n$, on considère une matrice $M = (a_{i,j}) \in \mathcal{M}_n(\mathbf{R})$ telle que $M^2 = J_n + (k-1) \times I_n$ avec :

$$J_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \quad \text{et} \quad I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

1 — python Écrire un programme Python permettant de déterminer, suivant la valeur de n, les valeurs propres de la matrice M^2, ses vecteurs propres et qui permet de vérifier les résultats obtenus. On étudiera, en particulier, le cas $n = 3$ et $k = 2$.

2 — 2.1. Déterminer, dans le cas général, le rang de J_n.

2.2. Étudier les valeurs propres éventuelles de J_n, et donner la dimension de ses sous espaces propres.

2.3. Justifier, de deux façons différentes, que J_n est diagonalisable.

3 — 3.1. Justifier que M^2 est également diagonalisable.

3.2. Déterminer les valeurs propres de M^2, et donner la dimension de ses sous espaces propres.

1. Notez bien ici le double parenthésage, ceci n'est pas une erreur ! La fonction prend en argument un objet du type `tuple`, et non du type `list`.

4 — Déterminer les valeurs propres possibles de M.

On considère un réseau social comportant n personnes, et tel que chaque couple de deux personnes distinctes ont exactement un ami en commun et que chaque personne a exactement k amis, avec $1 \leq k \leq n-1$. Une personne n'est pas amie avec elle-même. On numérote les personnes de 1 à n. On désigne par $A = (A_{i,j})_{1\leq i,j\leq n} \in \mathcal{M}_n(\mathbf{R})$, la matrice telle que $A_{i,j} = 1$ si les personnes i et j sont amies et $A_{i,j} = 0$ sinon.

5 — Déterminer un exemple de réseau vérifiant les hypothèses pour $n = 3$.

6 — Justifier que A est symétrique.
On admet, que le cœfficient $(A^2)_{i,j}$ avec $i \neq j$, donne le nombre de fois où la personne i a un ami en commun avec la personne j. On admet également que $(A^2)_{i,i}$ donne le nombre d'amis de la personne i.

7 — Donner une expression de la matrice A^2.

8 — **8.1.** Donner le nombre de couples comportant deux personnes distinctes du réseau.
8.2. Pour une personne donnée, déterminer le nombre de couples de deux personnes distinctes dont elle est un ami en commun.
8.3. En déduire la relation $k^2 - k + 1 = n$.

9 — **9.1.** En déduire les valeurs propres éventuelles de A.
9.2. On sait que $\sum_{i=1}^{n} n_i\lambda_i = 0$ où n_i est la dimension du sous espace propre associée à la valeur propre λ_i et Spec $A = \{\lambda_1, \ldots, \lambda_n\}$, montrer alors que $n = 3$.

{ALG}{CCAgroVéto}{3}

[Sol 2.3] ■ **Exercice 1.3** ***Agro—Véto, 2016, Polynômes de Tchebychev*** On considère la famille de polynômes (T_n) définie par :

$$T_0 = 1, \quad T_1 = X, \quad \forall n \in \mathbf{N}, \quad T_{n+2} = 2XT_{n+1} - T_n.$$

1 — **1.1.** Donner l'expression de T_2 et T_3.
1.2. Pour tout entier naturel $n \in \mathbf{N}$, donner le degré de T_n ainsi que l'expression du coefficient devant le terme de plus haut degré.
1.3. Pour tout entier naturel n, donner l'expression de chaque coefficient de T_{n+2} en fonction des coefficients de T_{n+1} et de T_n.

2 — **2.1.** python On décide de coder un polynôme sous forme d'une liste : celle des coefficients de ce polynôme classés dans l'ordre des degrés croissants. Écrire une fonction qui retourne le degré d'un polynôme.
2.2. python Écrire une fonction `etape(L, M)` qui donne en sortie la liste associée au polynôme $T = 2X \times M - L$ où L et M sont des listes définies comme dans la question précédente.
2.3. python Écrire une fonction `tchebychev (n)` qui, à partir d'un entier n, donne en sortie la liste associée au polynôme T_n.
2.4. Écrire une fonction d'en-tête `evalue(P,x)` qui, étant donné un polynôme P (défini sous forme d'une liste) et un réel x, évalue ce polynôme en x c'est à dire donne une valeur pour $P(x)$. On essayera de concevoir un algorithme utilisant le moins d'opérations algébriques possible (on se souviendra par exemple de l'algorithme de Hörner vu en TP).
2.5. Écrire une fonction d'en-tête `def TraceTchebychev(n)` qui à partir d'un entier n, trace la représentation graphique du polynôme T n sur l'intervalle $[-1, 1]$. Quelles observations peut-on faire ?

3 — **3.1.** Linéariser $\cos a \cos b$ pour $(a, b) \in \mathbf{R}^2$.
3.2. Pour tout réel a et $n \in \mathbf{N}$, prouver que : $T_n(\cos a) = \cos(na)$.
3.3. Pour tout entier naturel $n \in \mathbf{N}$, montrer que le polynôme T_n admet n racines distinctes, toutes dans $[-1, 1]$.

{ALG}{CCAgroVéto}{4}

[Sol 2.4] ■ **Exercice 1.4** ***Agro—Véto, 2018, Polynômes interpolateurs de Lagrange*** Soient $x_0, x_1, \ldots, x_n$ des réels deux à deux distincts. Soit ϕ l'application définie par :

$$\phi \left| \begin{array}{ccc} \mathbf{R}_n[X] & \longrightarrow & \mathbf{R}^{n+1} \\ Q & \longmapsto & (Q(x_0), Q(x_1), \ldots, Q(x_n)) \end{array} \right. .$$

1 — Montrer que ϕ est un isomorphisme de $\mathbf{R}_n[X]$ sur $\mathbf{R}^{n+1}$.

2 — **2.1.** Écrire la matrice M de ϕ dans les bases canoniques de $\mathbf{R}_n[X]$ et $\mathbf{R}^{n+1}$.
2.2. La matrice M est-elle inversible ?

Pour tout $i \in [\![0\,,\,n]\!]$, on pose

$$L_i(X) = \prod_{\substack{0\leqslant j\leqslant n\\ j\neq i}} \frac{X - x_j}{x_i - x_j}$$

(les L_i sont appelés *polynômes interpolateurs de Lagrange*).

3 — **3.1.** Montrer que $L_i \in \mathbf{R}_n[X]$.
3.2. Calculer $L_i(x_j)$ pour tout $j \in [\![0; n]\!]$. En déduire $\phi(L_i)$.
3.3. Montrer que la famille $\mathbf{B} = (L_0, L_1, \dots, L_n)$ est une base de $\mathbf{R}_n[X]$.
3.4. Écrire la matrice de ϕ dans la base $\mathbf{B}$ et la base canonique de $\mathbf{R}^{n+1}$.
3.5. python Écrire une fonction python qui décrit le polynôme L_i.

4 — Soit X une variable aléatoire discrète à valeurs dans $\{x_0, x_1, \dots, x_n\}$. On note V la matrice colonne définie par $\begin{pmatrix} \mathbf{P}(X = x_0) \\ \mathbf{P}(X = x_1) \\ \vdots \\ \mathbf{P}(X = x_n) \end{pmatrix}$.
4.1. Calculer le produit ${}^{\mathsf{T}}V \times M$ en fonction des moments d'ordre r de X ($0 \leq r \leq n$).
4.2. Supposons $\mathbf{E}(X), \mathbf{E}(X^2), \dots, \mathbf{E}(X^n)$ connus. Comment remonter à la loi de X ?
4.3. On suppose le module `numpy` importé. Les commandes suivantes sont rappelées :

- ✓ `import numpy as np` pour importer le module `numpy` ;
- ✓ `np.dot(A,B)` renvoie le produit de deux matrices A et B ;
- ✓ `np.linalg.inv(A)` renvoie l'inverse d'une matrice inversible A.

Créer un programme qui renvoie la loi de X sous forme de vecteur, connaissant $\mathbf{E}(X), \mathbf{E}(X^2), \dots, \mathbf{E}(X^n)$.

{ALG}{CCAgroVéto}{5}

[Sol 2.5] ■ **Exercice 1.5** ***Agro—Véto, Sujet 1, 2018*** On considère E l'ensemble des matrices de $\mathcal{M}_3(\mathbf{R})$ admettant le vecteur $U = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$ pour vecteur propre et l'ensemble :

$$F = \left\{ \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & a \end{pmatrix}, (a, b, c, d, e) \in \mathbf{R}^5 \right\}.$$

On pose $A = \begin{pmatrix} 1 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$.

1 — python À l'aide d'un programme Python, déterminer la plus petite valeur propre parmi les matrices de F dont les cœfficients sont égaux à 0 ou 1. *On pourra par exemple utiliser la fonction `numpy.linalg.eig`, comme le montre l'exemple suivant :*

```
import numpy.linalg as la
vap,vep = la.eig ( [ [1 ,2 ], [3 ,4]] )
```

Après cette suite d'instructions, la variable `vap` contient la liste des valeurs propres de la matrice $\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}$ et la variable `vep` est une matrice dont les colonnes sont des vecteurs propres de cette matrice.

2 — **2.1.** Montrer que E et F sont des sous-espaces vectoriels de $\mathcal{M}_3(\mathbf{R})$.
2.2. Donner une base de $E \cap F$.

3 — **3.1.** Montrer que $A \in E \cap F$.

3.2. Montrer que A est diagonalisable dans une base orthonormale de vecteurs propres et déterminer une matrice P inversible et une matrice D diagonale vérifiant $A = {}^{T}PDP$ où ${}^{T}P$ est la matrice transposée de P.

4 — Vérifier que ${}^{T}PMP$ est diagonale pour toute matrice M de $E \cap F$.

5 — Soit $(x, y, z) \in \mathbf{R}^3$. Déterminer le spectre de $M = \begin{pmatrix} y+z & y & x \\ y & x+z & y \\ x & y & y+z \end{pmatrix}$.

{ALG}{CCAgroVéto}{6}

[Sol 2.6]

Exercice 1.6 *Agro—Véto, 2016*

On considère $\mathbf{R}^4$ muni du produit scalaire usuel. On s'intéresse à l'évolution au cours du temps discrétisé de quatre manières différentes (en i) au moyen des suites $(a_i(n))_{n\geqslant 0}$ pour $i \in [\![1\,,\,4]\!]$ vérifiant :

$$\forall n \in \mathbf{N}, \quad \forall i \in [\![1\,,\,4]\!], \quad a_i(n+1) = \sum_{j=1}^{4} m_{i,j}\, a_j(n),$$

où $M = (m_{i,j})_{(i,j)\in[\![1\,,\,4]\!]^2} \in \mathcal{M}_4(\mathbf{R})$ est une matrice symétrique réelle carrée de format 4×4. On identifiera les matrices colonnes avec les n-uplets.

1 — Montrer qu'il existe une base orthonormée de $\mathbf{R}^4$, notée $\mathcal{B} = (E_1, E_2, E_3, E_4)$ et $(\lambda_1, \ldots, \lambda_4) \in \mathbf{R}^4$ tels que : $\forall k \in [\![1\,,\,4]\!], \quad ME_k = \lambda_k E_k$.
On suppose dans la suite que $|\lambda_1| \geqslant \cdots \geqslant |\lambda_4|$.

2 — On pose $X_n = \begin{pmatrix} a_1(n) \\ \vdots \\ a_4(n) \end{pmatrix}$. Les coordonnées de X_0 dans la base $\mathcal{B}$ sont notées $y_1, \ldots, y_4$. Montrer que :

$$\forall n \in \mathbf{N}, \quad X_n = \sum_{k=1}^{4} y_k \lambda_k^n E_k.$$

3 — Quelle relation matricielle existe-t-il entre les $(a_k(0))_{k\in[\![1\,,\,4]\!]}$ et les $(y_k)_{k\in[\![1\,,\,4]\!]}$?

4 — python On étudie à l'aide de l'outil informatique l'exemple suivant :

$$G = \begin{pmatrix} 64 & 0 & 0 & 0 \\ 0 & 64 & 22 & 28 \\ 0 & 22 & -28 & -22 \\ 0 & 28 & -22 & 12 \end{pmatrix}, \quad M = \frac{1}{64} G.$$

Dans l'environnement python on fera l'importation `import numpy.linalg as lin` pour le module d'algèbre linéaire. On rappelle que `lin.eig(A)` renvoie le 2-tuple formé du vecteur des valeurs propres et de la matrice des coordonnées des vecteurs propres de la matrice A. On rappelle aussi que `lin.inv(A)` retourne l'inverse de la matrice A.

4.1. Sous l'environnement python, faire afficher les valeurs propres et les vecteurs E_k pour $k \in [\![1\,,\,4]\!]$ associés à la matrice M.

4.2. On prend $(a_1(0), a_2(0), a_3(0), a_4(0)) = (2, 2, 4, 5)$. Calculer informatiquement les y_k pour $k \in [\![1\,,\,4]\!]$.

5 — python Écrire une fonction Python `a` prenant en argument un entier n et renvoyant en sortie le vecteur $\begin{pmatrix} a_1(n) \\ \vdots \\ a_4(n) \end{pmatrix}$.

{ALG}{CCAgroVéto}{7}

[Sol 2.7]

Exercice 1.7 *Agro—Véto,2017*

1 — python Soient $a_1, \ldots, a_n \in \mathbf{R}^\star$. Écrire une fonction qui renvoie `True` si $a_i^2 \sum_{k=1}^{n} \frac{1}{a_k^2} \geqslant 2$ pour tout $i \in \{1, \ldots, n\}$ et qui renvoie `False` sinon. La fonction aura pour seul paramètre une liste contenant les réels $a_1, \ldots, a_n$.

2 — On considère les vecteurs $u = \begin{pmatrix}1 & 0 & 0\end{pmatrix}$, $v = \sqrt{2}\begin{pmatrix}0 & 1 & 0\end{pmatrix}$, et $w = \sqrt{2}\begin{pmatrix}0 & 0 & 1\end{pmatrix}$, et $\mathcal{P}$ le plan de $\mathbf{R}^3$ admettant pour équation dans la base canonique :

$$y - z = 0.$$

Déterminer les projetés orthogonaux des vecteurs u, v, w sur le plan $\mathcal{P}$ et vérifier qu'ils ont même norme.

3 — Soit (e_1, e_2, e_3) la base canonique de $\mathbf{R}^3$ et $a_1, a_2, a_3 \in \mathbf{R}^\star$. On suppose qu'il existe un plan $\mathcal{P}$ tel que les projetés orthogonaux des vecteurs a_1e_1, a_2e_2 et a_3e_3 sur ce plan aient tous la même norme notée d. On considère $(\varepsilon_1, \varepsilon_2)$ une base orthonormée de $\mathcal{P}$ et ε_3 un vecteur normal à $\mathcal{P}$ de norme 1. On note p la projection orthogonale sur le plan $\mathcal{P}$.

3.1. Donner une expression de $p(e_i)$ pour $i \in \{1,2,3\}$ à l'aide des vecteurs ε_1 et ε_2.

3.2. Montrer que pour tout $i \in \{1,2,3\}$, on a : $\langle e_i|\varepsilon_1\rangle^2 + \langle e_i|\varepsilon_2\rangle^2 = \left(\frac{d}{a_i}\right)^2$.

3.3. Montrer que : $\displaystyle\sum_{i=1}^{3} \frac{1}{a_i^2} = \frac{2}{d^2}$.

3.4. Pour $i \in \{1,2,3\}$, montrer que $|a_i| \geqslant d$ puis que :

$$a_i^2 \sum_{k=1}^{3} \frac{1}{a_k^2} \geqslant 2.$$

3 Analyse

{AN}{CCAgroVéto}{1}

[Sol 2.8] ■ **Exercice 1.8** ***Agro—Véto, Sujet 7, 2017*** Soit $(u_n)_{n\in\mathbf{N}}$ une suite définie par la donnée de ses trois premiers termes u_0, u_1, u_2 et par la relation de récurrence :

$$\forall n \in \mathbf{N}, \quad u_{n+3} = \frac{1}{3}\left(u_n + u_{n+1} + u_{n+2}\right).$$

On note $\mathrm{A} = \begin{pmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ \frac{1}{3} & \frac{1}{3} & \frac{1}{3}\end{pmatrix}$, et pour tout entier $n \in \mathbf{N}$, on pose : $\mathrm{X}_n = \begin{pmatrix}u_n\\ u_{n+1}\\ u_{n+2}\end{pmatrix}$.

1 — python Écrire une fonction Python prenant en argument les trois premiers termes de la suite et renvoyant la liste de ses 100 premiers termes. Utiliser cette fonction pour étudier le comportement asymptotique de la suite sur quelques exemples.

2 — Démontrer que $0 \notin \mathrm{Spec\,A}$.

3 — Démontrer que pour tout complexe λ, nous avons : $\lambda \in \mathrm{Spec\,A}$ si et seulement si λ racine de $\mathrm{X}^3 - \frac{1}{3}\mathrm{X}^2 - \frac{1}{3}\mathrm{X} - \frac{1}{3}$.

4 — Démontrer qu'il existe $\mathrm{P} \in \mathcal{GL}_3(\mathbf{C})$, et z un complexe tel $|z| < 1$ tels que : $\mathrm{A} = \mathrm{P}\begin{pmatrix}1 & 0 & 0\\ 0 & z & 0\\ 0 & 0 & \overline{z}\end{pmatrix}\mathrm{P}^{-1}$.

5 — Pour tout entier $n \in \mathbf{N}$, exprimer X_{n+1} en fonction de A et X_n et en déduire une expression en fonction de n, A, X_0.

6 — Démontrer alors qu'il existe trois complexes a, b, c (que l'on ne demande pas d'expliciter) tels que :

$$\forall n \in \mathbf{N}, \quad u_n = a + bz^n + c\overline{z}^n.$$

7 — Démontrer que $\lim_{n\to\infty} \left|bz^n + c\overline{z}^n\right| = 0$. Que peut-on dire de $\lim_{n\to\infty} \mathrm{Re}\left(bz^n + c\overline{z}^n\right)$ et de $\lim_{n\to\infty} \mathrm{Im}\left(bz^n + c\overline{z}^n\right)$?

8 — En déduire que $a \in \mathbf{R}$ et que : $\lim_{n\to\infty} u_n = a$.[4]

4. On pourrait croire que cette question sort un peu du programme étant donné que la convergence des suites complexes (*i.e.*celle des

{AN}{CCAgroVéto}{2}

[Sol 2.9] ■ **Exercice 1.9** ***Agro—Véto, Sujet 5, 2018. Étude de la convergence de deux produits infinis.*** Soit x un réel de l'intervalle $[0;1[$ fixé. On définit les suites $(f_n(x))_{n\geq 1}$, $(g_n(x))_{n\geq 1}$ et $(h_n(x))_{n\geq 1}$ par :

$$f_n(x)=\prod_{k=1}^{n}(1+x^k),\quad g_n(x)=\prod_{k=1}^{n}(1-x^{2k-1}),\quad h_n(x)=f_n(x)g_n(x).$$

On pose, sous réserve d'existence, $f(x)=\lim\limits_{n\to\infty} f_n(x)$, $g(x)=\lim\limits_{n\to\infty} g_n(x)$ et $h(x)=f(x)g(x)$.

1 — Écrire un script Python qui affiche dans un repère les points de coordonnées $(f_n(x);g_n(x))$ lorsque x prend les valeurs $\frac{k}{100}$ avec $k\in\{0;\cdots;80\}$ et $n=100$. Faire une conjecture d'une relation simple entre $f(x)$ et $g(x)$ en admettant leurs existences.

2 — Montrer que pour tout $x\in[0;1[$, la suite $(f_n(x))_{n\geq 1}$ est croissante et que la suite $(g_n(x))_{n\geq 1}$ est décroissante.

3 — **3.1.** Établir que :

$$\forall t\in\mathbf{R},\quad 1+t\leq \mathrm{e}^t.$$

En déduire que, pour tout $x\in[0;1[$, $f(x)$ existe et vérifie :

$$1\leq f(x)\leq \exp\frac{x}{1-x}.$$

3.2. Montrer que f est continue en 0.

4 — **4.1.** Justifier l'existence de $g(x)$ pour tout $x\in[0;1[$.

4.2. Montrer que pour tout $t\in[0;1[$ et $x\in[0;1[$, $1-(1-x)^t\geq xt$. *(on pourra étudier une fonction de x ou utiliser la formule des accroissements finis.)*

4.3. En déduire l'encadrement suivant, pour tout $x\in[0;1[$:

$$\exp\left(\frac{\ln(1-x)}{1-x^2}\right)\leq g(x)\leq \exp\left(-\frac{x}{1-x^2}\right),$$

puis la continuité de g en 0.

5 — **5.1.** Montrer que, pour tout $x\in[0;1[$: $f_n(x^2)g_n(x^2)=f_{2n}(x)g_n(x)$. En déduire que $h(x^2)=h(x)$.

5.2. Montrer que pour tout $n\geq 1$, on a : $h(x^{2^n})=h(x)$. Conclure alors que pour tout $x\in[0;1[$, $h(x)=1$.

5.3. Ce dernier résultat confirme-t-il votre conjecture ?

{AN}{CCAgroVéto}{3}

[Sol 2.10] ■ **Exercice 1.10** ***Agro—Véto, Sujet 4, 2018*** Dans ce problème, on s'intéresse à l'équation suivante :

$$(\mathrm{E}_n):\quad \frac{\ln^2(x)}{x}=\frac{1}{n},$$

où n est un entier strictement positif, et x, l'inconnue, est un nombre réel strictement positif. Soit f la fonction définie sur $[1;+\infty[$ par :

$$\forall x\in[1;+\infty[,\quad f(x)=\frac{\ln^2(x)}{x}.$$

1 — **1.1.** Dresser le tableau de variations de f sur son ensemble de définition.

1.2. En déduire que l'équation (E_1) n'admet pas de solution.

1.3. Démontrer que, pour $n\geq 2$, l'équation (E_n) admet deux solutions, que l'on notera α_n et β_n, telles que :

$$1\leq \alpha_n\leq e^2\leq \beta_n.$$

parties réelles et imaginaires vers une limite finie) ne l'est pas. En fait ce n'est pas le cas (*c.f.*correction) notamment car (u_n) est une suite réelle (récurrence immédiate).

2 — À l'aide de l'outil informatique, représenter sur un même graphe la courbe représentative de f ainsi que les droites D_i, $1 \leq i \leq 6$, où D_i a pour équation $y = \frac{1}{i}$, pour $i \in [\![1\,,\,6]\!]$.

3 — Quelle conjecture peut-on émettre sur le sens de variations et sur les limites des suites $(\alpha_n)_{n\geq 2}$ et $(\beta_n)_{n\geq 2}$?

4 — On va étudier la suite $(\beta_n)_{n\geq 2}$ dans cette question.

4.1. Démontrer que la suite $(\beta_n)_{n\geq 2}$ est strictement monotone.

4.2. Montrer que la suite $(\beta_n)_{n\geq 2}$ admet une limite que l'on précisera.

4.3. Soit la suite $(u_n)_{n\geq 2}$ définie par $u_n = \frac{\beta_n}{n}$. On admet que $\ln(u_n) \underset{n\to\infty}{=} o(\ln(n))$. Prouver alors que $u_n \underset{n\to\infty}{\sim} \ln^2(n)$.

4.4. En déduire un équivalent de $(\beta_n)_{n\geq 2}$.

5 — On s'intéresse dans cette question à la suite $(\alpha_n)_{n\geq 2}$.

5.1. Montrer que la suite $(\alpha_n)_{n\geq 2}$ admet une limite que l'on précisera.

5.2. Donner un équivalent de $\alpha_n - 1$ lorsque n tend vers $+\infty$. Comment pourrait-on vérifier ce résultat avec l'outil informatique ?

{AN}{CCAgroVéto}{4}

[Sol 2.11] ■ **Exercice 1.11** ***Agro—Véto, 2016, Intégration numérique*** Pour tout réel x et tout entier $n \in \mathbf{N}^\star$, on pose :

$$f(x) = \int_0^\pi e^{-x\sin t}\,dt, \quad S_n(x) = \frac{1}{n}\sum_{k=0}^{n-1} e^{-x\sin\left(\frac{k\pi}{n}\right)}.$$

1 — python

1.1. Écrire une fonction python `Sn` qui prend en entrée des valeurs de x et n et donne en sortie la valeur de $S_n(x)$.

1.2. Écrire une fonction python `graphe` qui trace le graphe de la fonction S_n sur l'intervalle $[a, b]$, les valeurs de a, b et n étant données en paramètres.

2 — Pour tout entier $n \in \mathbf{N}^\star$, montrer que S_n est dérivable en zéro, et exprimer $S_n'(0)$ sans symbole Σ.

3 — Pour tout x, montrer que la suite $(S_n(x))_{n\in\mathbf{N}^\star}$ est convergente, quelle est sa limite ?

4 — ***(À propos de la fonction f)***

4.1. Soient $x \leqslant y$ deux réels, comparer $f(x)$ et $f(y)$, qu'en conclure ?

4.2. Montrer que : $\forall x \in \mathbf{R}, \quad f(x) = 2\int_0^{\frac{\pi}{2}} e^{-x\sin t}\,dt$.

4.3. Montrer que : $\forall t \in [0, \pi/2], \quad \sin t \geqslant \frac{2}{\pi}t$, en déduire la limite de f en $+\infty$.

{AN}{CCAgroVéto}{5}

[Sol 2.12] ■ **Exercice 1.12** ***Agro—Véto, 2015, Méthode numérique pour une équation différentielle*** On considère l'équation différentielle suivante, dans laquelle y désigne la fonction inconnue de la variable x

$$xy'(x) + 2y(x) = \frac{x^2}{1+x^2}. \tag{E}$$

1 — ***(Résolution de (E))***

1.1. Décrire l'ensemble des solutions de (E) définies sur $]-\infty; 0[\cup]0; \infty[$. *Indication : On pourra remarquer l'égalité $x^3 = x(1+x^2) - x$ pour tout $x \in \mathbf{R}$.*

1.2. On appelle solution sur $\mathbf{R}$, toute solution de (E) définie et dérivable sur $\mathbf{R}$. L'équation (E) admet-elle de telles solutions ? Si oui, exprimer ces solutions, et positionner localement leur courbe représentative par rapport à leur tangente en zéro.

1.3. Résoudre (E) avec la condition de Cauchy $y(1) = \frac{1}{2}$.

1.4. python Écrire une fonction Python nommée `solexacte` prenant en paramètre $b \in \mathbf{R}$ et traçant la solution sur un intervalle $[1, b]$.

2 — ***(Méthode d'Euler)*** Écrire une fonction Python nommée `soleuler` prenant en paramètre $N \in \mathbf{N}$ et qui renvoie les listes des valeurs $(x_i)_{i\in[\![0\,,\,N]\!]}$ et $(y_i)_{i\in[\![0\,,\,N]\!]}$ et qui trace les segments reliant les points (x_i, y_i) et (x_{i+1}, y_{i+1}) pour $i \in [\![0\,,\,N-1]\!]$. Comparer cette courbe à celle de la solution exacte.

{AN}{CCAgroVéto}{6}

[Sol 2.13] ■ **Exercice 1.13** ***Agro—Véto, 2018***

Rappel : algorithme de dichotomie. *On considère une fonction g continue sur un intervalle $[a;b]$. 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* $c_k = \frac{a_k+b_k}{2}$ *et :*

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

$$\text{sinon } a_{k+1} = c_k \text{ et } b_{k+1} = b_k$$

On sait alors que les suites $(a_k)_{k\geq 0}$ et $(b_k)_{k\geq 0}$ convergent toutes les deux vers α.

Pour $n \geqslant 1$, on définit la suite de fonctions (f_n) par :

$$\forall x \in \mathbf{R} \quad f_n(x) = nx^3 + n^2x - 2.$$

1 — 1.1. Montrer pour n fixé que l'équation $f_n(x) = 0$ admet une unique solution sur $\mathbf{R}$ notée a_n. Montrer que $a_n > 0$.

1.2. python Écrire une fonction `dicho(n)` qui calcule une valeur approchée de a_n pour un n donné, à précision 10^{-2}, en utilisant le principe de dichotomie. La tester pour $n = 2$.

1.3. Montrer que la suite $(a_n)_{n\geqslant 1}$ est convergente et déterminer sa limite.

2 — On pose $u_n = n^2a_n$ pour tout $n \geqslant 1$.

2.1. python Écrire un programme qui trace les termes u_n pour $n \in [\![10, 40]\!]$. En déduire une conjecture sur la limite de la suite (u_n).

2.2. Démontrer cette conjecture. En déduire un équivalent de la suite (a_n).

3 — Soit g définie sur $[0,1]$ par $g(x) = \frac{2x^3+1}{3x^2+2}$ pour tout $x \in [0,1]$.

3.1. Montrer que g est croissante sur $[a_2, 1]$.

3.2. On définit la suite $(x_n)_{n\in\mathbf{N}}$ par $x_0 = 1$ et $x_{n+1} = g(x_n)$ pour tout entier $n \in \mathbf{N}$. Montrer que la suite (x_n) converge et déterminer sa limite.

{AN}{CCAgroVéto}{7}

[Sol 2.14] ■ **Exercice 1.14** ***Agro—Véto, 2018*** Soit $(u_n)_{n\in\mathbf{N}^*}$ une suite décroissante, convergente vers 0.

On définit pour tout $n \in \mathbf{N}^*$, $S_n = \sum_{k=1}^{n} (-1)^k u_k$

1 — 1.1. python Écrire une fonction permettant de représenter les termes de la suite (S_n) et conjecturer. On pourra commencer la fonction par le préambule minimal suivant :

```
import numpy as np
import matplotlib.pyplot as plt
```

et utiliser a commande `plt.plot(x,y,"o")`.

1.2. Montrer que les deux suites (S_{2n}) et (S_{2n+1}) sont adjacentes.

1.3. En déduire que la suite (S_n) converge vers une limite ℓ.

1.4. Justifier que $S_{2n+1} \leq \ell \leq S_{2n}$ pour tout $n \in \mathbf{N}^*$.

2 — On pose $u_k = \frac{1}{k}$ pour tout $k \in \mathbf{N}^*$.

2.1. Montrer que $\lim_{n\to\infty} \int_0^1 \frac{x^n}{1+x}\,dx = 0$.

2.2. Montrer que pour tout entier naturel b : $\lim_{n\to\infty} \sum_{k=0}^{n-1} \frac{(-1)^k}{k+b+1} = \int_0^1 \frac{x^b}{1+x}\,dx$.

2.3. Montrer que $\lim_{n\to\infty} \left(\sum_{k=1}^{n} \frac{(-1)^k}{k}\right) = -\ln(2)$.

2.4. python À l'aide des questions 1. et 2., écrire une fonction Python qui donne une approximation de $\ln(2)$ à 10^{-3} près.

4 Probabilités & Statistiques

4.0.1. Aléatoire discret

Dans les exercices qui suivent, il est nécessaire de bien revoir l'intérêt de la loi faible des grands nombres dans l'approximation de probabilités et d'espérances.

{PS}{CCAgroVéto}{6}

[Sol 2.15] ■ **Exercice 1.15** ***Agro—Véto, Sujet 8, 2018*** Une urne contient initialement deux boules blanches et deux boules noires. Soit c un entier naturel. On effectue une série de tirages en suivant le protocole suivant :

- ✓ On tire au hasard une première boule. Si elle est blanche, on arrête Là. Si elle est noire, on remet la boule noire dans l'urne. Puis on rajoute encore c boules noires dans l'urne.
- ✓ On recommence ainsi jusqu'à obtenir une boule blanche (si on finit par obtenir une boule blanche), ou indéfiniment si on n'obtient jamais de boule blanche.

Pour tout entier naturel n non nul, on note E_n l'événement : « Les n premiers tirages ont eu lieu et n'ont donné que des boules noires ». Soit X la variable aléatoire égale au rang du tirage auquel on a obtenu une boule blanche si on finit par obtenir une boule blanche et égale à 0 sinon.

1 — Que dire de la loi de X si $c = 0$? Calculer $\mathbf{P}(X = 3)$ en fonction de c pour c quelconque.

2 — python

2.1. Écrire une fonction en langage Python qui prend en argument la valeur de c et un entier naturel s. Cette fonction doit simuler l'expérience ci-dessus, avec un nombre maximal de tirages égal à s. Elle doit renvoyer le rang d'apparition d'une boule blanche si une boule blanche a été obtenue et 0 sinon.

2.2. Utiliser la fonction précédente pour simuler un grand nombre de fois l'expérience pour donner une estimation de $\mathbf{P}(X = 0)$ pour $c = 1$, $c = 2$ et $c = 5$.

3 — Démontrer que : $\forall n \in \mathbf{N}^\star, \quad \mathbf{P}(E_n) = \prod_{k=0}^{n-1} \frac{2+kc}{4+kc}$.

4 — On suppose dans cette question que $c = 1$.

4.1. Calculer $\mathbf{P}(E_n)$ pour tout entier naturel n non nul. En déduire la valeur de $\mathbf{P}(X = 0)$.

4.2. Démontrer que : $\forall n \in \mathbf{N}^\star, \ \mathbf{P}(X = n) = \frac{12}{(n+1)(n+2)(n+3)}$.

4.3. En utilisant le théorème du transfert, démontrer que la variable aléatoire $X + 3$ admet une espérance, et calculer cette espérance. En déduire l'espérance de X.

4.4. python Utiliser la fonction de la question 2.1 pour vérifier ce résultat à l'aide de simulations.

5 — On suppose dans cette question que $c = 2$.

5.1. Calculer $\mathbf{P}(E_n)$ pour tout entier naturel n non nul. En déduire la valeur de $\mathbf{P}(X = 0)$.

5.2. Donner la loi de X. La variable X admet-elle une espérance ?

6 — Dans cette question, c est un entier naturel non nul quelconque.

6.1. Démontrer que : $\forall n \in \mathbf{N}^\star, \ -\ln(\mathbf{P}(E_n)) = \sum_{k=0}^{n-1} \ln(1 + \frac{2}{2+kc})$.

6.2. Déterminer alors la valeur de $\mathbf{P}(X = 0)$. *On pourra pour cela utiliser sans démonstration le résultat suivant : Si (u_n) et (v_n) sont deux suites positives et si $u_n \underset{n\to\infty}{\sim} v_n$, alors $\sum_{n\geq 0} u_n$ et $\sum_{n\geq 0} v_n$ ont même nature.*

{PS}{CCAgroVéto}{7}

[Sol 1.16] ■ **Exercice 1.16** ***Agro—Véto, 2017*** Une urne contient n boules numérotées de 1 à n. On tire aléatoirement une par une et sans remise les n boules. À chaque tirage, on définit le « record » comme étant le plus grand numéro obtenu jusque là.

Pour chaque entier $k \in [\![1\,,\, n]\!]$, on note $Y_{k,n}$ la variable aléatoire réelle qui vaut 1 si le k-ième tirage est un record (autrement dit si le k-ième numéro tiré est supérieur aux précédents) et 0 sinon. Par convention le premier tirage sera considéré comme un record, *i.e.* $Y_{1,n} = 1$.

1 — **1.1.** python Écrire une fonction `tirage(n)` qui simule le tirage des n boules.
Indication : *La fonction* `shuffle` *du module* `random` *permet de mélanger aléatoirement en place les éléments d'une liste.*

1.2. Écrire une fonction Python `maximum(L)` qui retourne le maximum d'une liste L passée en argument.

1.3. Écrire une fonction Python `simuleY(k,n)` qui simule $Y_{k,n}$.

1.4. Que retourne la fonction `f(k,n)` suivante ? Que permet-elle d'évaluer ?

```
def f(k,n):
    N = 10000
    li = [simuleY(k,n) for i in range(N)]
    return sum(li)/N
```

1.5. En choisissant différentes valeurs de k, conjecturer au sujet de $\mathbf{P}(Y_{k,n} = 1)$. Démontrer le résultat précédent.

2 — On note B_n le nombre de fois où l'on a obtenu un record au cours des n tirages.

2.1. Exprimer B_n en fonction des $Y_{k,n}$, $1 \leq k \leq n$. En déduire $\mathbf{E}(B_n)$.

2.2. Calculer $\mathbf{P}(B_n = 1)$ et $\mathbf{P}(B_n = n)$.

2.3. Montrer que pour tout $k \geq 2$:

$$\int_k^{k+1} \frac{1}{t}\,\mathrm{d}t \leq \frac{1}{k} \leq \int_{k-1}^{k} \frac{1}{t}\,\mathrm{d}t.$$

En déduire un équivalent de $\mathbf{E}(B_n)$ lorsque n tend vers $+\infty$.

3 — Pour $i \neq j$, calculer $\mathbf{P}[(Y_{i,n} = 1) \cap (Y_{j,n} = 1)]$. Les variables aléatoires réelles $Y_{i,n}$ et $Y_{j,n}$ sont-elles indépendantes ?

{PS}{CCAgroVéto}{8}

[Sol 1.17]

Exercice 1.17 *Agro—Véto, Sujet 9, 2018*

On s'intéresse à l'évolution d'une population de bactéries procaryotes dans un écosystème donné répondant au modèle suivant. L'évolution est supposée réalisée par étapes successives, suivant chacune le même fonctionnement ; à chaque étape donnée, chaque bactérie, indépendamment des autres peut :

- ✓ soit donner lieu à une fission binaire, et se diviser en deux bactéries identiques indépendantes, ceci avec une probabilité $\frac{2}{3}$;
- ✓ soit mourir et se désintégrer avec une probabilité $\frac{1}{3}$.

On appelle X_n la variable aléatoire égale au nombre de bactéries présentes après la n-ième étape. Au départ, il n'y a qu'une seule bactérie dans l'écosystème, et on note $X_0 = 1$.

1 — Donner la loi et l'espérance de X_1.

2 — **2.1.** Pour $n \geq 1$, justifier que X_n ne prend que des valeurs paires. Expliciter $X_n(\Omega)$.

2.2. python Écrire un programme informatique prenant en argument la valeur de n, et retournant les valeurs d'une simulation de $(X_1; X_2; \cdots; X_n)$.

2.3. Soit i tel que $2i \in X_n(\Omega)$. Déterminer la loi conditionnelle de X_{n+1} sachant $[X_n = 2i]$.

3 — Soit $n \in \mathbf{N}$. On définit la fonction G_n par :

$$\forall x \in \mathbf{R},\ G_n(x) = \sum_{k \in X_n(\Omega)} x^k \mathrm{P}(X_n = k) \quad \textit{(avec } 0^0 = 1 \textit{ par convention)}$$

et on admet que :

$$\forall n \in \mathbf{N},\ G_{n+1} = G_n \circ G_1 = G_1 \circ G_n.$$

3.1. Donner les valeurs de $G_n(1)$ et $G_n'(1)$.

3.2. En déduire une relation entre $\mathbf{E}(X_{n+1})$ et $\mathbf{E}(X_n)$.

3.3. Calculer alors l'espérance de X_n en fonction de n.

4 — On note $\forall n \in \mathbf{N}^\star,\ u_n = \mathbf{P}(X_n = 0)$, et soit R l'événement « La population de bactéries finit par s'éteindre ».

4.1. Montrer que : $\forall n \in \mathbf{N}^\star,\ \forall x \in \mathbf{R},\ G_{n+1}(x) = \frac{1}{3} + \frac{2}{3}(G_n(x))^2$.

4.2. En déduire que :

$$\forall n \in \mathbf{N}^\star,\quad u_{n+1} = \frac{1}{3} + \frac{2}{3}u_n^2.$$

4.3. Montrer que : $\forall n \in \mathbf{N}^\star, \quad 0 \leq u_n \leq \frac{1}{2}$. En déduire que la suite (u_n) converge vers un réel à déterminer.

5 — On note $\forall n \in \mathbf{N}$, D_n l'événement « la population disparaît exactement à l'issue de l'étape n ».
5.1. Montrer que $\forall n \in \mathbf{N}^\star, \quad \mathbf{P}(D_n) = u_n - u_{n-1}$.
5.2. En remarquant que $R = \cup_{n=1}^{+\infty} D_n$, déterminer alors la probabilité que la population de bactéries s'éteigne.

{PS}{CCAgroVéto}{9}

[Sol 1.18] ■ **Exercice 1.18** ***Agro—Véto, 2018*** On rappelle que la fonction `random` de la bibliothèque `random` renvoie un flottant entre 0 et 1. La fonction définie ci-dessous permet de représenter graphiquement la loi d'une variable aléatoire X_n à valeurs dans $[\![0\,,\,n]\!]$ pour $n \in \mathbf{N}$, la quantité `loi` étant la liste

$$[\mathbf{P}(X_n = 0), \mathbf{P}(X_n = 1), ..., \mathbf{P}(X_n = n)].$$

```
import matplotlib.pyplot as plt
def graphe(loi):
    lx=[i for i in range(len(loi))]
    plt.plot(lx,loi)
```

On admettra sans la démontrer la propriété suivante :

$$\forall n \in \mathbf{N}, \quad \forall p \in \mathbf{N}, \quad \sum_{k=0}^{p} \binom{n+k}{n} = \binom{n+p+1}{n+1}.$$

Un tournoi oppose deux adversaires, que nous nommerons A et B, dans une succession de n combats. Initialement, chaque combattant dispose d'un capital de points, que nous noterons α_0 et β_0. À l'issue d'un combat, le capital du vainqueur augmente de 1, celui du vaincu ne change pas. Si α_n et β_n désignent les capitaux respectifs des deux joueurs à l'issue du n-ième combat, et si A_n (*resp.* B_n) désigne l'événement « A (*resp.* B) remporte le n-ième combat », nous supposerons que la probabilité que le joueur A (*resp.* B) gagne le $(n+1)$-ième combat est

$$\mathbf{P}(A_{n+1} | [\alpha_n = a] \cap [\beta_n = b]) = \frac{a}{a+b} \quad resp. \quad \mathbf{P}(B_{n+1} | [\alpha_n = a] \cap [\beta_n = b]) = \frac{b}{a+b},$$

c'est-à-dire qu'elle est proportionnelle au capital de points dont dispose le joueur A (*resp.* B). Enfin X_n désignera le nombre de victoires du joueur A à l'issue de n combats.

Partie I— Premier cas : les capitaux initiaux sont égaux à 1.

On suppose ici $\alpha_0 = \beta_0 = 1$.

1 — **1.1.** Déterminer la loi des variables X_1 et X_2.
1.2. python Rédiger une fonction `simulX` qui reçoit un entier n, simule n combats et renvoie le nombre X_n de victoires du joueur A.

2 — On suppose encore ici que $\alpha_0 = \beta_0 = 1$.
2.1. Rédiger une fonction `loiX` qui reçoit un entier n et renvoie, sous forme de liste, des valeurs approchées des probabilités $\mathbf{P}(X_n = 0), \mathbf{P}(X_n = 1), ..., \mathbf{P}(X_n = n)$, obtenues en faisant 10000 simulations de la variable X_n.
2.2. Exécuter cette fonction pour différentes valeurs de n et représenter graphiquement (par exemple à l'aide de la fonction donnée en préambule) les lois correspondantes. Que peut-on conjecturer ?
2.3. Démontrer cette conjecture.

3 — Si $n = 2p+1$ avec $p \in \mathbf{N}$, calculer la probabilité que le joueur A remporte le tournoi à l'issue de n combats, c'est-à-dire qu'il gagne plus de combats que son adversaire à l'issue de ces $2p+1$ combats. Commenter cette réponse.

Partie II— Deuxième cas : les capitaux initiaux sont resp. égaux à a et 1.

On suppose désormais que $\alpha_0 = a$ et $\beta_0 = 1$.

4 — Déterminer la loi de la variable X_3 dans ce cas.

5 — Démontrer que : $\forall k \in [\![0\,,\,n]\!],\quad \mathbf{P}(X_n = k) = \frac{\binom{a+k-1}{a-1}}{\binom{a+n}{a}}$.

En déduire la probabilité que le joueur A perde le tournoi (c'est-à-dire qu'il remporte moins de victoires que B).

6 — Calculer l'espérance de la variable X_n.

{PS}{CCAgroVéto}{10}

[Sol 1.19] ■ **Exercice 1.19** ***Agro—Veto, 2018, Blanche Neige lance un dé équilibré*** Blanche Neige lance un dé équilibré,

- ✓ si elle obtient 1, elle relance le dé : $\begin{cases} \text{si elle obtient } 1,2 \text{ ou } 3, \text{ le nain numéro } 1 \text{ lance le dé } D_1 \\ \text{si elle obtient } 4,5 \text{ ou } 6, \text{ le nain numéro } 7 \text{ lance le dé } D_7 \end{cases}$
- ✓ si elle obtient 2, le nain numéro 2 lance le dé D_2
- ✓ etc…
- ✓ si elle obtient 6, le nain numéro 6 lance le dé D_6.

On note X le numéro du nain qui lance son dé. [8]

1 — Justifier que $\mathbf{P}(X = 1) = \mathbf{P}(X = 7) = \frac{1}{12}$ et pour tout $k \in [\![2\,,\,6]\!]$, $\mathbf{P}(X = k) = \frac{1}{6}$.

2 — python Soit $i \in [\![1\,,\,6]\!]$. On suppose que le dé numéro i comporte $i-1$ faces blanches et $7-i$ faces noires. On note Y la variable aléatoire réelle égale au nombre de lancers nécessaires pour obtenir une face blanche.
Écrire une fonction Python qui simule la loi de X et la loi de Y.

3 — On note B_k l'événement : « on n'obtient que des faces blanches lors des k premiers lancers ». Calculer $\mathbf{P}(B_k)$.

4 — Déterminer $\mathbf{P}(Y = j | X = i)$ puis $\mathbf{E}(Y|X = i)$.

4.0.2. Aléatoire continu

{PS}{CCAgroVéto}{1}

[Sol 1.20] ■ **Exercice 1.20** ***Agro—Veto, Sujet 7, 2018*** *On rappelle que si* X *et* Y *deux variables aléatoires indépendantes de densité* f_X *et* f_Y *alors* $X + Y$ *est une variable à densité et la fonction* h *définie sur* $\mathbf{R}$ *par :*

$$\text{Pour tout réel } t,\quad h(t) = \int_{-\infty}^{+\infty} f_X(u) f_Y(t-u)\,du$$

est une densité de $X + Y$. [9] On considère deux variables aléatoires indépendantes : U et V suivant, chacune, la loi uniforme sur $[0;1]$.

1 — Justifier son existence, puis déterminer une densité f de la variable aléatoire U^2, ainsi qu'une densité de V^2.

2 — **2.1.** On considère la variable aléatoire $Z = U^2 + V^2$. Justifier que Z admet une densité de probabilité, notée h.

2.2. python Écrire un programme permettant de simuler la variable aléatoire Z et d'estimer $\mathbf{P}(Z \leq 1)$.

3 — **3.1.** Montrer que, pour $0 < x \leq 1$, on a : $h(x) = \frac{1}{4}\int_0^x \frac{1}{\sqrt{x-t}}\frac{1}{\sqrt{t}}\,dt$.

3.2. Montrer que, pour $0 < x \leq 1$, on a : $h(x) = \frac{1}{4}\int_0^1 \frac{1}{\sqrt{1-y}}\frac{1}{\sqrt{y}}\,dy$.

3.3. Montrer que, sur $]0;1]$, on a : $h(x) = \frac{\pi}{4}$. *Indication : On pourra utiliser le changement de variable* $y = \sin^2(u)$)..

3.4. Interpréter graphiquement le résultat en terme d'aire.

4 — On considère une suite de variables aléatoires de Bernoulli $(Y_n)_{n\geq 1}$ mutuellement indépendantes et de même paramètre $\frac{\pi}{4}$, et on note $S_n = \frac{Y_1+\cdots+Y_n}{n}$ pour tout entier $n \geq 1$.

4.1. Soit $\varepsilon > 0$, déterminer, en fonction de n et ε, une majoration de $\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| \geq \varepsilon\right)$.

8. Attention, ici on ne suppose pas les dés D_i équilibrés. Seul le dé initial l'est.
9. Attention, ici la formule du produit de convolution est rappelée, mais ce n'est pas toujours le cas !

4.2. En déduire, à partir de quelle valeur de n, il est possible de définir un intervalle de confiance de niveau de confiance $0,95$ de $\frac{\pi}{4}$ et d'amplitude 2×10^{-2}.

4.3. À l'aide de la simulation précédente, déterminer un intervalle de confiance de niveau de confiance $0,95$ de $\frac{\pi}{4}$ et d'amplitude 2×10^{-2}.

5 — Existe-t-il d'autres alternatives pour déterminer un intervalle de confiance de niveau de confiance $0,95$ de $\frac{\pi}{4}$ et d'amplitude 2×10^{-2} ?

{PS}{CCAgroVéto}{3}

[Sol 1.21]

■ **Exercice 1.21** ***Agro—Véto, 2016, La seule loi possible à densité à absence de mémoire est l'exponentielle*** On s'intéresse à la dissolution d'une pastille de chlore. On note T la variable aléatoire correspondant au temps qu'elle met pour se dissoudre. On suppose qu'il existe un réel $\lambda>0$ tel que pour tout réel $t\in[0,+\infty[$:

$$\frac{\mathbf{P}(t<T<t+h|T>t)}{h}\underset{h\to0^+}{\longrightarrow}\lambda$$

1 — [10] Soit Y une variable aléatoire suivant une loi uniforme sur $[0,1[$; soit $X=\frac{-\ln(1-Y)}{\lambda}$.

1.1. Déterminer la fonction de répartition de X.

1.2. Quelle est la loi de X ? Reconnaître cette loi en et précisant le paramètre.

1.3. python Simuler la loi de X avec un programme Python.

2 — On note F la fonction de répartition de T.

2.1. On admet que $\mathbf{P}(T>0)=1$; que vaut F sur $]-\infty,0]$?. Montrer que :

$$\frac{F(t+h)-F(t)}{h}\underset{h\to0^+}{\longrightarrow}\lambda(1-F(t)).$$

En déduire que F est dérivable sur $[0,+\infty[$

2.2. Montrer que F est solution de l'équation différentielle sur $[0,+\infty[$: $y'+\lambda y=\lambda$.

2.3. Déterminer F.

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

{PS}{CCAgroVéto}{4}

[Sol 1.22]

■ **Exercice 1.22** ***Agro—Véto, Sujet 6, 2018*** ***Rappel : algorithme de dichotomie.*** *On considère une fonction g continue sur un intervalle $[a;b]$. On suppose que g s'annule exactement une fois sur $[a;b]$ en un point que l'on note α. On définit les suites $(a_k)_{k\geq0}$ et $(b_k)_{k\geq0}$ de la façon suivante :*

✓ $a_0=a$ *et* $b_0=b$.

✓ *Pour tout entier naturel k, on note $c_k=\frac{a_k+b_k}{2}$ et :*

$$\text{si } g(a_k)g(c_k)\leq0\text{, alors } a_{k+1}=a_k \text{ et } b_{k+1}=c_k$$

$$\text{sinon } a_{k+1}=c_k \text{ et } b_{k+1}=b_k$$

On sait alors que les suites $(a_k)_{k\geq0}$ et $(b_k)_{k\geq0}$ convergent toutes les deux vers α.

Soit f la fonction définie sur $\mathbf{R}_+^\star$ par :

$$\forall x\in\mathbf{R}_+^\star,\quad f(x)=\ln(x)-\ln(x+1)+\frac{1}{x}.$$

1 — Montrer que l'équation $f(x)=1$ admet une unique solution notée α.

2 — En utilisant des valeurs approchées de $\ln(2)$ et de $\ln(3)$ à l'aide de Python, justifier que $\frac{1}{3}\leq\alpha\leq\frac{1}{2}$.

3 — python En utilisant l'algorithme de dichotomie, écrire une fonction qui prend en argument un entier naturel n, deux réels a et b et la fonction f, et qui renvoie α à 10^{-n} près.

10. Question archi classique ! Déjà faite plusieurs fois en TD

4 — Soit Φ la fonction définie sur $\mathbf{R}$ par :

$$\Phi(x) = \begin{cases} \frac{1}{x^2(x+1)} & x > \alpha, \\ 0 & \text{sinon.} \end{cases}$$

Montrer que Φ est une densité de probabilité.

5 — Montrer que $\int_{-\infty}^{+\infty} t\Phi(t)\,dt$ converge absolument.

6 — Montrer que : $\forall t > \alpha, f'(t) = t\Phi(t) - \frac{1}{t^2}$.

7 — Soit X une variable aléatoire admettant Φ pour densité. Calculer l'espérance de X de deux manières différentes et en donner un encadrement par deux entiers consécutifs.

4.0.3. Statistiques

{PS}{CCAgroVéto}{2}

[Sol 1.23] **Exercice 1.23** ***Agro—Véto, Sujet 3, 2018***

On pourra utiliser pour les programmes `Python` la fonction `linalg.matrix_rank()` du module `numpy`, qui permet de déterminer le rang d'une matrice, comme le montre l'exemple suivant :

```
import numpy as np
A = np. array ( [ [1 ,2 ,1] , [2 ,3 ,2], [3 ,5 ,3]] )
print ( np. linalg . matrix_rank (A) )
```

La dernière ligne affiche le rang de la matrice $\begin{pmatrix} 1 & 2 & 1 \\ 2 & 3 & 2 \\ 3 & 5 & 3 \end{pmatrix}$, *c'est-à-dire* 2.

On pourra aussi utiliser la fonction `randint()` du module `random`. Pour `a` et `b` deux entiers `randint(a,b)` retourne un entier équiprobablement entre `a` et `b` (`a` et `b` étant inclus). On considère la matrice :

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 0 & 1 \\ -3 & 0 & -1 \end{pmatrix}.$$

1 — **1.1.** Écrire une fonction Python prenant en arguments deux vecteurs de taille 3 et renvoyant un booléen (`True` ou `False`) indiquant s'ils sont colinéaires. *(On pourra représenter les vecteurs par des listes).*

1.2. Écrire une fonction Python `vecteurs_propres(u)` prenant en argument un vecteur de taille 3 et renvoyant un booléen (`True` ou `False`) indiquant s'il est un vecteur propre de A.

2 — **2.1.** Vérifier que -1, 1, 2 sont valeurs propres de A et préciser pour chacune un vecteur propre associé.

2.2. La matrice A est-elle diagonalisable ?

3 — Soient $X_1, \cdots, X_n$, n variables aléatoires indépendantes suivant la loi de Bernoulli de paramètre $p \in]0;1[$. On note : $M_n = \frac{1}{n}\sum_{k=1}^{n} X_k$ et $M_n^\star = \frac{M_n - p}{\sqrt{\frac{p(1-p)}{n}}}$.

3.1. Donner, pour $\alpha \in \mathbf{R}_+^\star$, l'approximation de la probabilité $P([-\alpha < M_n^\star < \alpha])$ donnée par le théorème central limite.

3.2. En déduire que $\left[M_n - \frac{1}{\sqrt{n}}, M_n + \frac{1}{\sqrt{n}}\right]$ est un intervalle de confiance de p au seuil de 95%. *On pourra admettre que, $\forall x \in [0;1]$, $x(1-x) \leq \frac{1}{4}$ et si Φ désigne la fonction de répartition d'une variable suivant une loi normale centrée réduite, alors $\Phi(1;96) \approx 0,975$.*

4 — On note N_V le nombre de vecteurs propres de A dont les cœfficients sont des entiers de $[\![-5\,,\,5]\!]$.

4.1. Expliquer comment le programme suivant permet d'estimer la valeur de N_V :

```
def simul ():
    u = [ randint ( -5 ,5) for k in range (3) ]
    return vecteurs_propres(u)
n = 10000 #Valeur de n a definir.
nb = 0
```

```
for k in range (n):
    if simul ():
        nb += 1
print ( round (nb/n *11**3)) # round (x) = l'entier le plus proche de x.
```

4.2. Comment choisir n pour que l'on soit sûr à 95% de la valeur affichée ?

4.3. Commenter le résultat obtenu.

5 Solutions

Solution (Exercice 3.1)

1 — 1.1. C'est presque une question de cours ! Comme pour P de degré $p \geqslant 1$, le degré de P′ est $p-1$, on obtient par récurrence : $\boxed{\text{si } k \leqslant p,\ \deg(P^{(k)}) = \deg(P) - k, \text{ et si } k > p, P^{(k)} = 0 (\Rightarrow \deg = -\infty)}$

1.2. La dérivation d'un polynôme donne un polynôme donc $\Phi(P) \in \mathbf{R}[X]$. Si $\deg(P) \leqslant n$, $\forall k \in [\![0\,,\,n]\!]$ $\deg P^{(k)} \leqslant n \Rightarrow \deg(\Phi(P)) \leqslant n$ par propriété sur le degré d'une combinaison linéaire de polynômes. Donc $\boxed{\Phi(P) \in \mathbf{R}_n[X]}$, de plus la linéarité découle de la linéarité de la dérivation. Ainsi, en conclusion : $\boxed{\Phi \text{ est un endomorphisme de } \mathbf{R}_n[X].}$

2 — 2.1. La base canonique de $\mathbf{R}_n[X]$ est $(1, X, X^2, ..., X^n)$ et pour tout $k \leqslant n$, $(X^k)' = kX^{k-1}$, on en déduit la forme de D :

$$\boxed{D = \begin{pmatrix} 0 & 1 & 0 & 0 & \cdots & 0 \\ \vdots & 0 & 2 & 0 & \cdots & 0 \\ \vdots & & \ddots & \ddots & \ddots & \vdots \\ \vdots & & & \ddots & \ddots & 0 \\ \vdots & & & & \ddots & n \\ 0 & 0 & \cdots & \cdots & \cdots & 0 \end{pmatrix}.}$$

2.2.
```
import numpy as np
def D(n):
    """
    en entree un entier n plus grand ou egal a 1
    en sortie la matrice D
    """
    M=np.zeros((n+1,n+1))
    for i in range(n):
        M[i,i+1]=i+1
    return(M)
```

2.3. Pour tout $P \in \mathbf{R}_n[X]$, tout $k \leqslant n$, $P^{(k)} = \Delta^k(P)$ et la matrice de $\Delta^k = \underbrace{\Delta \circ \Delta \circ ... \circ \Delta}_{k \text{ fois}}$ est D^k. Donc l'endomorphisme Φ s'écrit aussi $\Phi = \sum_{k=0}^{n} \binom{n}{k} \Delta^k$. Comme I_{n+1} et D commutent, on peut appliquer la formule du binôme et $(I_{n+1} + D)^n = \sum_{k=0}^{n} \binom{n}{k} D^k$, alors par propriété sur la matrice d'une combinaison linéaire d'endomorphismes on reconnait que $\sum_{k=0}^{n} \binom{n}{k} D^k$ est la matrice de $\sum_{k=0}^{n} \binom{n}{k} \Delta^k$, donc que

$\boxed{(I_{n+1} + D)^n \text{ est la matrice de } \Phi, \text{ c'est-à-dire } (I_{n+1} + D)^n = A.}$

2.4. On commence par écrire un programme qui calcule A à l'aide de la question précédente.
```
def A(n):
    I=np.eye(n+1)
    M=I+D(n)
    A=I                 #on initialise A = M^0
    for k in range(n): #on passe n fois dans la boucle
        A=np.dot(A,M)   #on obtient bien A = M^n
    return A
```
Puis on définit une fonction prenant aussi comme paramètre n :
```
def Phi(n,coordonneesP):
    """
    en entree un entier n >= 1 et une liste de n+1 termes représentant les coordonnées
    de P dans la base (1, X, .., X^n)
    en sortie une liste des coordonnees de Phi(P) dans la meme base
    """
    return np.dot(A(n),coordonneesP)
```

3 — 3.1. Si $P \neq 0$ alors : $\forall k \in [\![1\,,\,n]\!]$, $\deg(P^{(k)}) < \deg(P)$. De plus aucun coefficient de la combinaison linéaire n'est nul, donc $\deg\left(\sum_{k=0}^{n} \binom{n}{k} P^{(k)}\right) = \deg(P) \iff \boxed{\deg(\Phi(P)) = \deg(P)}$.

3.2. $P \in \text{Ker}(\Phi) \iff \Phi(P) = 0$. Or si $P \neq 0$, on vient de voir $\deg(\Phi(P)) = \deg(P)$ donc $\Phi(P) \neq 0$, donc $P \notin \text{Ker}(\Phi)$. En conclusion : $\boxed{\text{Ker}(\Phi) = \{0\} \text{ et } \Phi \text{ est injective}}$

3.3. L'application Φ est un endomorphisme injectif en dimension finie, donc $\boxed{\Phi \text{ est bijectif.}}$

4 — 4.1. Le plus simple est de revenir à A la matrice associée à Φ qui est une combinaison linéaire de matrices triangulaires supérieures, donc est triangulaire supérieure. Dans ce cas, on sait que les valeurs propres de A se lisent sur sa diagonale. Or par construction de D, seul $D^0 = I_{n+1}$ a une diagonale non nulle, et $\binom{n}{0} = 1$ donc la diagonale de A n'est composée que de 1. En conclusion :

$\boxed{\Phi, \text{ comme A, a 1 comme unique valeur propre.}}$

4.2. Cherchons alors $E_1(\Phi)$ en revenant à la définition : $P \in E_1(\Phi) \iff \Phi(P) = P$.
Tout vecteur propre $P \in \mathbf{R}_n[X]$ vérifie

$$P + \sum_{k=1}^{n} \binom{n}{k} P^{(k)} = P \iff \sum_{k=1}^{n} \binom{n}{k} P^{(k)} = 0.$$

Puisque

$$\deg\left(\sum_{k=2}^{n} \binom{n}{k} P^{(k)}\right) < \deg(-P'),$$

nous obtenons clairement une contradiction sauf si $P' = 0$ (auquel cas l'inégalité précédente devient « $-\infty < -\infty$ »), les autres dérivées sont alors nulles elles aussi. Réciproquement si $P(X) = a$ avec $a \in \mathbf{R}$, alors $\Phi(P) = P$. Conclusion : $\boxed{E_1(\Phi) = \mathbf{R}_0[X]}$ de dimension 1.

4.3.

✓ Φ a une unique valeur propre : 1, et l'espace propre associé est de dimension $1 \neq n+1$ qui est la dimension de $\mathbf{R}_n[X]$, donc Φ n'est pas diagonalisable.

✓ Autre argument : on s'intéresse cette fois à la matrice A. Comme A a une unique valeur propre 1, elle est diagonalisable si et seulement si elle est semblable à la matrice $1 \times I_{n+1} = I_{n+1}$: $A = P^{-1} I_{n+1} P = I_{n+1} P^{-1} P = I_{n+1}$, ce qui n'est pas vrai.

Donc A n'est pas diagonalisable, donc $\boxed{\Phi \text{ n'est pas diagonalisable}}$.

■■ *Solution (Exercice 3.2)* Soit $(n,k) \in \mathbf{N}^2$ avec $1 \leq k \leq n$; on considère une matrice $M = (a_{i,j}) \in \mathcal{M}_n(\mathbf{R})$ telle que

$$M^2 = J_n + (k-1) \times I_n \text{ avec } J_n = \begin{pmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \cdots & \vdots \\ 1 & 1 & \cdots & 1 \end{pmatrix} \text{ et } I_n = \begin{pmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & 1 \end{pmatrix}.$$

1 — Fonction qui calcule les valeurs propres de M^2, les vecteurs propres et qui vérifient les résultats obtenus :

```
import numpy as np
def elt_propre(n,k):
    I = np.eye(n,n) # matrice unité d'ordre n
    J = np.ones((n,n))
    A = J + (k-1)*I # M^2
    vap,vep = np.linalg.eigh(A) # valeur propres, tableau des vecteurs propres en colonne, matrice symétriqu
    Ecart = np.zeros((n,n))
    for i in range(n):
        lbda = vap[i] #récupère la (i+1)ème valeur propre
        V = vep[:,i] # vecteur propre associé (i+1)ème colonne de vep
        D = np.dot(A,V) - lbda*V #différence entre les deux colonnes
        Ecart[:,i] = D
    return vap,vep,Ecart
vap,vep,Ecart = elt_propre(3,2)
```

On obtient pour les valeurs propres : [1. 1. 4.], pour les vecteurs propres [[-0.81649658 0. -0.57735027] [0.40824829 -0.70710678 -0.57735027] [0.40824829 0.70710678 -0.57735027]] et enfin l'écart : [[-1.11022302e-16 0.00000000e+00 4.44089210e-16] [6.66133815e-16 0.00000000e+00 -1.77635684e-15] [6.66133815e-16 0.00000000e+00 -1.33226763e-15]].

2 — 2.1. Par définition, $\mathrm{Rg}(J_n) = \dim \mathrm{Vect}(C_1, \ldots, C_n)$ où $C_1, \ldots, C_n$ désignent les n colonnes de J_n. Comme toutes les colonnes sont égales à C_1 et $C_1 \neq 0$, on en déduit que $\boxed{\mathrm{Rg}(J_n) = 1}$.

2.2. ✓ 1er cas : $n = 1$ donc $J_1 = (1) = I_1$; alors 1 est la seule valeur propre de J_1 et $\dim E_1(J_1) = 1$.

✓ 2e cas : $n \geq 2$ $\mathrm{Rg}(J_n - 0I_n) = \mathrm{Rg}(J_n) = 1 < n$ donc 0 est valeur propre immédiate de J_n. De plus, $\dim E_0(J_n) = \dim \mathrm{Ker}(J_n) = n - \mathrm{Rg}(J_n) = n - 1$. Par ailleurs, il est clair que le vecteur $U = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}$ vérifie $J_n U = \begin{pmatrix} n \\ \vdots \\ n \end{pmatrix} = nU$ et comme $U \neq 0$, on en déduit que n est valeur propre de J_n et U est un vecteur propre associé. Comme la somme des dimensions des sous-espaces propres ne peut dépasser n (puisque la matrice est de taille n), sachant qu'on a trouvé $\dim E_0(J_n) = n - 1$, on conclut que $\dim E_n(J_n) = 1$ et J_n n'admet pas d'autres valeurs propres que 0 et n.

Conclusion : $\boxed{\mathrm{Spec}(J_n) = \{0, n\}}$, $\boxed{\dim E_0(J_n) = n - 1}$ et $\boxed{\dim E_n(J_n) = 1}$.

2.3. La matrice J_n est diagonalisable car J_n est une matrice symétrique réelle. De plus J_n est diagonalisable car J_n est carrée d'ordre n et la somme des dimensions des sous-espaces propres est égale à n.

3 — On supposera dorénavant $n \geq 2$, le cas $n = 1$ ne présentant pas d'intérêt.

3.1. Comme $\dim E_0(J_n) = n - 1$ et $\dim E_n(J_n) = 1$, J_n est semblable à la matrice diagonale $D_n = \begin{pmatrix} 0 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & 0 & 0 \\ 0 & \cdots & 0 & n \end{pmatrix}$.

Ainsi il existe une matrice P inversible telle que $P^{-1}J_nP = D_n$. Or $M^2 = J_n + (k-1)I_n$ donc $P^{-1}M_n^2P = P^{-1}J_nP + (k-1)P^{-1}I_nP = D_n + (k-1)I_n$.

$$P^{-1}M_n^2P = D_n + (k-1)I_n = \underbrace{\begin{pmatrix} k-1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \vdots \\ \vdots & \ddots & k-1 & 0 \\ 0 & \cdots & 0 & k-1+n \end{pmatrix}}_{\Delta_n}.$$

$\boxed{\text{Ainsi } M^2 \text{ est diagonalisable car elle est semblable à la matrice diagonale } \Delta_n = D_n + (k-1)I_n}$.

3.2. Les valeurs propres de M^2 sont les termes diagonaux de Δ_n :

$$\boxed{\mathrm{Spec}(M^2) = \{k-1, k-1+n\}}.$$

Comme $n \neq 0$, ces deux valeurs propres sont distinctes. De plus, $\dim E_{k-1}(M^2) = \dim \mathrm{Ker}(M^2 - (k-1)I_n) = \dim \mathrm{Ker}(J_n) = n - 1$, donc $\dim E_{k-1+n}(M^2) = \dim \mathrm{Ker}(M^2 - (k-1+n)I_n) = \dim \mathrm{Ker}(J_n - nI_n) = \dim E_n(J_n) = 1$.

$$\boxed{\dim E_{k-1}(M^2) = n-1} \text{ et } \boxed{\dim E_{k-1+n}(M^2) = 1}.$$

4 — Soit $\mu \in \mathbf{R}$; si μ est valeur propre de M, alors il existe un vecteur X non nul tel que $MX = \mu X$. On a alors $M^2X = M(\mu X) = \mu MX = \mu^2 X$. Comme $M^2X = \mu^2 X$ avec $X \neq 0$, alors μ^2 est valeur propre de M^2. On a donc montré que **si** μ est valeur propre de M **alors** μ^2 est valeur propre de M^2 *i.e.* $\mu^2 \in \{k-1, k-1+n\}$.

On a $k - 1 \geq 0$ et $k - 1 + n \geq 0$ car $1 \leq k \leq n$.

Donc : $\mu^2 = k - 1 \iff \mu = \pm\sqrt{k-1}$ et $\mu^2 = k - 1 + n \iff \mu = \pm\sqrt{k-1+n}$.

Les seules valeurs propres possibles de M sont : $-\sqrt{k-1}, \sqrt{k-1}, -\sqrt{k-1+n}$ et $\sqrt{k-1+n}$.

$$\boxed{\mathrm{Spec}(M) \subset \{-\sqrt{k-1}, \sqrt{k-1}, -\sqrt{k-1+n}, \sqrt{k-1+n}\}}.$$

5 — Un exemple de réseau vérifiant les hypothèses pour $n = 3$: $A = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix}$ (chaque personne est amie avec chacune des deux autres personnes).

6 — Si les personnes i et j sont amies, alors $A_{i,j} = 1$ mais on a également $A_{j,i} = 1$ donc A est symétrique.

7 — D'après l'énoncé, $(A^2)_{i,i}$ donne le nombre d'amis de la personne i ; or chaque personne a exactement k amis par hypothèse donc $\boxed{(A^2)_{i,i} = k}$. Pour $i \neq j$, $(A^2)_{i,j}$ donne le nombre d'amis communs des personnes i et j ; or deux

personnes distinctes ont par hypothèse exactement un ami en commun donc $\boxed{(A^2)_{i,j} = 1}$. Ainsi

$$\boxed{A^2 = \begin{pmatrix} k & 1 & \cdots & 1 \\ 1 & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 1 \\ 1 & \cdots & 1 & k \end{pmatrix}} \text{ soit } \boxed{A^2 = J_n + (k-1)I_n}.$$

8 — 8.1. Il y a $\binom{n}{2}$ possibilités de choisir deux personnes distinctes parmi les n, autrement dit :
$\boxed{\text{le nombre de couples comportant deux personnes distinctes du réseau est } \binom{n}{2}}$.

8.2. Considérons une personne i donnée (avec $1 \leq i \leq n$), celle-ci est amie avec exactement k personnes du réseau donc
$\boxed{\text{le nombre de couples de personnes qui ont la personne } i \text{ comme ami en commun est donc } \binom{k}{2}}$.

8.3. Il y a donc $\binom{k}{2}$ couples de personnes qui ont la personne 1 comme ami commun, $\binom{k}{2}$ couples de personnes qui ont la personne 2 comme ami commun, ..., $\binom{k}{2}$ couples de personnes qui ont la personne n comme ami commun. Comme il y a au total $\binom{n}{2}$ couples et que chaque couple de deux personnes distinctes ont exactement un ami en commun on a donc :
$\binom{n}{2} = \binom{k}{2} + \binom{k}{2} + \ldots + \binom{k}{2}$ soit $\boxed{\binom{n}{2} = n\binom{k}{2}}$. Ainsi on a : $\dfrac{n(n-1)}{2} = n\dfrac{k(k-1)}{2}$ ce qui équivaut à $\boxed{k^2 - k + 1 = n}$.

9 — 9.1. On a vu dans la question précédente que A s'écrit sous la forme $A = J_n + (k-1)I_n$. Dès lors, on peut appliquer le résultat de la question 4. On en déduit que les valeurs propres possibles de A sont : $-\sqrt{k-1}, \sqrt{k-1}, -\sqrt{k-1+n}, \sqrt{k-1+n}$.
Or $k^2 - k + 1 = n$ donc $k - 1 + n = k^2$ et par conséquent $\sqrt{k-1+n} = \sqrt{k^2} = k$ ($k \geq 0$).
$\boxed{\text{Les valeurs propres éventuelles de A sont donc : } -\sqrt{k-1}, \sqrt{k-1}, -k, k}$.

9.2. Commençons par remarquer que n est impair : en effet, $n = k^2 - k + 1 = k(k-1) + 1$ et pour tout entier naturel k, $k(k-1)$ est pair. Notons $n_1 = \dim E_k(A) = \dim \mathrm{Ker}(A - kI_n)$, $n_2 = \dim E_{-k}(A) = \dim \mathrm{Ker}(A + kI_n)$, $n_3 = \dim E_{\sqrt{k-1}}(A) = \dim \mathrm{Ker}(A - \sqrt{k-1}I_n)$ et $n_4 = \dim E_{-\sqrt{k-1}}(A) = \dim \mathrm{Ker}(A + \sqrt{k-1}I_n)$. On rappelle que pour tout $X \in \mathcal{M}_{n,1}(\mathbf{R})$: $AX = kX \Rightarrow A^2X = k^2X$. On en déduit que $E_k(A) \subset E_{k^2}(A^2)$ et par conséquent $\dim E_k(A) \leq \dim E_k(A^2)$. Or $\dim E_{k^2}(A^2) = \dim E_{k-1+n}(A^2) = 1$ d'après la deuxième partie de la question 3 donc $\dim E_k(A) \leq 1$. $A_{i,j} = 1$ si les personnes i et j sont amies et $A_{i,j} = 0$ sinon. Donc la somme des coefficients de la i-ième ligne de A correspond au nombre d'amis de la personne i à savoir k. On en déduit que pour tout $i \in [\![1, n]\!]$, $\displaystyle\sum_{j=1}^{n} A_{i,j} = k$ ce qui se traduit matriciellement par :

$$A\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = \begin{pmatrix} k \\ \vdots \\ k \end{pmatrix} = k\begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix}.$$

Cela prouve que k est valeur propre de A et par conséquent $\dim E_k(A) \geq 1$. Ainsi on peut affirmer que $\dim E_k(A) = 1$ *i.e.* $\boxed{n_1 = 1}$. De même, $AX = -kX \Rightarrow A^2X = k^2X$ donc $E_{-k}(A) \subset E_{k^2}(A^2)$ et par conséquent $n_2 \leq 1$. D'après l'énoncé, on sait que : $kn_1 - kn_2 + \sqrt{k-1}n_3 - \sqrt{k-1}n_4 = 0$ soit :

$$\boxed{k(n_1 - n_2) + \sqrt{k-1}(n_3 - n_4) = 0}.$$

On a vu que $n_1 = 1$. Supposons que l'on ait aussi $n_2 = 1$ alors $k(n_1 - n_2) = 0$. Donc on aurait $\sqrt{k-1}(n_3 - n_4) = 0$.
Or $k \geq 2$ car chaque couple de personnes distinctes a un ami en commun, cette personne a donc au minimum deux amis.
On a donc $\sqrt{k-1} \neq 0$ et par conséquent $n_3 - n_4 = 0$ *i.e.* $n_3 = n_4$. Par ailleurs, A étant symétrique, A est diagonalisable donc

$$\boxed{n_1 + n_2 + n_3 + n_4 = n}.$$

On aurait alors $2n_1 + 2n_3 = n$ ce qui est impossible car n est impair.
On en déduit que $\boxed{n_2 = 0}$ (autrement dit $-k$ n'est pas valeur propre de A).
Comme $n_1 = 1$ et $n_2 = 0$, la relation $k(n_1 - n_2) + \sqrt{k-1}(n_3 - n_4) = 0$ devient $k + \sqrt{k-1}(n_3 - n_4) = 0$ soit $n_4 - n_3 = \dfrac{k}{\sqrt{k-1}}$.

D'où $(n_4 - n_3)^2 = \dfrac{k^2}{k-1} = \dfrac{k^2-1+1}{k-1} = k+1+\dfrac{1}{k-1}$.
On a donc $\dfrac{1}{k-1} = (n_4-n_3)^2 - (k+1)$.
Comme n_4, n_3 et $k+1$ sont des entiers, on en déduit que $\dfrac{1}{k-1}$ est un entier (positif).
Or ceci n'est possible que si $k-1=1$ *i.e.* $\boxed{k=2}$.
De fait, $n = k^2-k+1 = 4-2+1 = 3$. On a donc montré que $\boxed{n=3}$

■■ *Solution (Exercice 2.3)*

1 — 1.1. On calcule facilement $\boxed{T_2 = 2X^2-1 \text{ et } T_3 = 4X^3-3X.}$

1.2. Le degré et le coefficient dominant de T_0 sont évidents. On va montrer par récurrence que pour tout entier naturel non nul n, T_n es de degré n et que le coefficient dominant de T_n est $c_{n,n} = 2^{n-1}$.
Pour $n=1$, nous avons effectivement T_1 de degré 1 avec un coefficient dominant égal à $1 = 2^{1-1}$, et T_2 de degré 2 avec un coefficient dominant égal à $2 = 2^{2-1}$.
Supposons maintenant que : T_n est de degré n, avec pour coefficient dominant $c_{n,n} = 2^{n-1}$ et aussi, T_{n+1} de degré $n+1$, avec pour coefficient dominant $c_{n+1,n+1} = 2^n$.
Nous avons alors $\deg(2XT_{n+1}) = n+2$ et $\deg(T_n) = n$ comme ces degrés sont distincts, le degré de la somme de ces deux polynômes est égal au maximum des deux degrés et nous avons $\deg(T_{n+2}) = n+2$. D'autre part, le coefficient de plus haut degré de T_{n+2} provient exclusivement de $2XT_{n+1}$, plus précisément :

$$c_{n+2,n+2} = 2c_{n+1,n+1} = 2\times 2^n = 2^{n+1} = 2^{n+2-1}$$

et la preuve est faite par récurrence (sur deux termes). $\boxed{\forall n \in \mathbf{N},\ \deg(T_n) = n}$ Si l'on note $c_{n,n}$ le coefficient dominant du polynôme T_n, nous avons $\boxed{c_{0,0} = 1 \text{ et } \forall n \in \mathbf{N}^*,\ c_{n,n} = 2^{n-1}}$

1.3. Pour $n \in \mathbf{N}$ et $k \in [\![0\,,\ n]\!]$, notons $c_{n,k}$ le coefficient de X^k dans le polynôme T_n. Pour tout entier naturel n, nous avons :

$$\begin{aligned}
T_{n+1} &= \sum_{k=0}^{n+1} c_{n+1,k}X^k \\
2XT_{n+1} &= \sum_{k=0}^{n+1} 2c_{n+1,k}X^{k+1} \\
2XT_{n+1} &= \sum_{k=1}^{n+2} 2c_{n+1,k-1}X^k \\
2XT_{n+1} - T_n &= \sum_{k=1}^{n+2} 2c_{n+1,k-1}X^k - \sum_{k=0}^{n} c_{n,k}X^k \\
T_{n+2} &= 2c_{n+1,n+1}X^{n+2} + \left(\sum_{k=1}^{n+1}\left(2c_{n+1,k-1} - c_{n,k}\right)X^k\right) - c_{n,0}
\end{aligned}$$

On peut conclure $\boxed{c_{n+2,n+2} = 2^{n+1},\quad \forall k \in [\![1\,,\ n]\!],\quad c_{n+2,k} = 2c_{n+1,k-1} - c_{n,k},\ c_{n+2,0} = -c_{n,0}}$

2 — 2.1.
```
import numpy as np
def degre(P):
    """
    retourne le degre du polynome P
    P est en fait la liste des coefficients
    on traite le cas où il y a des zeros à la fin
    """
    n=len(P)
    deg=-np.Inf
    k=n-1
    while (P[k]==0) and (k>=0):
        k-=1
    if (k>=0):
        deg=k
    return deg
```

2.2.
```
def etape(L,M):
```

```
    degM=degre(M)
    degL=degre(L)
    d=max(degM+1,degL)
    P=[0]*(d+1) #On initialise la sortie
    #Il y aura un soucis si un des deux polynomes est nuls
    # (gestion du degre -infini non faite)
    for k in range(int(degL)+1):
        P[k]=-L[k]

    for k in range(int(degM)+1):
        P[k+1]+=2*M[k]
    return P
```

2.3.
```
def Tchebychev(n):
    T0=[1]
    T1=[0,1]
    if (n==0):
        T2=T0
    elif (n==1):
        T2=T1
    else:
        for k in range(n-1):
            T2=etape(T0,T1)
            T0=T1
            T1=T2
    return T2
```

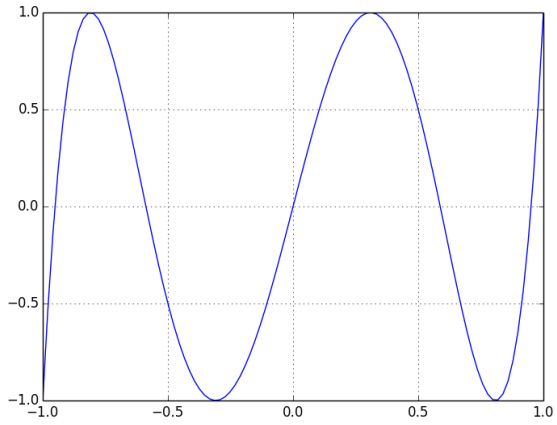
Par exemple, pour $n = 2$, nous obtenons : [-1, 0, 2] qui code $-1 + 2X^2$, ce qui est cohérent avec l'énoncé.

2.4.
```
def evalue(P,x):
    """
    Evaluation d'un polynome par l'algorithme de Horner : on retourne P(x)
    """
    d=degre(P)
    sortie=P[d]
    for k in range(d-1,-1,-1):
        sortie=sortie*x+P[k]
    return sortie
```

Cette fonction peut aussi être écrite de manière récursive (voir le TP associé).

2.5.
```
def TraceTchebychev(n):
    """
    fonction qui trace le graphe du n ieme
    polynome de Tchebychev sur [-1,1]
    """
    X=np.linspace(-1,1,100)
    Y=[]
    T=Tchebychev(n)
    for x in X:
        Y.append(evalue(T,x))
    plt.plot(X,Y)
    plt.grid()
    return T

TraceTchebychev(5)
```



Il semble que, sur le segment $[-1, 1]$, les polynômes de Tchebychev prennent leurs valeurs dans $[-1, 1]$ et admettent n racines réelles dans ce même intervalle.

3 — 3.1. On a d'après le cours :

$$\begin{aligned} \cos(a+b) &= \cos(a)\cos(b) - \sin(a)\sin(b) \\ \cos(a-b) &= \cos(a)\cos(b) + \sin(a)\sin(b) \\ \cos(a+b) + \cos(a-b) &= 2\cos(a)\cos(b) \end{aligned}$$

3.2. a est un réel quelconque. On va procéder par récurrence (double) sur l'entier n.
Pour $n = 0$, nous avons $T_0(\cos(a)) = 1 = \cos(0a)$.
Pour $n = 1$, nous avons $T_1(\cos(a)) = \cos(a) = \cos(1.a)$
Supposons maintenant que pour un certain entier n, $T_n(\cos(a)) = \cos(na)$ et $T_{n+1}(\cos(a)) = \cos((n+1)a)$.
Nous avons alors (par définition des polynômes de Tchebychev)

$$\begin{aligned} T_{n+2}(\cos(a)) &= 2\cos(a)T_{n+1}(\cos(a)) - T_n(\cos(a)) \\ &= 2\cos(a)\cos((n+1)a) - \cos(na) \text{ (par hypothèse de récurrence)} \\ &= 2\times\frac{1}{2}(\cos((n+1)a+a) + \cos((n+1)a-a)) - \cos(na) \\ &= \cos((n+2)a) + \cos(na) - \cos(na) \\ &= \cos((n+2)a) \end{aligned}$$

La preuve est alors faite par récurrence (sur deux termes).

3.3.
$$\cos(na) = 0 \iff \exists k \in \mathbf{Z} \quad na = \frac{\pi}{2} + k\pi$$

Si n est pair, prenons $a_k = \frac{\pi}{2n} + k\frac{\pi}{n}$, pour $k \in I_n = [\![-\frac{n}{2}, \frac{n}{2} - 1]\!]$. Si n est impair, prenons $a_k = \frac{\pi}{2n} + k\frac{\pi}{n}$, pour $k \in I_n = [\![-\frac{n-1}{2}, \frac{n+1}{2} - 1]\!]$. Nous obtenons n valeurs distinctes de l'intervalle $\left]-\frac{\pi}{2}, \frac{\pi}{2}\right]$. Comme la fonction cosinus est strictement décroissante sur cet intervalle, elle est en particulier injective et on déduit que les n valeurs $\cos(a_k)$ sont deux à deux distinctes. Nous avons alors

$$\forall k \in I_n \quad T_n(\cos(a_k)) = \cos(na_k) = 0$$

Nous venons de trouver n racines réelles distinctes dans l'intervalle $[-1, 1]$ $\cos(a_k)$ pour $k \in I_n$.
Nous savons qu'un polynôme de degré $n \in \mathbf{N}$ admet au plus n racines réelles distinctes. On en déduit donc que $\forall n \in \mathbf{N}$, T_n admet exactement n racines réelles distinctes toutes dans l'intervalle $[-1, 1]$.

■■ *Solution (Exercice 2.4)*

1 — L'application Φ est clairement linéaire par définition de la loi d'espace vectoriel sur $\mathbf{R}^{n+1}$: soient $P_1, P_2 \in \mathbf{R}_n[X]$ et $\lambda, \mu \in \mathbf{R}$. Alors :

$$\begin{aligned} \Phi(\lambda P + \mu Q) &= ((\lambda P + \mu Q)(x_0), (\lambda P + \mu Q)(x_1), \ldots, (\lambda P + \mu Q)(x_n)) \\ &= \lambda(P(x_0), P(x_1), \ldots, P(x_n)) + \mu(Q(x_0), Q(x_1), \ldots, Q(x_n)) = \lambda\Phi(P) + \mu\Phi(Q). \end{aligned}$$

Puisque $\dim \mathbf{R}_n[X] = n + 1 = \dim \mathbf{R}^{n+1}$, il suffit d'établir par exemple que Φ est injective. Soit donc $P \in \operatorname{Ker}\Phi$. Alors P s'annule en $x_0, \ldots, x_{n+1}$ donc en $n+1$ points distincts de $\mathbf{R}$, mais puisque $\deg P \leqslant n$, nécessairement $P = 0$[2]. Donc
$\boxed{\Phi \text{ est un isomorphisme.}}$

2 — **2.1.** On a $\Phi(X^k) = (x_0^k, x_1^k, \ldots, x_n^k)$ pour tout entier $k \in [\![1, n]\!]$, donc

$$M = \underset{\mathcal{B}^{can},\mathcal{B}^{can}}{\mathcal{M}at}(\Phi) = \boxed{\begin{pmatrix} 1 & \ldots & x_0^k & \ldots & x_0^n \\ \vdots & \ldots & & \ldots & \vdots \\ 1 & \ldots & x_n^k & \ldots & x_n^n \end{pmatrix} = \left(x_i^j\right)_{0\leqslant i,j\leqslant n}}.$$

C'est une matrice de Vandermonde.

2.2. La matrice M est canoniquement associée à un isomorphisme, elle est donc inversible (et la matrice inverse est $\underset{\mathcal{B}^{can},\mathcal{B}^{can}}{\mathcal{M}at}(\Phi^{-1})$).

3 — **3.1.** L_i est un produit de n polynômes de degré 1, donc $\boxed{L_i \in \mathbf{R}_n[X].}$

3.2. Nous avons

$$L_i(x_j) = \prod_{k=0,k\neq i}^{n} \frac{x_j - x_k}{x_i - x_k}.$$

Si $i \neq j$, alors dans le produit apparaît le facteur nul $x_j - x_j$ donc le produit total est nul.

2. Si on préfère établir la surjectivité — disons-le de suite moins évidente — alors on pourra montrer que si $(y_0, \ldots, y_{n+1}) \in \mathbf{R}^{n+1}$, le polynôme $\sum_{i=0}^{n} y_i L_i$ est un antécédent de $(y_0, \ldots, y_{n+1})$ où les L_i sont définis dans la suite.

Si $i = j$, alors on exclut le terme $k = j$ donc

$$L_i(x_j) = \prod_{k=0,k\neq i}^{n} \frac{x_i - x_k}{x_i - x_k} = \prod_{k=0,k\neq i}^{n} 1 = 1.$$

Finalement $\boxed{L_i(x_j) = 1 \text{ si } i = j \text{ et } 0 \text{ sinon.}}$ Donc $\Phi(L_j) = (0,\dots,0,\underbrace{1}_{\text{position } j},0\dots,0)$. Autrement dit

$\boxed{(\Phi(L_0),\Phi(L_n)) \text{ est la base canonique de } \mathbf{R}^{n+1}.}$

3.3. D'après la question précédente, $(L_0,\dots,L_n)$ est l'image de la base canonique par Φ^{-1} qui est un isomorphisme. Ainsi $(L_0,\dots,L_n)$ est une base de $\mathbf{R}_n[X]$.

3.4. Notons $(e_0,\dots,e_n)$ la base canonique de $\mathbf{R}^{n+1}$. Alors comme $\Phi(L_i) = e_i$ pour tout i, il vient immédiatement :

$$\boxed{\underset{(L_0,\dots,L_n),(e_0,\dots,e_n)}{\mathcal{M}\text{at}}(\Phi) = I_n.}$$

3.5.
```
#X est une liste contenant les xi
def L(i,X,x):
    '''fonction qui renvoie Li(x)'''
    n=len(X)
    P=1
    for j in range(0,n+1):
        if j=!i:
            P=P*((x-X[j])/(X[i]-X[j]))
    return P
```

4 — 4.1. On effectue le produit matriciel :

$$(\mathbf{P}(X = x_0) \quad \mathbf{P}(X = x_1) \quad \dots \quad \mathbf{P}(X = x_n))\begin{pmatrix} 1 & \dots & x_0^k & \dots & x_0^n \\ \vdots & \dots & & \dots & \vdots \\ 1 & \dots & x_n^k & \dots & x_n^n \end{pmatrix} \boxed{= (1 \quad \mathbf{E}(X) \quad \mathbf{E}(X^2) \quad \dots \quad \mathbf{E}(X^n))}$$

d'après le théorème de transfert, applicable puisque X est à support fini.

4.2. L'hypothèse revient à supposer que la matrice ${}^{\mathbf{T}}VM$ est connue. Puisque M est inversible (déjà vu précédemment), nous obtenons :

$$\boxed{V = (1 \quad \mathbf{E}(X) \quad \mathbf{E}(X^2) \quad \dots \quad \mathbf{E}(X^n))\,M^{-1}}.$$

Cette égalité donne accès à la loi de X.

4.3.
```
#L est une liste contenant les moments
import numpy as np
def LoiX(L,M):
    '''fonction qui renvoie la loi de X'''
    return np.dot(L,np.inv(M))
```

La matrice M est ici mise en argument, mais on peut facilement la construire avec Python à partir de la donnée des x_i (*i.e.*la donnée du support de X).

```
import numpy as np
def ConstrM(X):
    M=np.zeros((len(X),len(X)))
    for i in range(len(X)):
        for j in range(len(X)):
            M[i,j]=X[i]**j
    return M
```

■■ *Solution (Exercice 2.5)*

1 —
```
import numpy.linalg as la
lignes = [[a,b,c] for a in [0,1] for b in [0,1] for c in [0,1]]
#tous les choix possibles de lignes contenant des zéros et des 1
#cardinal de cet ensemble = 2 puissance 9
m = 0 # 0 est la première valeur propre (première matrice nulle)
for li1 in lignes:
    for li2 in lignes:
```

```
    for li3 in lignes:
        M = [li1,li2,li3]
        vap = la.eig(M)[0]# liste des valeurs propres de M
        if vap[0] < m:
            m = vap[0]
        if vap[1] < m:
            m = vap[1]
        if vap[2] < m:
            m = vap[2]
```

On trouve : -1.41421356237.

2 — 2.1. E est l'ensemble des matrices de $\mathcal{M}_3(\mathbf{R})$ admettant $U = \begin{pmatrix}1\\1\\1\end{pmatrix}$ pour vecteur propre.

Montrons que E est un sous-espace vectoriel de $\mathcal{M}_3(\mathbf{R})$:

❶ $E \subset \mathcal{M}_3(\mathbf{R})$.

❷ La matrice nulle 0_3 de $\mathcal{M}_3(\mathbf{R})$ appartient à E car $0_3 U = 0.U$ donc U est vecteur propre de 0_3 associé à la valeur propre 0.

❸ Soient $(M, N) \in E^2$ et $(\alpha, \beta) \in \mathbf{R}^2$; comme U est vecteur propre de M et de N, il existe deux réels λ, μ tels que $MU = \lambda U$ et $NU = \mu U$. On a alors : $(\alpha M + \beta N)U = \alpha MU + \beta NU = \alpha\lambda U + \beta\mu U = (\alpha\lambda + \beta\mu)U$ donc U est vecteur propre de $\alpha M + \beta N$ ce qui signifie que $\alpha M + \beta N \in E$: E est stable par combinaisons linéaires.

On conclut que $\boxed{\text{E est bien un sous-espace vectoriel de } \mathcal{M}_3(\mathbf{R})}$. $F = \left\{ \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & a \end{pmatrix}, (a,b,c,d,e) \in \mathbf{R}^5 \right\}$. Montrons que F est un sous-espace vectoriel de $\mathcal{M}_3(\mathbf{R})$.

$$F = \left\{ a\underbrace{\begin{pmatrix}1&0&0\\0&0&0\\0&0&1\end{pmatrix}}_{A_1} + b\underbrace{\begin{pmatrix}0&1&0\\1&0&0\\0&0&0\end{pmatrix}}_{A_2} + c\underbrace{\begin{pmatrix}0&0&1\\0&0&0\\1&0&0\end{pmatrix}}_{A_3} + d\underbrace{\begin{pmatrix}0&0&0\\0&1&0\\0&0&0\end{pmatrix}}_{A_4} + e\underbrace{\begin{pmatrix}0&0&0\\0&0&1\\0&1&0\end{pmatrix}}_{A_5}, (a,b,c,d,e) \in \mathbf{R}^5 \right\} :$$

F est l'ensemble des combinaisons linéaires des matrices A_1, A_2, A_3, A_4, A_5 de $\mathcal{M}_3(\mathbf{R})$ donc $\boxed{\text{F est un sous-espace vectoriel de } \mathcal{M}_3(\mathbf{R})}$: $F = \text{Vect}(A_1, A_2, A_3, A_4, A_5)$.

2.2. Soit $M \in F$; alors il existe $(a,b,c,d,e) \in \mathbf{R}^5, M = \begin{pmatrix} a & b & c \\ b & d & e \\ c & e & a \end{pmatrix}$.

$$\begin{aligned} M \in E &\iff \exists \lambda \in \mathbf{R}, MU = \lambda U \\ &\iff \exists \lambda \in \mathbf{R}, \begin{cases} a+b+c = \lambda \\ b+d+e = \lambda \\ c+e+a = \lambda \end{cases} \\ &\iff \begin{cases} a+b+c = b+d+e \\ a+b+c = c+e+a \end{cases} \\ &\iff \begin{cases} a = -c+d+e \\ b = e \end{cases} \end{aligned}$$

$$E \cap F = \left\{ \begin{pmatrix} -c+d+e & e & c \\ e & d & e \\ c & e & -c+d+e \end{pmatrix}, (c,d,e) \in \mathbf{R}^3 \right\}$$

$$E \cap F = \left\{ d\underbrace{\begin{pmatrix}1&0&0\\0&1&0\\0&0&1\end{pmatrix}}_{I} + e\underbrace{\begin{pmatrix}1&1&0\\1&0&1\\0&1&1\end{pmatrix}}_{A} + c\underbrace{\begin{pmatrix}-1&0&1\\0&0&0\\1&0&-1\end{pmatrix}}_{B}, (c,d,e) \in \mathbf{R}^3 \right\} = \text{Vect}(I, A, B).$$

Il est clair que : $dI + eA + cB = 0_3 \iff d = e = c = 0$ donc (I, A, B) est une famille libre, c'est une base de $E \cap F$.

3 — 3.1. $A \in E \cap F$ car on vient de voir que $E \cap F = \text{Vect}(I, A, B)$.

3.2. A est une matrice symétrique réelle donc A est diagonalisable dans une base orthonormée, ce qui justifie l'existence d'une matrice P inversible et d'une matrice D diagonale vérifiant $A = P^T D P$. Afin de déterminer P et D, on détermine les valeurs propres et sous-espaces propres de A. *comme* $A \in E$, *on s'attend à trouver* U *comme vecteur propre !*

On pose $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$;

$$AX = \lambda X \iff \left\{ \begin{array}{rcll} (1-\lambda)x + y & = & 0 & \\ x - \lambda y + z & = & 0 & L_2 \leftrightarrow L_1 \\ y + (1-\lambda)z & = & 0 & \end{array} \right.$$

$$\iff \left\{ \begin{array}{rcll} x - \lambda y + z & = & 0 & \\ (1-\lambda)x + y & = & 0 & L_2 \leftarrow L_2 + (\lambda-1)L_1 \\ y + (1-\lambda)z & = & 0 & \end{array} \right.$$

$$\iff \left\{ \begin{array}{rcll} x - \lambda y + z & = & 0 & \\ (1+\lambda-\lambda^2)y + (\lambda-1)z & = & 0 & \text{permutation } y \leftrightarrow z \\ y + (1-\lambda)z & = & 0 & \end{array} \right.$$

$$\iff \left\{ \begin{array}{rcll} x + z - \lambda y & = & 0 & \\ (\lambda-1)z + (1+\lambda-\lambda^2)y & = & 0 & L_3 \leftarrow L_3 + L_2 \\ (1-\lambda)z + y & = & 0 & \end{array} \right.$$

$$\iff \left\{ \begin{array}{rcl} x + z - \lambda y & = & 0 \\ \boxed{(\lambda-1)}\, z + (1+\lambda-\lambda^2)y & = & 0 \\ \boxed{(2+\lambda-\lambda^2)}\, y & = & 0 \end{array} \right.$$

$2 + \lambda - \lambda^2 = 0 \iff \lambda = -1$ ou $\lambda = 2$. Le système admet d'autres solutions que $(0,0,0)$ si et seulement si $\lambda \in \{-1,1,2\}$. On en déduit que $\boxed{\text{Spec}(A) = \{-1,1,2\}}$.

$AX = 2X \iff \left\{ \begin{array}{l} x = z \\ y = z \end{array} \right.$ donc $E_2 = \text{Vect}(U)$ avec $U = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}$.

$AX = -X \iff \left\{ \begin{array}{l} x = z \\ y = -2z \end{array} \right.$ donc $E_{-1} = \text{Vect}(V)$ avec $V = \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix}$

$AX = X \iff \left\{ \begin{array}{l} x = -z \\ y = 0 \end{array} \right.$ donc $E_1 = \text{Vect}(W)$ avec $W = \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}$. On retrouve le fait que A est diagonalisable car elle est de taille trois et admet trois valeurs propres distinctes. En déterminant une base orthonormée de chaque sous-espace propre, on obtiendra par recollement une base orthonormée de $\mathbf{R}^3$ et par conséquent une matrice de passage P orthogonale (c'est-à-dire qui vérifie $P^{-1} = {}^{\mathbf{T}}P$). Notons alors $U' = \frac{1}{\sqrt{3}}U$, $V' = \frac{1}{\sqrt{6}}V$, $W' = \frac{1}{\sqrt{2}}W$; ces trois vecteurs sont des vecteurs propres de A associés respectivement à $2, -1$ et 1 et forment une base orthonormée de $\mathbf{R}^3$ formée de vecteurs propres.

Donc en posant $\boxed{P = \frac{1}{\sqrt{6}} \begin{pmatrix} \sqrt{2} & 1 & \sqrt{3} \\ \sqrt{2} & -2 & 0 \\ \sqrt{2} & 1 & -\sqrt{3} \end{pmatrix}}$ et $\boxed{D = \begin{pmatrix} 2 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}}$ on a $A = P^{\mathbf{T}}DP$.

4 — Soit $M \in E \cap F = \text{Vect}(I, A, B)$; alors il existe trois réels α, β, γ tels que $M = \alpha I + \beta A + \gamma B$. Il s'agit de prouver que M est diagonalisable dans la même base de vecteurs propres (U', V', W'). On a $A = P^{\mathbf{T}}DP$; de plus $I = P^{\mathbf{T}}IP$.

Par ailleurs, $BU = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$ donc $BU = 0U$. Comme U' est colinéaire à U, on a aussi $BU' = 0U'$: U' est vecteur propre de B associé à 0.

$BV = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ -2 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$: $BV = 0V$ et par conséquent $BV' = 0V'$ ce qui signifie que V' est vecteur propre de B associé à 0. $BW = \begin{pmatrix} -1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix} = \begin{pmatrix} -2 \\ 0 \\ 2 \end{pmatrix} = -2W$ et par conséquent $BW' = -2W'$: W' est vecteur propre de B associé à -2.

On en déduit que B se diagonalise sous la forme $B = P\Delta^{\mathbf{T}}P$ avec $\Delta = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -2 \end{pmatrix}$.

De fait, $M = \alpha I + \beta A + \gamma B = \alpha P^{\mathbf{T}}IP + \beta P^{\mathbf{T}}DP + \gamma P\Delta^{\mathbf{T}}P = P^{\mathbf{T}}(\alpha I + \beta D + \gamma\Delta)P$ avec ${}^{\mathbf{T}}P = P^{-1}$. D'où

$\boxed{{}^{\mathsf{T}}PMP = \alpha I + \beta D + \gamma\Delta = \begin{pmatrix} \alpha+2\beta & 0 & 0 \\ 0 & \alpha-\beta & 0 \\ 0 & 0 & \alpha+\beta-2\gamma \end{pmatrix}}$: c'est une matrice diagonale.

5 — Soit $M = \begin{pmatrix} y+z & y & x \\ y & x+z & y \\ x & y & y+z \end{pmatrix}$.
On remarque que M est combinaison linéaire de I, A et B puisqu'on peut écrire M sous la forme : $M = zI+yA+x(B+I) = (x+z)I + yA + xB$.
D'après l'étude précédente, on sait que pour $M = \alpha I + \beta A + \gamma B$, les valeurs propres de M sont $\alpha+2\beta$, $\alpha-\beta$ et $\alpha+\beta-2\gamma$ (termes situés sur la matrice diagonale semblable à M trouvée dans la question 4).
On en déduit ici que les valeurs propres de M sont les réels $x+z+2y$, $x+z-y$ et $x+z+y-2x$ *i.e.*

$$\boxed{\text{Spec}(M) = \{x+z+2y, x+z-y, z+y-x\}}.$$

■■ *Solution (Exercice 2.6)*

1 — La matrice M étant symétrique réelle, on peut appliquer le théorème spectral : il existe une matrice $P = (E_1 \mid \cdots \mid E_n)$ avec $E_i \in \mathbf{R}^n$ pour tout i, et une matrice diagonale réelle $D = (\lambda_1 \quad \lambda_2 \quad \dots \quad \lambda_n)$ telles que :

$$M = {}^{\mathsf{T}}PDP.$$

Les colonnes $(E_1, \dots, E_4)$ sont orthonormées puisque ${}^{\mathsf{T}}P = P^{-1}$ *i.e.*P est orthogonale.

2 — Par hypothèse, la relation matricielle suivante est vérifiée : $X_{n+1} = MX_n$. Ainsi, pour tout entier $n \in \mathbf{N}$, $X_n = M^nX_0$ par récurrence immédiate.
La relation $\underset{X_0}{\mathcal{M}at}(\mathcal{B}) = \begin{pmatrix} y_1 \\ \dots \\ y_4 \end{pmatrix}$ signifie que :

$$X_0 = \sum_{j=1}^{4} y_j E_j.$$

En remplaçant on trouve :

$$X_n = M^nX_0 = M^n\left(\sum_{j=1}^{4} y_j E_j\right) = \sum_{j=1}^{4} y_j M^n E_j.$$

Mais comme $ME_j = \lambda_j E_j$ nous obtenons par récurrence immédiate : $\forall n \in \mathbf{N}$, $M^nE_j = \lambda_j^n E_j$. En conclusion :

$$\boxed{X_n = \sum_{j=1}^{4} y_j \lambda_j^n E_j}.$$

3 — Les y_k sont les coordonnées de X_0 dans la base propre orthonormée fournie par le théorème spectral (voir première question), donc elles sont données par un produit scalaire :

$$y_k = \langle X_0 | E_k \rangle \quad \boxed{= (y_0 \quad \dots \quad y_4)\, E_k}.$$

4 — **4.1.**
```
import numpy as np
G=np.array([[64,0,0,0],[0,64,22,28],[0,22,-28,-22],[0,28,-22,12]])
M=(1/64)*G
```

On trouve comme éléments propres pour M : (array([-0.73408094, 0.27198286, 1.21209808, 1.]), array([[0. , 0. , 0. , 1.], [-0.2815041 , 0.24811393, 0.92692768, 0.], [0.8476079 , 0.51710739, 0.11899909, 0.], [0.44979583, -0.81916996, 0.35587118, 0.]]))

4.2. On utilise donc l'expression rappelée dans la troisième question, en prenant le produit scalaire entre les E_k trouvés dans la question """""""""précédente, et les $a_i(0)$ pour tout i.

```
VP,VcP=np.linalg.eig(M)
def y(k):
    a=[2,2,4,5]
    return sum([a[i]*VcP[k-1][i] for i in range(4)])
```

Par exmeple, nous trouvons y_1=5.0 avec Python.

4.3.
```
VP,VcP=np.linalg.eig(M)
def a(n):
    L=[2,2,4,5]
    S=y(1)*(VP[0]**n)*VcP[0]
    for k in range(2,5):
        S=S+y(k)*(VP[k-1]**n)*VcP[k-1]
    return S
```

Par exemple pour $n = 100$: [18.9056578 10.78512497 2.85468712 0.2272006].

■■ *Solution (Exercice 2.7)*

1 —

```
def question1(L):
    n=len(L)
    for i in range(n):
        if (L[i]**2)*sum([1/(L[j]**2) for j in range(n)]) < 2:
            return False
    return True
```

Pour `[1,2,3]`, nous obtenons False.

2 — Pour ne pas refaire trois fois le même travail, déterminons l'expression générale $p(x, y, z)$ de la projection orthogonale sur $\mathcal{P}$ du vecteur (x, y, z). Par ailleurs, on vérifie sans peine que $\left(\begin{pmatrix}0\\1\\1\end{pmatrix}, (1,0,0)\right)$ est une base de $\mathcal{P}$ (aucune condition sur x, et les deux autres coordonnées doivent être égales).
Nous avons alors $(x', y', z') = p(x, y, z)$ si et seulement si :

$$\begin{cases} (x,y,z) - p(x,y,z) \quad \perp \mathcal{P} \\ \qquad (x',y',z') \in \mathcal{P} \end{cases} \iff \begin{cases} (x-x', y-y', z-z') \quad \perp \begin{pmatrix}0\\1\\1\end{pmatrix} \\ (x-x', y-y', z-z') \quad \perp (1,0,0) \\ \qquad\qquad y'-z' = 0 \end{cases}$$

$$\iff \begin{cases} y-y'+z-z' &= 0 \\ x-x' &= 0 \\ y'-z' &= 0 \end{cases}$$

$$\iff \begin{cases} y'+z' &= y+z \\ x' &= x \\ y'-z' = 0 \in \mathcal{P} \end{cases}$$

$$\iff \begin{cases} y' = z' &= \frac{y+z}{2} \\ x' &= x. \end{cases}$$

On trouve alors $\boxed{p(u) = (1,0,0),\ p(v) = \sqrt{2}\left(0, \frac{1}{2}, \frac{1}{2}\right) \text{ et } p(w) = \sqrt{2}\left(0, \frac{1}{2}, \frac{1}{2}\right).}$ Les trois vecteurs sont bien de norme 1.

3 — **3.1.** Comme $(\varepsilon_1, \varepsilon_2)$ est une base orthonormée de $\mathcal{P}$, nous avons [3] :

$$\boxed{p(e_i) = \langle e_i|\varepsilon_1\rangle\, \varepsilon_1 + \langle e_i|\varepsilon_2\rangle\, \varepsilon_2.}$$

3.2. La quantité $\langle e_i|\varepsilon_1\rangle^2 + \langle e_i|\varepsilon_2\rangle^2$ vaut exactement $\left\|p(e_i)\right\|^2$. Mais comme $\left\|p(a_i e_i)\right\|^2 = a_i^2 \left\|p(e_i)\right\|^2 = d^2$, nous obtenons la formule donnée.

3.3. Divisons par d^2 l'égalité précédente, et sommons-Là en i pour $i \in [\![1\,,\,3]\!]$. Il vient :

$$\sum_{i=1}^{3} \frac{1}{a_i^2} = \frac{1}{d^2} \sum_{i=1}^{d} \left(\langle e_i|\varepsilon_1\rangle^2 + \langle e_i|\varepsilon_2\rangle^2\right) = \frac{1}{d^2}(\|\varepsilon_1\|^2 + \|\varepsilon_2\|^2) = \frac{1}{d^2},$$

où nous avons appliqué le théorème de Pythagore appliqué à la base orthonormée (e_1, e_2, e_3). C'est l'égalité souhaitée.

3. d'après le cours

3.4. Nous avons, toujours d'après le théorème de Pythagore :

$$\|a_i e_i\|^2 = \|p(a_i e_i)\|^2 + \|a_i e_i - p(a_i e_i)\|^2 \quad \Longrightarrow \quad a_i^2 \|a_i e_i\|^2 \geqslant \|p(a_i e_i)\|^2 = d^2.$$

En prenant la racine de chaque côté nous obtenons l'inégalité souhaitée. Enfin, combinant ceci et la question précédente, il vient :

$$a_i^2 \sum_{k=1}^{3} \frac{1}{a_k^2} \geqslant d^2 \frac{2}{d^2} \quad \boxed{= 2}.$$

■■ *Solution (Exercice 2.8)*

1 —
```
def u(a, b, c, n=100):
    L=[a,b,c]
    d=0
    for k in range(n):
        d=(a+b+c)/3
        a=b
        b=c
        c=d
        L.append(d)
    return L
```

2 — 0 Spec A si et seulement si Ker A $\neq \{0\}$. Une simple résolution de système linéaire donne Ker A $= \begin{pmatrix}0\\0\\0\end{pmatrix}$. Donc $\boxed{0 \notin \text{Spec A.}}$

3 — Méthode du pivot de Gauß.

4 — Notons $Q = X^3 - \frac{1}{3}X^2 - \frac{1}{3}X - \frac{1}{3}$. On a $Q(1) = 0$, donc 1 est racine évidente, on en déduit après calculs :

$$Q = (X-1)(X^2 + \frac{2}{3}X + \frac{1}{3}).$$

Notant $z = \frac{-2+3i\sqrt{-\Delta}}{6}$ avec $\Delta = -\frac{8}{9}$, nous obtenons les racines de Q : $1, z\overline{z}$. Comme $|z|^2 = \frac{1}{3}$, $|z| < 1$.
Au final, nous avons trois valeurs propres réelles distinctes, donc A est diagonalisable sur $\mathbf{R}$, de valeurs propres $1, z, \overline{z}$.
Donc :
Il existe $P \in \mathcal{GL}_n(\mathbf{R})$ telle que

$$A = P \begin{pmatrix}1 & 0 & 0\\ 0 & z & 0\\ 0 & 0 & \overline{z}\end{pmatrix}.$$

5 — Un simple calcul montre que $X_{n+1} = AX_n$. Montrons par récurrence la propriété $\mathcal{P}_n \quad X_n = A^n \times X_0$.
■ ***Initialisation.*** Claire pour $n = 0$, puisque $A^0 X_0 = I_3 X_0$. ■ ***Hérédité.*** Supposons $\mathcal{P}_n$ vraie, pour un certain entier n. Alors montrons que $\mathcal{P}_{n+1}$ est vraie. En effet, $X_{n+1} = AX_n = AA^n X_0 = A^{n+1} X_0$. D'où le résultat par principe de récurrence.

6 — Nous avons

$$\begin{pmatrix}u_n\\ u_{n+1}\\ u_{n+2}\end{pmatrix} = P \begin{pmatrix}1 & 0 & 0\\ 0 & z^n & 0\\ 0 & 0 & \overline{z}^n\end{pmatrix} P^{-1}.$$

Notant $P = \begin{pmatrix}\alpha_1 & \beta_1 & \gamma_1\\ \alpha_2 & \beta_2 & \gamma_2\\ \alpha_3 & \beta_3 & \gamma_3\end{pmatrix}$, nous obtenons en notant aussi t, u, v les coefficients de la première colonne de P^{-1} :

$$u_n = \alpha_1 t + \beta_1 u z^n + \gamma_1 v \overline{z}^n.$$

7 — Puisque $|z| < 1$, nous avons $\lim\limits_{n\to\infty} |bz^n| = 0$. Mais aussi $\lim\limits_{n\to\infty} |c\overline{z}^n| = 0$ puisque $|\overline{z}| < 1$. Donc d'après l'inégalité triangulaire :

$$\lim_{n\to\infty} |bz^n + c\overline{z}^n| = 0.$$

8 — À la question précédente, nous avons donc établi la convergence des suites réelles ci-dessous :

$$\lim_{n\to\infty} \text{Re}\,(u_n - a) = 0, \quad \lim_{n\to\infty} \text{Im}\,(u_n - a) = 0.$$

Or, comme (u_n) est une suite réelle, nous avons : $\forall n \in \mathbf{N}$, $\operatorname{Im}(u_n) = 0$ donc la seconde condition fournit $\operatorname{Im}(a) = 0$. Ainsi $a \in \mathbf{R}$, et la première condition donne

$$\lim_{n\to\infty} \operatorname{Re}(u_n) = u_n = a.$$

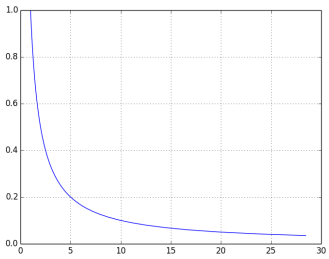
D'où : $\boxed{\lim_{n\to\infty} u_n = a}$.

■■ *Solution (Exercice 2.9)*

1 — Par exemple en adoptant une solution récursive pour créer les produits partiels (autre solution : deux listes que l'on complète progressivement à l'aide d'une boucle `for`).

```
import matplotlib.pyplot as plt
def f_part(n, x):
    if n==1:
        return 1+x
    else :
        return (1+x**n)*f_part(n-1, x)

def g_part(n, x):
    if n==1:
        return 1-x
    else :
        return (1-x**(2*n-1))*g_part(n-1, x)
n=100
plt.plot([f_part(n, k/100) for k in range(81)], [g_part(n, k/100) for k in range(81)])
plt.grid()
```



On conjecture donc que $f_n(x)g_n(x) = 1$ pour tout x et n, *i.e.*le nuage de points tracé est situé sur la courbe $y = 1/x$ ou dit encore autrement $h = 1$.

2 — Le plus simple est ici de former les quotients consécutifs. En effet, pour tout $x \in [0,1]$ et $n \in \mathbf{N}$, nous avons (les dénominateurs étant non nuls) :

$$\frac{f_{n+1}(x)}{f_n(x)} = 1 + x^{n+1} \geqslant 1, \quad \frac{g_{n+1}(x)}{g_n(x)} = 1 - x^{2(n+1)-1} = 1 - x^{2n+1} \leqslant 1.$$

Ainsi $\boxed{\text{la suite } (f_n(x))_{n\geqslant 1} \text{ est croissante, la suite } (g_n(x))_{n\geqslant 1} \text{ est décroissante}}$.

3 — **3.1.** Étude classique de fonction. Notons $g : t \in \mathbf{R} \longmapsto \mathrm{e}^t - t - 1$. Elle est dérivable sur $\mathbf{R}$ et $g'(t) = \mathrm{e}^t - 1 \geqslant 0$ si $t \geqslant 0$ et négatif sinon. Ainsi g est croissante sur $\mathbf{R}^+$ et décroissante sur $\mathbf{R}^-$. Donc comme $g(0) = 0$, la fonction g est positive. D'où l'inégalité : $\boxed{\forall t \in \mathbf{R}, \quad 1 + t \leqslant \mathrm{e}^t}$.
Appliquant cet encadrement à $f_n(x)$ pour tout $n \in \mathbf{N}$ et $x \in [0,1[$, nous obtenons :

$$\prod_{k=1}^{n} 1 \leqslant f_n(x) \leqslant \prod_{k=1}^{n} \mathrm{e}^{x^k} = \exp\left(\sum_{k=1}^{n} x^k\right) = \exp\left(\frac{x}{1-x}\right), \quad \text{car} x < 1.$$

Cet encadrement nous apprend la chose suivante : la suite $(f_n(x))$ est croissante majorée, donc elle converge vers une limite finie notée $f(x)$ dans le sujet. De plus, par passage à la limite, nous obtenons $\boxed{f(x) \leqslant \exp\left(\frac{x}{1-x}\right).}$

3.2. La fonction f est continue en zéro, puisque $f(0) = \lim_{n\to\infty} \prod_{k=1}^{n}(1+0) = 1$, et d'autre part, faisant $x \longrightarrow 0$ dans l'encadrement établi :

$$1 \leqslant \lim_{x\to 0} f(x) \leqslant \exp(0) = 1.$$

Donc $f(0) = \lim_0 f$ et $\boxed{f \text{ est continue en zéro.}}$

4 — **4.1.** Cette fois-ci c'est plus simple, puisque $(g_n(x))_n$ est décroissante, donc puisqu'elle est minorée par zéro, nous $\boxed{\text{obtenons l'existence de } g(x)}$ par théorème de convergence monotone.

4.2. Considérons $h(x) = (1-x)^t$ pour $x \in [0,1[$ à $t \in [0,1[$ fixé. Le membre de gauche de l'inégalité n'est finalement rien d'autre que $h(0) - h(x)$. La fonction h est continue sur $[0,x]$ (car $x < 1$), et dérivable sur $]0,x[$ de dérivée $h'(x) = -t(1-x)^{t-1}$, donc d'après l'égalité des accroissements finis :

$$\exists c_x \in]0,x[, \quad h(0) - h(x) = h'(c_x)(-x) = xt(1-c_x)^{t-1} \geqslant xt$$

car $(1-c_x)^{t-1} \geqslant 1$ ($t \leqslant 1$ et $1-c_x \leqslant 1$). D'où l'inégalité de l'énoncé : $\boxed{1-(1-x)^t \geq xt.}$

4.3. D'après la question **3.1** nous obtenons

$$g(x) \leqslant \lim_{n\to\infty} \prod_{k=1}^{n} \exp\left(-x^{2k-1}\right) = \lim_{n\to\infty} \exp\left(-\sum_{k=1}^{n} x^{2k-1}\right) = \lim_{n\to\infty} \exp\left(-\frac{1}{x}(x^2)\frac{1-x^{2n}}{1-x^2}\right) = \exp\left(-\frac{x}{1-x^2}\right).$$

D'après la question précédente, nous avons $1-xt \geqslant (1-x)^t$ donc $1-x^{2k-1} \geqslant (1-x)^{x^{2k-2}}$. On déduit alors :

$$g(x) \geqslant \lim_{n\to\infty} \prod_{k=1}^{n} (1-x)^{x^{2k-2}} = \lim_{n\to\infty} (1-x)^{\sum_{k=1}^{n} x^{2k-2}} = \lim_{n\to\infty} (1-x)^{\frac{1}{x^2}x^2\frac{1-x^{2n}}{1-x^2}} = (1-x)^{\frac{1}{1-x^2}} = \exp\left(\frac{1}{1-x^2}\ln(1-x)\right).$$

On obtient l'encadrement de l'énoncé. Puisque $g_n(0) = 1$, nous avons aussi $g(0) = 1$. Et en faisant $x \longrightarrow 0$ dans l'encadrement précédent nous obtenons :

$$\exp(\ln(1)) = 1 \leqslant \lim_{x\to 0} g(x) \leqslant \exp(0) = 1,$$

donc $\boxed{g(0) = \lim_{x\to 0} g(x) = 1,}$ g est continue en zéro.

5 — 5.1.

$$\begin{aligned} f_n(x^2)g_n(x^2) = f_n(x) &= \prod_{k=1}^{n}(1+x^{2k})\prod_{k=1}^{n}(1-x^{4k-2}) \\ &= \prod_{k=1}^{n}(1+x^{2k})(1-x^{2k-1})(1+x^{2k-1}) \\ &= \left(\prod_{k=1}^{n}(1+x^{2k})(1+x^{2k-1})\right)\left(\prod_{k=1}^{n}(1-x^{2k-1})\right) \\ &\boxed{= f_{2n(x)}g_n(x).} \end{aligned}$$

Il reste ensuite à faire $n \longrightarrow \infty$. Nous obtenons :

$$h(x^2) = \lim_{n\to\infty} f_{2n}(x)\lim_{n\to\infty} g_n(x) = \lim_{n\to\infty} f_n(x)\lim_{n\to\infty} g_n(x) = h(x).$$

5.2. On montre le résultat par récurrence. ■ ***Initialisation.*** Pour $n = 1$, c'est la question précédente. ■ ***Hérédité.***

Supposons que $h(x) = h(x^{2^n})$ pour un entier $n \in \mathbf{N}$ et $x \in [0,1[$. Alors $h(x) = h(x^{2^n}) = h((x^{2^n})^2) = h(x^{2^{n+1}})$. C'est la propriété au rang suivant.

Donc $\boxed{\forall n \in \mathbf{N},\quad h(x) = h(x^{2^n}).}$

Ainsi en faisant $n \longrightarrow \infty$ dans l'égalité précédente on obtient par continuité de h en zéro :

$$\boxed{h(0) = h(x),\quad \forall x \in [0,1[,} \quad \text{car} \quad x^{2n} \underset{n\to\infty}{\longrightarrow} 0.$$

5.3. La fonction h est donc constante égale à $1 = h(0)$, on retrouve la conjecture initiale !

■■ *Solution (Exercice 2.10)*

1 — 1.1. La fonction f est dérivable sur $[1;\infty[$ comme quotient de fonctions dérivables. De plus,

$$\forall x \in [1;\infty[,\quad f'(x) = \frac{2\ln x - \ln^2(x)}{x^2} = \frac{\ln x(2-\ln(x))}{x^2}.$$

Le signe de f' est donc celui de $x \in [1;\infty[\longmapsto 2-\ln(x)$, et de plus, par croissances comparées, $f(x) \underset{x\to\infty}{\longrightarrow} 0$. On déduit alors le tableau ci-dessous.

x	1		e^2		$+\infty$
$f(x)$	0	↗	$\frac{4}{e^2}$	↘	0

1.2. L'équation (E_1) est finalement $f(x) = 1$. Comme $\frac{4}{e^2} \approx 0.54 < 1$, par simple lecture du tableau *supra* on déduit que (E_1) n'admet pas de solution.

1.3. Supposons $n \geqslant 2$, alors $\frac{1}{n} \leqslant \frac{1}{2}$ donc $\frac{1}{n} \in \left[0; f(e^2)\right]$. Puisque f est continue sur $[1;\infty[$ en tant que quotient de fonctions continues sur $[1;\infty[$, et que :

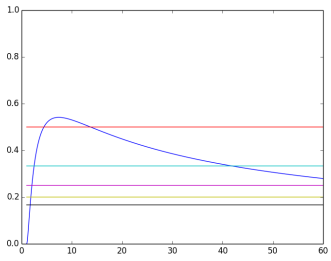
$$f(1) < \frac{1}{n} < \frac{4}{e^2}, \quad \frac{4}{e^2} > \frac{1}{n} > \lim_{x\to\infty} f(x),$$

on déduit par le théorème des valeurs intermédiaires l'existence de α_n et β_n solutions telles que $1 \leqslant \alpha_n \leqslant e^2 \leqslant \beta_n$.

2 — Choisissons de le faire par exemple avec python.

```
import matplotlib.pyplot as plt
import numpy as np
x=np.linspace(1,60,10000)
f=lambda x:(np.log(x)**2/x)
y=f(x)
plt.plot(x,y)
for i in range(1,7):
    plt.plot(x,[(1/i) for k in x])
plt.show()
```

On retrouve les faits précédents sur le graphe ci-dessous : aucune solution lorsque $i = 1$, et deux solutions si $i \geqslant 2$.



3 — On conjecture alors :

✓ la suite (α_n) semble décroissante,

✓ la suite (β_n) semble croissante.

4 — **4.1.** Puisque $f(\beta_n) = \frac{1}{n}$ et $f(\beta_{n+1}) = \frac{1}{n+1}$ pour tout entier $n \in \mathbf{N}$, on a donc $f(\beta_{n+1}) < f(\beta_n)$. Puisque f est décroissante sur $[e^2;\infty[$, nous avons alors $\beta_{n+1} \geqslant \beta_n$. La suite (β_n) est alors croissante.

4.2. La suite (β_n) est décroissante donc (β_n) converge vers une limite finie ou diverge vers $+\infty$.
Supposons qu'elle converge vers une limite finie notée $\ell_1 \in \mathbf{R}$ dans la suite. En passant à la limite dans l'égalité $f(\beta_n) = \frac{1}{n}$, et en utilisant la continuité de f, on obtient $f(\ell_1) = 0$. Donc forcément $\ln(\ell_1) = 0$ puis $\ell_1 = 1$. C'est une contradiction car (β_n) est minorée par $e^2 > 1$. Donc : $\beta_n \underset{n\to\infty}{\longrightarrow} \infty$.

4.3. Par hypothèse (relation vérifiée par β_n), nous avons pour tout $n \geqslant 1$:

$$\ln^2(\beta_n) = \frac{\beta_n}{n} = u_n \iff \ln^2(nu_n) = u_n.$$

Calculons $\frac{u_n}{\ln^2(n)}$. Nous avons

$$\frac{u_n}{\ln^2(n)} = \frac{\ln^2(nu_n)}{\ln^2(n)} = \left(\frac{\ln n + \ln u_n}{\ln n}\right)^2 = \left(1 + \frac{\ln u_n}{\ln n}\right)^2.$$

Or $\frac{\ln u_n}{\ln n} \underset{n\to\infty}{\longrightarrow} 0$ par hypothèse. Ainsi : $u_n \underset{n\to\infty}{\sim} \ln^2(n)$.

4.4. Comme $\beta_n = nu_n$ pour tout entier n, nous déduisons $\beta_n \underset{n\to\infty}{\sim} n\ln^2(n)$.

5 — On réadapte les points précédents à cette nouvelle suite.

5.1. La suite (α_n) est cette fois-ci décroissante, puisque $f(\alpha_n) = \frac{1}{n}$ et $f(\alpha_{n+1}) = \frac{1}{n+1}$, on a donc $f(\alpha_{n+1}) < f(\alpha_n)$. Puisque f est croissante sur $[1;e^2[$, nous avons alors $\alpha_{n+1} \leqslant \alpha_n$. La suite (α_n) est alors décroissante.

Elle est de plus minorée par un, donc converge vers une limite finie notée $\ell_2 \in \mathbf{R}$. Nous avons, de manière similaire à précédemment (en passant à la limite et en invoquant la continuité de f) la relation $f(\ell_2) = 0$. Donc $\ell_2 = 1$.
Donc $\boxed{\alpha_n \underset{n\to\infty}{\longrightarrow} 1.}$

5.2. On a : $\ln^2(\alpha_n) = \frac{\alpha_n}{n}$ d'où : $n\ln^2(\alpha_n) = \alpha_n$. Cette fois-ci, comme $\alpha_n \underset{n\to\infty}{\longrightarrow} 1$, on peut faire un développement limité du membre de gauche. En effet, $\ln(u) \underset{u\to 1}{\sim} u-1$ donc

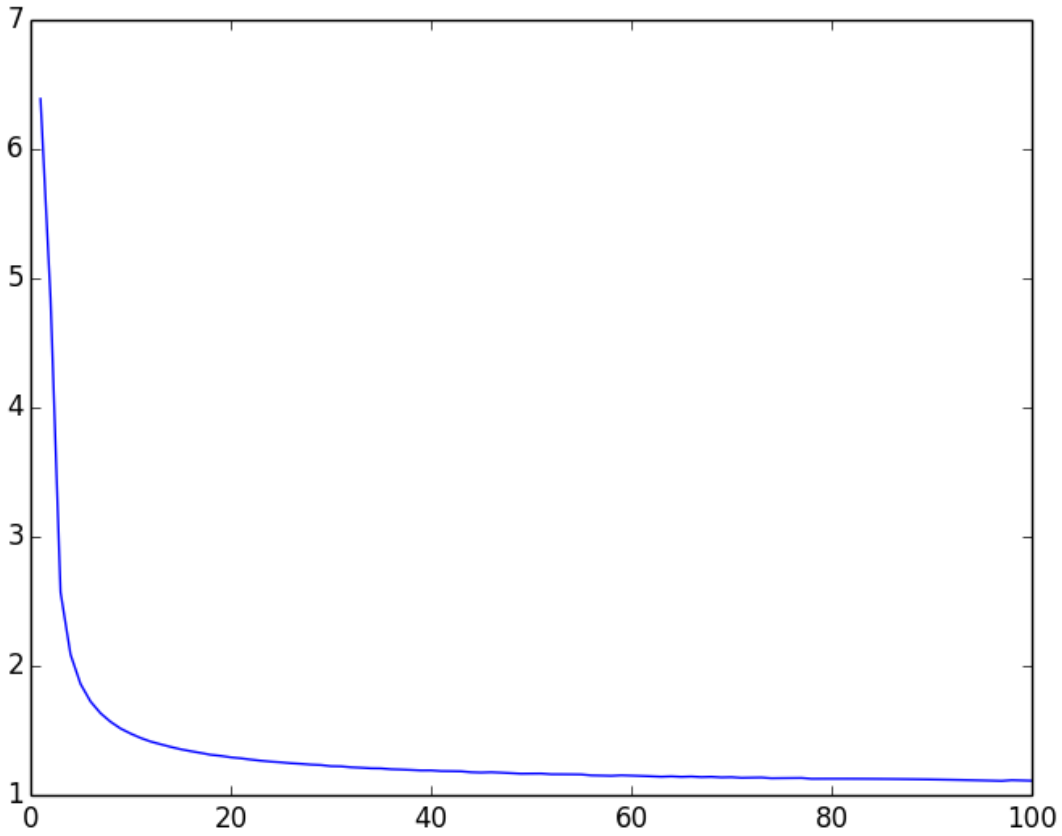
$$n(\alpha_n - 1)^2 \underset{n\to\infty}{\sim} \alpha_n \underset{n\to\infty}{\sim} 1.$$

Donc $\boxed{\alpha_n - 1 \underset{n\to\infty}{\sim} \frac{1}{\sqrt{n}}}$ car $\alpha_n \geqslant 1$ pour tout n.

python Le premier point consiste à écrire l'algorithme de Dichotomie pour avoir une bonne valeur approchée de α_n pour tout entier $n \in \mathbf{N}$. Ensuite, on vérifie la convergence vers 1 de la suite $(\sqrt{n}(\alpha_n - 1))$ par exemple en traçant les termes successifs.

```
def dicho(a,b,f,prec):
    while b-a>prec:
        c = (a+b)/2
        if f(a)*f(c) <= 0:
            b=c
        else:
            a=c
    return (a+b)/2

import matplotlib.pyplot as plt
import numpy as np
f=lambda x:(np.log(x)**2)/x
N=np.linspace(1,100,100)
L=[np.sqrt(k)*(dicho(1,np.exp(1)**2,lambda x:(f(x)-1/k),0.001)-1) for k in range(1,101)]
plt.plot(N,L)
```



■■ *Solution (Exercice 2.11)*

1 — **1.1.**
```
import numpy as np
def Sn(x,n):
    L=[np.exp(-x*np.sin((k*np.pi)/n)) for k in range(n)]
    return (1/n)*np.sum(L)
```
Pour `Sn(3,10)` nous obtenons 0.247607821352.

1.2.
```
import matplotlib.pyplot as plt
def graphe(a,b,n):
    x=np.linspace(a,b,1000)
    y=[Sn(X,n) for X in x]
    plt.plot(x,y)
    plt.show()
```

2 — Puisque S_n est une somme de fonctions dérivables, elle est elle-même dérivable également. De plus, pour tout $x \in \mathbf{R}$:

$$S_n'(x) = -\frac{1}{n}\sum_{k=0}^{n-1} \sin\left(\frac{k\pi}{n}\right) e^{-x\sin\left(\frac{k\pi}{n}\right)}.$$

Ainsi : $S_n'(0) = -\frac{1}{n}\sum_{k=0}^{n-1}\sin\left(\frac{k\pi}{n}\right) = -\frac{1}{n}\mathrm{Im}\left(\sum_{k=0}^{n-1}\left(e^{i\frac{\pi}{n}}\right)^k\right) = -\frac{1}{n}\mathrm{Im}\left(\frac{1-e^{i\pi}}{1-e^{i\frac{\pi}{n}}}\right)$.

En mettant l'angle moitié en facteur, on trouve :

$$\boxed{S_n'(0) = -\frac{2}{n}\,\mathrm{Im}\,\frac{1}{-2i\sin(\pi/2n)} = -\frac{1}{n\sin(\pi/2n)}}.$$

3 — Il s'agit simplement d'une somme de Riemann du type $\frac{1}{n}\sum_{k=0}^{n-1} g\left(\frac{k}{n}\right)$ avec g définie ci-après : ainsi, puisque $g : x \longmapsto e^{-\sin(x)}$ est continue, nous avons $\boxed{S_n(x) \underset{n\to\infty}{\longrightarrow} f(x)}$ pour tout réel x.

4 — 4.1. La fonction $g_t : x \longmapsto \mathrm{e}^{-x\sin t}$ est, pour tout t, décroissante. Donc $f(x) \leqslant f(y)$ en intégrant l'inégalité $g_t(x) \geqslant g_t(y)$ pour sur $t \in [0;\pi]$.

4.2. Il s'agit donc de montrer que :

$$\int_0^{\pi/2} \mathrm{e}^{-x\sin t}\,\mathrm{d}t = \int_{\pi/2}^{\pi} \mathrm{e}^{-x\sin t}\,\mathrm{d}t.$$

Le résultat découle ensuite de la relation de Chasles. Dans la première intégrale, faisons le changement de variable affine $u = \pi - t$ [5], possible car la fonction $t \longmapsto \pi - t$ est de classe C^1 [6]. On obtient :

$$\int_{\pi/2}^{\pi} \mathrm{e}^{-x\sin t}\,\mathrm{d}t = \int_{\pi/2}^{0} \mathrm{e}^{-x\sin(u)}(-\mathrm{d}u) = \int_0^{\pi/2} \mathrm{e}^{-x\sin(u)}\,\mathrm{d}u.$$

4.3. Simple étude de fonction. Considérons la fonction h définie pour $t \in [0;\frac{\pi}{2}]$ par $h(t) = \sin t - \frac{2}{\pi}t$. Alors h est dérivable et $h'(t) = \cos t - \frac{2}{\pi}$. La fonction h' est décroissante, donc
La minoration précédente fournit, en intégrant et en multipliant par x positif de chaque côté :

$$f(x) \leqslant \int_0^{\pi/2} -\frac{2x}{\pi}t\,\mathrm{d}t = -\frac{2x}{\pi}\int_0^{\pi/2} t\,\mathrm{d}t.$$

Le membre de droite tend vers $-\infty$ lorsque x tend vers $+\infty$. Donc par théorème de majoration : $\boxed{\lim\limits_{x\to\infty} f(x) = -\infty.}$

■■ *Solution (Exercice 2.12)*

1 — *(Résolution de* (E)*)*

1.1. Notons $I_1 =]0,+\infty[$ et $I_2 =]-\infty,0[$. En divisant par x, on se ramène à une équation linéaire de première année sur l'un des intervalles I_1, I_2.

$$y'(x) + \frac{2}{x}y(x) = \frac{x}{1+x^2}.$$

On va raisonner sur I_1 par exemple. La fonction $a : x \mapsto \frac{2}{x}$ est continue sur I_1, une primitive est donnée par $A : x \mapsto 2\ln|x|$. Le cours nous dit alors que la solution générale de l'équation homogène sur I_1 est donnée par :

$$\forall x \in I_1, \quad y_0(x) = \lambda_1 e^{-2\ln|x|} = \frac{\lambda_1}{x^2}$$

avec $\lambda_1 \in \mathbf{R}$. On cherche ensuite une solution particulière par variation de la constante. On pose donc $y(x) = \frac{\lambda(x)}{x^2}$ avec λ une fonction dérivable à déterminer, nous avons ensuite

$$\begin{aligned} y'(x) &= \frac{-2}{x^3}\lambda(x) + \frac{\lambda'(x)}{x^2} \\ y'(x) + \frac{2}{x}y(x) &= \frac{-2}{x^3}\lambda(x) + \frac{\lambda'(x)}{x^2} + \frac{2}{x^3}\lambda(x) \\ &= \frac{\lambda'(x)}{x^2}. \end{aligned}$$

Nous avons donc la relation

$$\begin{aligned} \frac{\lambda'(x)}{x^2} &= \frac{x}{1+x^2} \\ \lambda'(x) &= \frac{x^3}{1+x^2} \end{aligned}$$

Il nous reste à chercher une primitive de λ, on peut par exemple prendre $\forall x \in I_1$:

$$\begin{aligned} \lambda(x) &= \int_0^x \frac{t^3}{1+t^2}\,\mathrm{d}t \\ &= \int_0^x \frac{t\left(1+t^2\right) - t}{1+t^2} \\ &= \int_0^x t\,\mathrm{d}t - \int_0^x \frac{t}{1+t^2}\,\mathrm{d}t \\ &= \frac{x^2}{2} - \frac{1}{2}\ln\left(1+x^2\right). \end{aligned}$$

5. ce qui permet d'exploiter la symétrie (en t) par rapport à l'axe $x = \frac{\pi}{2}$ de la fonction intégrée
6. Attention, pas d'hypothèse de bijectivité, l'intégrale n'est pas généralisée !

La solution générale est enfin somme de la solution générale de l'équation homogène et de la solution particulière obtenue :

$$\exists \lambda_1,\ \forall x \in I_1 \quad y(x) = \frac{\lambda_1}{x^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln\left(1+x^2\right)}{x^2}.$$

On fait la même étude sur I_2 pour obtenir finalement :

$$\boxed{\exists (\lambda_1,\lambda_2) \in \mathbf{R}^2,\ y(x) = \begin{cases} \frac{\lambda_1}{x^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln(1+x^2)}{x^2} & \text{si } x > 0 \\ \frac{\lambda_2}{x^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln(1+x^2)}{x^2} & \text{si } x < 0 \end{cases}}$$

1.2. On remarque qu'au voisinage de 0,

$$\frac{\ln\left(1+x^2\right)}{x^2} \underset{x\to 0}{\sim} \frac{x^2}{x^2} = 1$$

Ainsi, on voit facilement que si $\lambda \neq 0$

$$\lim_{x\to 0} \frac{\lambda_1}{x^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln\left(1+x^2\right)}{x^2} = \pm\infty$$

et si $\lambda = 0$

$$\lim_{x\to 0} \frac{1}{2} - \frac{1}{2}\frac{\ln\left(1+x^2\right)}{x^2} = 0$$

Ainsi dans ce seul cas on peut prolonger notre solution par continuité en 0 en posant $y(0) = 0$.
Nous posons donc

$$y(x) = \begin{cases} \frac{1}{2} - \frac{1}{2}\frac{\ln(1+x^2)}{x^2} & \text{si } x \neq 0 \\ 0 & \text{si } x = 0 \end{cases}$$

Il nous faut encore vérifier la dérivabilité en 0.

Pour justifier la dérivabilité et positionner la courbe par rapport à sa tangente, on peut réaliser un développement limité de y en 0, nous savons qu'une fonction est dérivable si et seulement si elle admet un développement limité d'ordre 1, et en regardant le signe du terme non nul suivant, nous pourrons en déduire la position relative.

$$\begin{aligned} y(x) &= \frac{1}{2} - \frac{1}{2}\frac{x^2 - \frac{x^4}{2} + x^4\varepsilon(x)}{x^2} \\ &= \frac{1}{2} - \frac{1}{2}\left(1 - \frac{x^2}{2} + x^2\varepsilon(x)\right) \\ &= x^2\left(\frac{1}{4} + \eta(x)\right). \end{aligned}$$

On déduit donc l'existence du développement, nous remarquons de plus que, dans ce développement, le coefficient de x est nul donc $\boxed{y'(0) = 0}$, d'où $y = 0$ comme tangente. Enfin comme $\frac{1}{4} > 0$ et $\lim_{x\to 0} \eta(x) = 0$ nous voyons que localement, la courbe est dessus sa tangente en 0.

1.3. Résoudre (E) avec la condition de Cauchy $y(1) = \frac{1}{2}$ sur $I_1 =]0, +\infty[$.
Il nous faut déterminer la constante λ pour que $y(1) = \frac{1}{2}$

$$\begin{aligned} \frac{\lambda}{1^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln 2}{1} &= \frac{1}{2} \\ \lambda &= \frac{\ln 2}{2} \end{aligned}$$

$$\boxed{\text{La solution du problème de Cauchy est } x \mapsto \frac{\ln 2}{2x^2} + \frac{1}{2} - \frac{1}{2}\frac{\ln(1+x^2)}{x^2}.}$$

1.4.
```
import numpy as np
import matplotlib.pyplot as plt
def solexacte(b):
    N=100
    X=np.linspace(1,b,N)
    Y=[]
    for x in X:
        Y.append(np.log(2)/(2*x*x)+(1/2)*(1-np.log(1+x*x)/(x*x)))
    plt.plot(X,Y)
    return X,Y
```

2 — Rappelons le principe de la méthode.[7] L'équation différentielle est du type :

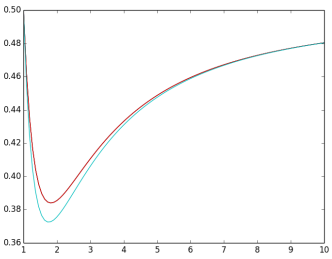
$$y'(x) = F(x, y(x)) \quad \text{où} \quad F(x,z) = \frac{x}{1+x^2} - \frac{2}{x}z.$$

Si y est solution, alors $y(x+h) \sim y(x) + hy'(x) = y(x) + hy'(x)F(x, y(x))$. On définit donc une suite de points y_i par récurrence en posant :

$$\forall k \in [\![0\,,\,N]\!], \quad y_{i+1} = y_i + h \times F(t_i, y_i) = y_i + h \times \frac{t_i}{1+t_i^2} - \frac{2}{t_i}y_i = y_i\left(1 - \frac{2}{t_i}\right) + h \times \frac{t_i}{1+t_i^2}.$$

```
import numpy as np
import matplotlib.pyplot as plt
def solexacte(b):
    N=100
    X=np.linspace(1,b,N)
    Y=[]
    for x in X:
        Y.append(np.log(2)/(2*x*x)+(1/2)*(1-np.log(1+x*x)/(x*x)))
    plt.plot(X,Y)
    return X,Y
plt.plot(solexacte(10)[0],solexacte(10)[1])

def f(x,y):
    return x/(1+x**2)-(2/x)*y
def soleuler(N,b):
    h=(b-1)/N
    X=[1]
    x=1
    Y=[1/2]
    y=1/2
    for i in range(N):
        y=y+h*f(x,y)
        Y=Y+[y]
        x=x+h
        X=X+[x]
    return X,Y
plt.plot(soleuler(100,10)[0],soleuler(100,10)[1])
```



■■ *Solution (Exercice 2.13)*

1 — Pour n fixé f_n est dérivable sur $\mathbf{R}$ (c'est un polynôme) et $f_n'(x) = 3nx^2+n^2 > 0$ pour tout x, donc $\boxed{f_n \text{ strictement croissante sur } \mathbf{R}}$. La limite de f_n en $-\infty$ est $-\infty$, et en $+\infty$ c'est $+\infty$. De plus f_n est continue sur $\mathbf{R}$, donc d'après le théorème de la bijection, f_n s'annule une unique fois : $\boxed{\text{Il existe un unique réel } a_n \text{ tel que } f_n(a_n) = 0}$ De plus $f_n(0) = -2 < 0 \Rightarrow \boxed{0 < a_n}$ car f_n strictement croissante et $f_n(a_n) = 0$.

1.1. Pour démarrer le principe de dichotomie, on a besoin aussi d'un majorant de a_n, or $f_n(1) = n+n^2-2 \geqslant 1+1-2 = 0$ car $n \geqslant 1$, donc $\boxed{1 \geqslant a_n}$.

```
def f(n,x):
    return n*x**3+n*n*x-2

def dicho(a,b,n,prec):
    while b-a>prec:
        c = (a+b)/2
        if f(n,a)*f(n,c) <= 0:
            b=c
        else:
            a=c
    return (a+b)/2
```

Nous obtenons alors 0.03125 pour $n = 10$ et une précision $1/10$.

1.2. Pour n fixé, $f_{n+1}(a_n) = (n+1)a_n^3 + (n+1)^2 a_n - 2 = \underbrace{na_n^3 + n^2 a_n - 2}_{=f_n(a_n)=0} + \underbrace{a_n^3 + (2n+1)a_n}_{>0 \text{ car } a_n>0} > 0 = f_{n+1}(a_{n+1})$.

7. Dans l'énoncé originel, le principe de la méthode était rappelé.

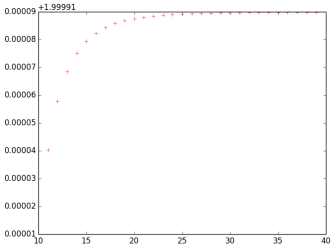
Donc $a_n > a_{n+1}$ puisque la fonction f_{n+1} est strictement croissante sur $\mathbf{R}$, et ceci pour tout $n \geqslant 1$. On en déduit que $\boxed{\text{la suite } (a_n)_{n\geqslant 1} \text{ est décroissante, et minorée par 0, donc convergente.}}$

De plus $f_n(a_n) = 0 \iff na_n^3 + n^2 a_n - 2 = 0 \iff a_n = \frac{2}{n}\left(\frac{1}{a_n^2+n}\right)$. Or quand $n \to +\infty$, $a_n \to \ell \in \mathbf{R}$ donc $a_n^2 + n \to +\infty \Rightarrow \frac{1}{a_n^2+n} \to 0$ et $\boxed{a_n \to 0 \text{ donc } \ell = 0}$.

2 — 2.1. Si l'on garde la précision de 10^{-3} utilisé au départ dans le programme de dichotomie, on obtient une représentation graphique incohérente des u_n, on va donc demander une précision `eps` supérieure.

```
import matplotlib.pyplot as plt
def f(n,x):
    return n*x**3+n*n*x-2

def dicho(a,b,n,prec):
    while b-a>prec:
        c = (a+b)/2
        if f(n,a)*f(n,c) <= 0:
            b=c
        else:
            a=c
    return (a+b)/2
```




```
eps= 10**(-12)
X=[n for n in range(10,41)]
Y=[n*n*dicho(0,1,n,eps) for n in X]
plt.plot(X,Y,'+r')
```

On « conjecture » que la suite $(u_n)_{n\geqslant 1}$ converge vers 2.

2.2. $\forall n \geqslant 1 \quad u_n = n^2 a_n$ donc $f_n(a_n) = 0 \iff \frac{u_n}{n} a_n^2 + u_n - 2 = 0 \iff u_n\left(1 + \frac{a_n^2}{n}\right) = 2$. Comme $\frac{a_n^2}{n} \underset{n\to\infty}{\longrightarrow} 0$, nous obtenons : $\boxed{u_n \underset{n\to\infty}{\longrightarrow} 2.}$ Ainsi, nous obtenons l'équivalent : $\boxed{u_n \underset{n\to\infty}{\sim} \frac{2}{n}.}$

3 — 3.1. Soit $g(x) = \frac{2x^3+1}{3x^2+2}$. On peut remarquer qu'elle est définie et dérivable sur $\mathbf{R}$. De plus, pour tout $x \in \mathbf{R}$:

$$g'(x) = \frac{6x^2(3x^2+2) - 6x(2x^3+1)}{(3x^2+2)^2} = \frac{6x}{(3x^2+2)^2} \underbrace{(x^3 + 2x - 1)}_{h(x)}.$$

La fonction h est donc dérivable sur $\mathbf{R}$ et $h'(x) = 3x^2 + 2 > 0$ donc h est strictement croissante sur $\mathbf{R}$, donc sur $[0,1]$. Or $a_2 \in [0,1]$ et $h(a_2) = a_2^3 + 2a_2 - 1 = \frac{1}{2} \underbrace{(2a_2^3 + 4a_2 - 2)}_{=0 \text{ par definition de } a_2}$, donc g' du signe de h sur $[0,1]$ et est donc négative avant a_2 et $\boxed{g' \text{ est positive sur } [a_2,1]}$, donc $\boxed{g \text{ est croissante sur } [a_2,1]}$.

3.2. On remarque de plus que $g(1) = \frac{3}{5} < 1$, $g(a_2) - a_2 = \frac{2a_2^3+1}{3a_2^2+2} - a_2 = \frac{-a_2^3-2a_2+1}{3a_2^2+2} = 0$ donc $g(a_2) = a_2$. Donc $g([a_2,1]) \subset [a_2,1]$ et g est croissante et continue sur cet intervalle. Nous obtenons successivement :

❶ $x_n \in [a_2,1]$ pour tout entier $n \in \mathbf{N}$ par stabilité de $[a_2,1]$ par g,

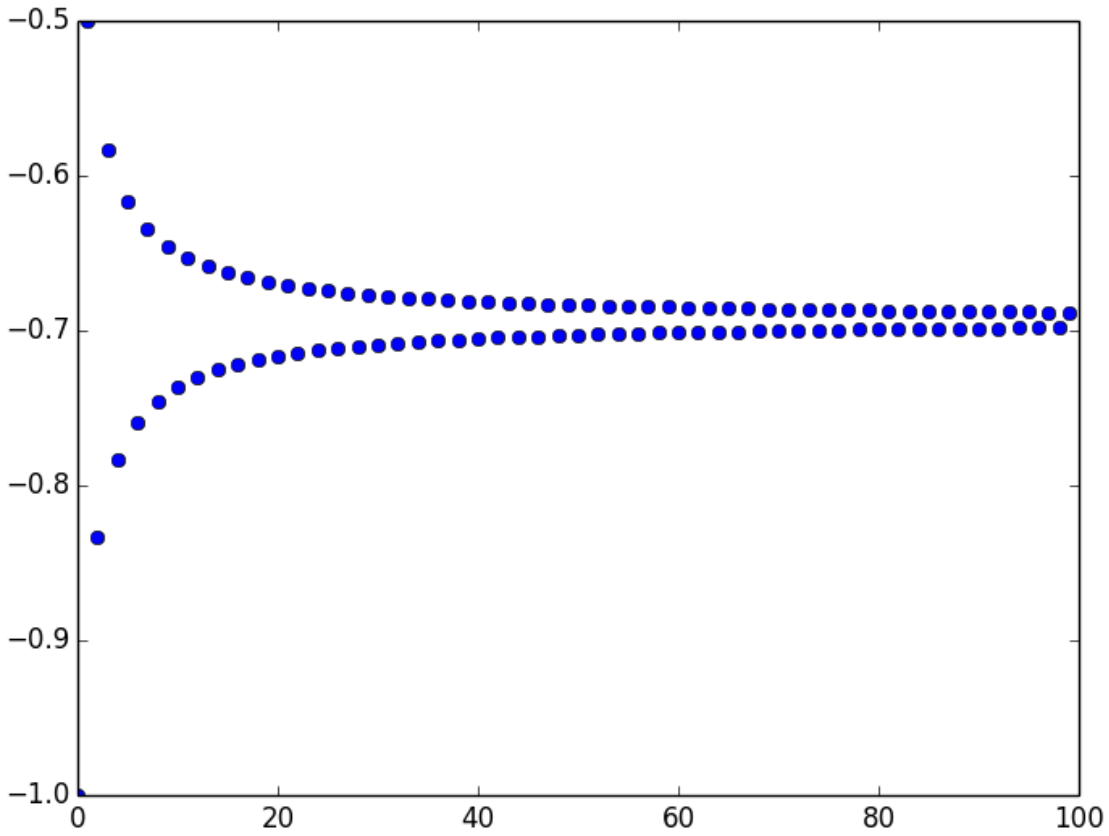
❷ $x_1 = g(x_0) = \frac{3}{5} < x_0$, donc par récurrence (en utilisant la croissance de g), on peut montrer que : $\boxed{\forall n,\ x_{n+1} < x_n, \text{ c'est-à-dire que } (x_n) \text{ est décroissante.}}$ De plus cette suite est bornée, donc elle converge, vers x_ℓ vérifiant $g(x_\ell) = x_\ell$ par continuité de g. Or l'unique solution dans $[0,1]$ de cette équation est a_2. En conclusion : $\boxed{(x_n) \text{ converge vers } a_2}$

■■ *Solution (Exercice 2.14)*

1 — 1.1. Testons par exemple avec la suite $\left(\frac{1}{n}\right)_{n\in\mathbf{N}^*}$.

```
import numpy as np
import matplotlib.pyplot as plt
n=100
x = [i for i in range(0,n)]
y = [-1]
for k in range(2,n+1):
```

```
    y.append(y[-1]+((-1)**k)*(1/k))
plt.plot(x,y,"o")
```



1.2. Soit $n \in \mathbf{N}$.

✓ $S_{2n+2} - S_{2n} = u_{2n+2} + (-1)^{2n+1}u_{2n+1} = u_{2n+2} - u_{2n+1} \geqslant 0$,

✓ $S_{2n+3} - S_{2n+1} = -u_{2n+3} - u_{2n+2} \leqslant 0$ puisque (u_n) est une suite décroissante.

✓ $|S_{2n+1} - S_{2n}| = \left|(-1)^{2n+1}u_{2n+1}\right| \underset{n\to\infty}{\longrightarrow} 0$.

Ainsi, les suites (S_{2n}) et (S_{2n+1}) sont adjacentes, par conséquent
elles convergent vers une même limite ℓ.

1.3. D'après le cours, si les suites extraites paires et impaires convergent vers une même limite, alors c'est le cas de la suite globale. Donc (S_n) converge vers ℓ.

1.4. Ce résultat est compris dans le théorème de convergence des suites adjacentes.

2 — 2.1. Pour tout $x \in [0,1]$, nous avons $\frac{x^n}{1+x} \leqslant \frac{x^n}{1}$. Donc $\int_0^1 \frac{x^n}{1+x}\,dx \leqslant \int_0^1 x^n\,dx = \frac{1}{n+1} \underset{n\to\infty}{\longrightarrow} 0$.

Par théorème de majoration, puisque la quantité étudiée est positive, nous obtenons : $\lim_{n\to\infty} \int_0^1 \frac{x^n}{1+x}\,dx = 0$.

2.2. Remarquons que :

$$\int_0^1 x^{b+k}\,dx = \frac{1}{k+b+1}.$$

Ainsi :

$$\sum_{k=0}^{n-1} \frac{(-1)^k}{k+b+1} = \sum_{k=0}^{n-1} (-1)^k \int_0^1 x^{b+k}\,dx = \int_0^1 x^b \sum_{k=0}^{n-1} (-x)^k\,dx = \int_0^1 x^b \frac{1-(-x)^n}{1+x}\,dx$$

d'après la linéarité de l'intégrale et la formule de somme de termes d'une suite géométrique. Ainsi, nous déduisons :

$$\sum_{k=0}^{n-1} \frac{(-1)^k}{k+b+1} = \int_0^1 \frac{x^b}{1+x}\,dx - (-1)^n \int_0^1 \frac{x^{n+b}}{1+x}\,dx.$$

Le second terme converge vers zéro d'après la question précédente, et car

$$\left|(-1)^n \int_0^1 \frac{x^{n+b}}{1+x}\,dx\right| \leqslant \int_0^1 \frac{x^{n+b}}{1+x}\,dx \underset{n\to\infty}{\longrightarrow} 0.$$

D'où le résultat :

$$\boxed{\sum_{k=0}^{n-1} \frac{(-1)^k}{k+b+1} \underset{n\to\infty}{\longrightarrow} \int_0^1 \frac{x^b}{1+x}\,dx}.$$

2.3. Faire simplement $b = 1$ dans ce qui précède, et calculer l'intégrale.

2.4.
```
import numpy as np
def approxln(n):
    return sum([(-1)**k/k for k in range(1,n+1)])
```
Pour $n = 1000$ cela donne -0.69264743056, alors que la valeur exacte est -0.69314718056.

■■ *Solution (Exercice 2.15)* Pour tout n entier naturel non nul, on note :

E_n l'événement : « les n premiers tirages ont eu lieu et n'ont donné que des boules noires. »
N_n l'événement : « le n-ième tirage a lieu et donne une boule noire. »
B_n l'événement : « le n-ième tirage a lieu et donne une boule blanche. »
Donc $E_n = N_1 \cap N_2 \ldots \cap N_n$.
Soit X la variable aléatoire égale au rang du tirage auquel on a obtenu une boule blanche si on finit par obtenir une boule blanche et égale à 0 sinon.

1 — Si $c = 0$, on effectue des tirages successifs avec remise, à chaque tirage la probabilité d'obtenir une blanche ("succès") est $p = \frac{2}{4} = \frac{1}{2}$. La variable aléatoire X correspond au temps d'attente du premier succès dans une suite d'épreuves de Bernoulli indépendantes et de même paramètre $p = 1/2$. Donc X suit une loi géométrique $\mathcal{G}(p)$:

✓ $X(\Omega) = \mathbf{N}^\star$

✓ $\mathbf{P}(X = k) = (1-p)^{k-1}p = \frac{1}{2^k}$, $\forall k \in \mathbf{N}^\star$

Pour c quelconque, $\mathbf{P}(X = 3) = \mathbf{P}(N_1 \cap N_2 \cap B_3) = \mathbf{P}(B_3|N_1 \cap N_2) \times \mathbf{P}(N_2|N_1) \times \mathbf{P}(N_1)$ d'après la formule des probabilités composées.
$\mathbf{P}(N_1) = \frac{2}{4}$, $\mathbf{P}(N_2|N_1) = \frac{2+c}{4+c}$ (on a rajouté c noires après le premier tirage)
$\mathbf{P}(B_3|N_1 \cap N_2) = \frac{2}{4+2c}$ (on a rajouté $2c$ noires au total et l'urne contient 2 blanches).

D'où $\mathbf{P}(X = 3) = \frac{2}{4+2c} \times \frac{2+c}{4+c} \times \frac{2}{4}$ soit $\boxed{\mathbf{P}(X = 3) = \frac{1}{2(4+c)}}$.

2 — **2.1.** Fonction qui simule l'expérience avec un nombre maximal de tirages égal à s et qui doit renvoyer le rang d'apparition d'une boule blanche si une boule blanche a été obtenue et 0 sinon :

```
from random import random

# première version avec boucle while
def rang_blanche(c, s):
    N = 4 # nombre total de boules
    n = 1 # premier tirage
    p = 2/N # probabilité d'obtenir une blanche
    succes = random() < p # succes = obtenir une blanche
    while n < s and not succes:
        N = N + c # on rajoute c noires
        n += 1 # tirage suivant
        p = 2/N
        succes = random() < p
    if succes:
        return n
    return 0

#deuxième version avec boucle for
def rang_blanche1(c, s):
    N = 4 # nombre total de boules
    for k in range(1, s + 1): # s tirages au plus
        p = 2/N # probabilité d'obtenir une blanche
        if random() < p: # on obtient une blanche
```

```
                return k
            N = N + c # sinon on rajoute c noires pour le tirage suivant
        return 0
```

2.2. Estimation de $\mathbf{P}(X=0)$ pour $c=1$, $c=2$ et $c=5$. D'après la loi faible des grands nombres, la fréquence de réalisation d'un événement A est une bonne approximation de sa probabilité $\mathbf{P}(A)$ lorsqu'on réalise un très grand nombre de simulations.

```
def freq_empirique(c, s, nb_simul = 10000):
    compteur = 0
    for i in range(nb_simul):
        if rang_blanche(c, s) == 0:
            compteur += 1
    return compteur/nb_simul

for c in [1, 2, 5]:
    for s in range(100,1000,100):
        print("pour c =",c,"\t",freq_empirique(c,s))
```

On obtient des valeurs proches de 0 qui laisse penser que $\mathbf{P}(X=0)=0$.
Quand c augmente, la fréquence de 0 obtenus augmente (ce qui est logique car ça devient plus compliqué d'obtenir une blanche au fil des tirages), mais elle a tendance à se rapprocher tout de même de 0 lorsqu'on augmente la valeur de s.

3 — On applique à nouveau la formule des probabilités composées :

$$\begin{aligned}\mathbf{P}(E_n) &= \mathbf{P}(N_1 \cap \ldots \cap N_{n-1} \cap N_n)\\ &= \mathbf{P}(N_n|N_1 \cap \ldots \cap N_{n-1}) \times \mathbf{P}(N_{n-1}|N_1 \cap \ldots \cap N_{n-2}) \times \ldots \times \mathbf{P}(N_2|N_1) \times \mathbf{P}(N_1)\end{aligned}$$

Il est clair que $\mathbf{P}(N_1) = \frac{2}{4} = \frac{2+0c}{4+0c}$ et pour $k \in [\![1\,,\, n-1]\!]$, $\mathbf{P}(N_{k+1}|N_1 \cap \ldots \cap N_k) = \frac{2+kc}{4+kc}$ car il y a eu k tirages avant le $(k+1)$-ième qui ont donné une noire donc on a rajouté kc boules noires au total. Conclusion :

$$\boxed{\forall n \in \mathbf{N}^\star,\ \mathbf{P}(E_n) = \prod_{k=0}^{n-1} \frac{2+kc}{4+kc}}.$$

4 — On suppose dans cette question que $c=1$.

4.1. Pour $c=1$: $\mathbf{P}(E_n) = \prod_{k=0}^{n-1} \frac{2+k}{4+k} = \frac{2\times3\times\ldots\times n+1}{4\times5\times\ldots\times n+3} = \frac{(n+1)!\times2\times3}{(n+3)!}$. Soit :

$$\boxed{\mathbf{P}(E_n) = \frac{6}{(n+2)(n+3)},\ \forall n \in \mathbf{N}^\star}.$$

$\{X=0\} = \cap_{n=1}^{+\infty} E_n$ donc $\{X=0\} \subset E_n$ et par conséquent $0 \leq \mathbf{P}(X=0) \leq \mathbf{P}(E_n)$, pour tout $n \in \mathbf{N}^\star$. Comme $\lim_{n\to+\infty} \mathbf{P}(E_n) = 0$, on en déduit par encadrement, que $\boxed{\mathbf{P}(X=0)=0}$.

4.2. $\{X=n\}$ signifie que n tirages ont eu lieu, on a obtenu $n-1$ noires au cours des $n-1$ premiers tirages puis une blanche au n-ième tirage.
$\mathbf{P}(X=n) = \mathbf{P}(E_{n-1} \cap B_n) = \mathbf{P}(B_n|E_{n-1}) \times \mathbf{P}(E_{n-1})$
$\mathbf{P}(B_n|E_{n-1}) = \frac{2}{4+n-1} = \frac{2}{n+3}$ et d'après 3., avec $c=1$, on a $\mathbf{P}(E_{n-1}) = \frac{6}{(n+1)(n+2)}$ donc on a bien

$$\boxed{\mathbf{P}(X=n) = \frac{12}{(n+1)(n+2)(n+3)},\ \forall n \in \mathbf{N}^\star}.$$

4.3. Théorème du transfert : la variable aléatoire $X+3$ admet une espérance si et seulement si la série $\sum_{n\geq0}(n+3)\mathbf{P}(X=n)$ converge absolument.
Il s'agit d'une série à termes positifs ;
$\mathbf{P}(X=0)=0$ et pour $n \geq 1$, $(n+3)\mathbf{P}(X=n) = \frac{12}{(n+1)(n+2)} = 12\left(\frac{1}{n+1} - \frac{1}{n+2}\right)$.
La série étant télescopique, on peut simplifier la somme partielle de rang n :

$$S_n = 3\mathbf{P}(X=0) + \sum_{k=1}^{n}(k+3)\mathbf{P}(X=k) = 0 + 12\sum_{k=1}^{n}\left(\frac{1}{k+1} - \frac{1}{k+2}\right) = 12\left(\frac{1}{2} - \frac{1}{n+2}\right).$$

Comme $\lim_{n\to+\infty} \frac{1}{n+2} = 0$, la suite (S_n) converge et $\lim_{n\to+\infty} S_n = \frac{12}{2} = 6$.
Ainsi la série $\sum_{n\geq1}(n+3)\mathbf{P}(X=n)$ converge donc $\mathbf{E}(X+3)$ existe et $\mathbf{E}(X+3) = \lim_{n\to+\infty} S_n$.
$\boxed{\mathbf{E}(X+3)=6}$ donc $\boxed{\mathbf{E}(X)=3}$ car $\mathbf{E}(X) = \mathbf{E}(X+3-3) = \mathbf{E}(X+3) - 3$.

4.4. Simulations avec Python :

```
def moy_empirique(c, s, nb_simul=100000):
    m = 0
    for k in range(nb_simul):
        m += rang_blanche(c, s)
    return m/nb_simul
```

L'exécution `moy_empirique(1,1000)` renvoie 2.98492, c'est une valeur proche de 3.

5 — On suppose dans cette question que $c = 2$.

5.1. Pour $c = 2$: $\mathbf{P}(E_n) = \prod_{k=0}^{n-1} \frac{2+2k}{4+2k} = \prod_{k=0}^{n-1} \frac{1+k}{2+k} = \frac{1\times2\times\ldots\times n}{2\times3\times\ldots\times(n+1)} = \frac{n!}{(n+1)!}$

$$\boxed{\mathbf{P}(E_n) = \frac{1}{n+1},\ \forall n \in \mathbf{N}^\star}.$$

Comme précédemment, $0 \leq \mathbf{P}(X = 0) \leq \mathbf{P}(E_n)$ et $\lim_{n\to+\infty} \mathbf{P}(E_n) = 0$ donc

$$\boxed{\mathbf{P}(X = 0) = 0}.$$

5.2. Pour $n \geq 1$: $\mathbf{P}(X = n) = \mathbf{P}(E_{n-1} \cap B_n) = \mathbf{P}(B_n|E_{n-1}) \times \mathbf{P}(E_{n-1}) = \frac{2}{4+2(n-1)} \times \frac{1}{n}$

$\mathbf{P}(X = n) = \frac{2}{2n+2} \times \frac{1}{n} = \frac{1}{n(n+1)}$.

$$\boxed{\mathbf{P}(X = n) = \frac{1}{n(n+1)},\ \forall n \in \mathbf{N}^\star}.$$

$n\mathbf{P}(X = n) = \frac{1}{n+1}$ et on sait que la série harmonique $\sum_{k\geq1} \frac{1}{k}$ diverge donc

$$\boxed{\text{la variable X n'admet pas d'espérance}}.$$

6 — Dans cette question, c est un entier naturel non nul quelconque.

6.1. Soit $n \in \mathbf{N}^\star$; $-\ln \mathbf{P}(E_n) = \ln\left(\frac{1}{E_n}\right) = \ln\left(\prod_{k=0}^{n-1} \frac{4+kc}{2+kc}\right) = \sum_{k=0}^{n-1} \ln\left(\frac{4+kc}{2+kc}\right)$ et $\frac{4+kc}{2+kc} = 1 + \frac{2}{2+kc}$ donc

$$\boxed{\forall n \in \mathbf{N}^\star,\ -\ln(\mathbf{P}(E_n)) = \sum_{k=0}^{n-1} \ln\left(1 + \frac{2}{2+kc}\right)}.$$

6.2. $\frac{2}{2+nc} \underset{n\to+\infty}{\sim} \frac{2}{nc}$ et $\lim_{n\to+\infty} \frac{2}{nc} = 0$ donc $\ln\left(1 + \frac{2}{2+nc}\right) \underset{n\to+\infty}{\sim} \frac{2}{2+nc} \underset{n\to+\infty}{\sim} \frac{2}{nc}$.

Posons $u_n = \ln\left(1 + \frac{2}{2+nc}\right)$ et $v_n = \frac{2}{nc} = \frac{2}{c} \times \frac{1}{n}$.

Les familles (u_n) et (v_n) sont deux suites positives et $u_n \underset{n\to+\infty}{\sim} v_n$, donc (indication de l'énoncé) les séries $\sum_{n\geq0} u_n$ et $\sum_{n\geq0} v_n$ sont de même nature.

La série harmonique diverge donc $\sum_{n\geq0} v_n$ diverge et par conséquent $\sum_{n\geq0} u_n$ diverge également.

Comme il s'agit d'une série à termes positifs, on en déduit que $\lim_{n\to+\infty} \sum_{k=0}^{n-1} u_k = +\infty$ autrement dit : $\lim_{n\to+\infty} -\ln(\mathbf{P}(E_n)) = +\infty$ et par suite $\lim_{n\to+\infty} \ln(\mathbf{P}(E_n)) = -\infty$.

On conclut que $\lim_{n\to+\infty} \mathbf{P}(E_n) = \lim_{n\to+\infty} \exp(\ln(\mathbf{P}(E_n))) = \lim_{x\to-\infty} \exp(x) = 0$.

Comme précédemment, on en déduit que

$$\boxed{\mathbf{P}(X = 0)}.$$

■■ *Solution (Exercice 1.16)*

1 — **1.1.**

1.2.

1.3.

```
#On renvoie le résultat sous forme de liste
import random as rd
def tirage(n):
    L = list(range(1,n+1))
    rd.shuffle(L) #une liste de tirages
    #attention : le mélange se fait EN PLACE, la fonction ne crée pas de copie de L
    return L

#programme à la main utilisant choice du module random pour le choix d'un élément dans une liste
def tiragebis(n):
    L = []
    N = list(range(1,n+1))
    while len(L) != n:
```

```
            r = rd.choice(N)
            #choix d'un élément aléatoire dans la liste L
            L.append(r)
            N.remove(r)
        return L

    def MAX(L):
        x = L[0]
        for y in L[1:]:
            if y > x:
                x = y
        return x

    def simuleY(k,n):
        L = tirage(n)
        return int(L[k-1] == MAX(L[:k]))

    def f(k,n):
        N = 10000
        li = [simuleY(k,n) for _ in range(N)]
        return sum(li)/N
```

Par exemple `f(8,10)` renvoie 0.1213, `f(6,10)` renvoie 0.1523 et `f(1,10)` renvoie 1.0.

1.4. La fonction précédente renvoie le nombre de records sur N simulations, elle permet donc d'évaluer la probabilité $\mathbf{P}(Y_{k,n}=1)$. En analysant les différentes exécution précédente, nous pouvons conjecturer de manière non délirante que $\mathbf{P}(Y_{k,n}=1)=\frac{1}{k}$ pour tout entier k.
Démontrons le résultat. Posé de cette manière, le sujet est un peu sec. Introduisons l'évènement $R_{k,j}$ « un record a lieu au tirage k avec la boule numérotée j », de sorte que :

$$\{Y_{k,n}=1\}=\bigcup_{j=1}^{n} R_{k,j}.$$

Le nombre de façons d'avoir un record au temps k avec la boule j est :

$$\begin{cases} 0 & \text{si } k>j,\\ \underbrace{(j-1)(j-2)\times\cdots\times(j-k-1)}_{k-1 \text{ premières sans remise}}\times \underbrace{1}_{j \text{ au temps } k} \times \underbrace{(n-k)!}_{n-k \text{ boules restantes}} & \text{sinon.}\end{cases}$$

Donc

$$\mathbf{P}\left(R_{k,j}\right)=\begin{cases} 0 & \text{si } k>j,\\ \frac{\binom{j-1}{k-1}(k-1)!(n-k)!}{n!} & \text{sinon.}\end{cases}.$$

Enfin, d'après la propriété d'additivité dénombrable d'une probabilité, nous obtenons :

$$\mathbf{P}\left(Y_{k,n}=1\right)=\sum_{j=k}^{n}\mathbf{P}\left(R_{k,j}\right)=\sum_{j=k}^{n}\frac{\binom{j-1}{k-1}(k-1)!(n-k)!}{n!}=\frac{(n-k)!(k-1)!}{n!}\sum_{j=k}^{n}\binom{j-1}{k-1}=\frac{(n-k)!(k-1)!}{n!}\binom{n}{k}.$$

La dernière égalité, découle de l'égalité

$$\binom{p}{p}+\binom{p+1}{p}+\cdots+\binom{p+q}{p}=\binom{p+q+1}{p+1}$$

valable pour tous les entiers p, q, et que l'on peut démontrer par récurrence sur q en utilisant la formule de Pascal.
En simplifiant, on trouve enfin :

$$\boxed{\mathbf{P}\left(Y_{k,n}=1\right)=\frac{1}{k}}.$$

Autrement dit, $Y_{k,n}\hookrightarrow\mathcal{B}(1/k)$.

2 — 2.1. Nous avons $B_n=\sum_{k=1}^{n}Y_{k,n}$. Par linéarité de l'espérance, on déduit :

$$\mathbf{E}\left(B_n\right)=\sum_{k=1}^{n}\mathbf{E}\left(Y_{k,n}\right)=\boxed{\sum_{k=1}^{n}\frac{1}{k}}.$$

2.2. L'ensemble $\{B_n = 1\}$ est l'évènement « l'unique record est celui obtenu au temps 1 ». Si le premier élément est différent de n, alors forcément nous aurons un second record plus tard. Ainsi le cardinal de $\{B_n = 1\}$ est celui de l'ensemble des listes d'éléments distincts de $[\![1\,,\ n]\!]$ dont n est en position, *i.e.* $(n-1)!$. D'où :

$$\mathbf{P}(B_n = 1) = \frac{(n-1)!}{n!} = \boxed{= \frac{1}{n}.}$$

De-même :

$$\mathbf{P}(B_n = n) = \mathbf{P}(Y_{1,n} = 1, \dots, Y_{n,n} = 1) = \boxed{\frac{1}{n!}},$$

puisque le cardinal de $\{B_n = n\}$ est celui du singleton $(1, 2, 3, \dots, n)$ (seul moyen de réaliser un record à chaque tour, *i.e.* $(n-1)!$.

2.3. Simple comparaison série intégrale : la fonction $t \longmapsto \frac{1}{t}$ est continue sur les intervalles $[\![k\,,\ k+1]\!]$ et décroissante. En sommant, nous obtenons :

$$\int_2^{n+1} \frac{1}{t}\,\mathrm{d}t = \ln\left(\frac{n+1}{2}\right) \leqslant \mathbf{E}(B_n) - 1 \leqslant \int_1^n \frac{1}{t}\,\mathrm{d}t = \ln(n)\,.$$

Soit :

$$\ln\left(\frac{n+1}{2}\right) + 1 \leqslant \mathbf{E}(B_n) \leqslant \ln(n) + 1.$$

En divisant par $\ln(n)$, on tire facilement $\boxed{\mathbf{E}(B_n) \underset{n\to\infty}{\sim} \ln n}$, notamment car :

$$\frac{\ln(n+1) - \ln 2}{\ln n} = \left(1 + \frac{\ln(1+1/n)}{\ln n}\right) - \frac{\ln 2}{\ln n} \underset{n\to\infty}{\longrightarrow} 1.$$

■■ *Solution (Exercice 1.17)*

1 — $\boxed{X_1(\Omega) = \{0, 2\}}$ car la bactérie présente au départ peut se diviser en deux bactéries ou mourir. $\boxed{\mathbf{P}(X_1 = 0) = 1/3}$ et $\boxed{\mathbf{P}(X_1 = 2) = 2/3}$

D'où $\mathbf{E}(X_1) = 0\mathbf{P}(X_1 = 0) + 2\mathbf{P}(X_1 = 2)$ soit $\boxed{\mathbf{E}(X_1) = 4/3}$.

2 — **2.1.** Soit $n \geq 1$; si, parmi les X_{n-1} bactéries présentes après la $(n-1)$-ième étapes, on dénombre k bactéries qui se divisent et $X_{n-1} - k$ qui meurent, alors X_n prend la valeur $2k$: X_n ne prend donc que des valeurs paires.

Par récurrence, on vérifie que $\boxed{X_n(\Omega) = \{0, 2, 4, 6, \dots, 2^n\} = \{2i, i \in [\![0, 2^{n-1}]\!]\}}$.

2.2. On renvoie un résultat sous forme de liste.

```
import random as rd
def simu(n):
    L=[1]
    for k in range(n):
        x=L[len(L)-1]
        if rd.random()<1/3:
            L.append(x-1)
        else:
            L.append(x+1)
    return L
```

2.3. Loi conditionnelle de X_{n+1} sachant $\{X_n = 2i\}$:

- 1er cas : $i = 0$

  il est clair que $X_n = 0$ implique $X_{n+1} = 0$ *i.e.* $\mathbf{P}(X_{n+1} = 0 | X_n = 0) = 1$,

- 2e cas : $1 \leq i \leq 2^{n-1}$

  Notons Y_{n+1} le nombre de bactéries présentes après l'étape n qui se divisent en deux à l'étape $n+1$; alors $X_{n+1} = 2Y_{n+1}$.

  On suppose ici que $X_n = 2i$; chacune des $2i$ bactéries présentes après la n-ième étape a une probabilité $p = 2/3$ de se diviser en deux, et ce indépendamment des autres bactéries présentes. Donc Y_{n+1} sachant $(X_n = 2i)$ suit une loi binomiale $\mathcal{B}(2i, p)$.

  D'où $\mathbf{P}(X_{n+1} = 2j | X_n = 2i) = \mathbf{P}(Y_{n+1} = j) = \binom{2i}{j} p^j (1-p)^{2i-j}$, $\quad \forall j \in [\![0, 2i]\!]$.

On peut remarquer que cette dernière formule est encore vraie pour $i = 0$ donc on conclut que pour tout $i \in X_n(\Omega)$:

$$\boxed{\mathbf{P}(X_{n+1}=2j|X_n=2i)=\binom{2i}{j}\left(\frac{2}{3}\right)^j\left(\frac{1}{3}\right)^{2i-j},\quad \forall j\in[\![0,2i]\!]}.$$

3 — 3.1. Nous avons, puisque les ensemble $\{X_n=k\}$ pour $k\in X_n(\Omega)$ forment un système complet d'évènements, $\boxed{G_{X_n}(1)=1.}$

En dérivant la somme finie G_n, et en faisant $x=1$, on trouve :

$$G_n'(1)=\sum_{k\in X_n(\Omega)} k\mathbf{P}(X_n=k)= \quad \boxed{\mathbf{E}(X_n)}.$$

3.2. Par dérivation d'une fonction composée on a d'une part $G_{n+1}'=G_n'\circ G_1G_1'$, puis en faisant $x=1$: $G_{n+1}'(1)=G_n'(1)G_1'(1)$ qui fournit la relation de récurrence suivante : $\mathbf{E}(X_{n+1})=\mathbf{E}(X_n)\,\mathbf{E}(X_1)$ $\boxed{=\frac{4}{3}\mathbf{E}(X_n)}$.

3.3. La suite $(\mathbf{E}(X_n))$ est une suite géométrique de raison $4/3$. On en déduit que pour tout entier naturel n, $\mathbf{E}(X_n)=(4/3)^n\times\mathbf{E}(X_0)$.

4 — *(Étude de la probabilité d'extinction)* On note $u_n=\mathbf{P}(X_n=0)$. D'après l'énoncé, pour tout réel x on a : $G_{n+1}(x)=(G_1\circ G_n)(x)=G_1(G_n(x))$. Or : $\forall t\in\mathbf{R}$, $G_1(t)=\sum_{k\in X_1(\Omega)}t^k\mathbf{P}(X_1=k)=t^0\mathbf{P}(X_1=0)+t^2\mathbf{P}(X_1=2)=1/3+2/3t^2$. D'où on tire $G_1(G_n(x))=1/3+2/3(G_n(x))^2$. Ainsi, on a démontré que :

$$\boxed{\forall n\in\mathbf{N},\ \forall x\in\mathbf{R},\ G_{n+1}(x)=\frac{1}{3}+\frac{2}{3}\left(G_n(x)\right)^2}.$$

En particulier pour $x=0$, on a donc $G_{n+1}(0)=\frac{1}{3}+\frac{2}{3}\left(G_n(0)\right)^2$. Or $G_n(0)=\mathbf{P}(X_n=0)=u_n$ car $0^0=1$ et $0^k=0$ si $k\neq 0$. Ainsi on a bien :

$$\boxed{\forall n\in\mathbf{N},\quad u_{n+1}=\frac{1}{3}+\frac{2}{3}u_n^2}.$$

4.1. Pour tout $n\in\mathbf{N}$, on a $u_{n+1}=f(u_n)$ où f est la fonction définie par : $f(x)=\frac{1}{3}+\frac{2}{3}x^2$.

La fonction f est croissante sur $[0,1]$ et $f([0,1/2])=[f(0),f(1/2)]=[1/3,1/2]$. Comme $u_0=\mathbf{P}(X_0=0)=0\in[0,1/2]$ et $f([0,1/2])\subset[0,1/2]$, il est facile de démontrer par récurrence que pour tout entier naturel n, on a : $u_n\in[0,1/2]$. Ainsi, on a bien :

$$\boxed{\forall n\in\mathbf{N},\quad 0\leq u_n\leq\frac{1}{2}}.$$

Pour établir la convergence de la suite (u_n), on se propose d'étudier sa monotonie autrement dit le signe de $u_{n+1}-u_n=f(u_n)-u_n$. Pour tout réel x, $f(x)-x=\frac{1}{3}\left(2x^2-3x+1\right)=\frac{1}{3}(x-1)(2x-1)$. Sur l'intervalle $[0,1/2]$, $f(x)-x\geq 0$. Comme $u_n\in[0,1/2]$, on conclut que $u_{n+1}-u_n\geq 0$: $\boxed{\text{la suite } (u_n) \text{ est croissante}}$ (ce qui est logique car $X_n=0$ implique $X_{n+1}=0$ donc $\mathbf{P}(X_n=0)\leq\mathbf{P}(X_{n+1}=0)$). La suite (u_n) étant croissante et majorée par $1/2$, elle converge et sa limite ℓ appartient à $[0,1/2]$. Pour tout n de $\mathbf{N}$, on a : $u_{n+1}=f(u_n)$, avec f continue, ainsi en passant à la limite on en déduit que ℓ vérifie $\ell=f(\ell)$. Une résolution simple d'équation montre que $\ell\in$ *ensemble*$1,1/2$. Mais puisque $\ell\in[0;1/2]$, on a forcément

$$\boxed{(u_n)\text{ converge et }\lim_{n\to+\infty}u_n=\frac{1}{2}}.$$

5 — On note $\forall n\in\mathbf{N}$, D_n l'événement « la population disparaît exactement à l'issue de l'étape n ».

5.1. D_n signifie qu'il y a encore au moins une bactérie à l'issue de l'étape $n-1$ et qu'il n'y en a plus à l'issue de l'étape n. Donc $D_n=(X_n=0)\cap(X_{n-1}\neq 0)$. De plus $((X_{n-1}\neq 0),(X_{n-1}=0))$ est un système complet d'événements ; en appliquant la formule des probabilités totales on a donc :

$$\mathbf{P}(X_n=0)=\mathbf{P}((X_n=0)\cap(X_{n-1}\neq 0))+\mathbf{P}((X_n=0)\cap(X_{n-1}=0)).$$

On a $\mathbf{P}(X_n=0)=u_n$ et $\mathbf{P}((X_n=0)\cap(X_{n-1}\neq 0))=\mathbf{P}(D_n)$. Par ailleurs, $\mathbf{P}((X_n=0)\cap(X_{n-1}=0))=\mathbf{P}(X_{n-1}=0)=u_{n-1}$ car $(X_{n-1}=0)$ implique $(X_n=0)$. Finalement, on a bien :

$$\boxed{\forall n\in\mathbf{N}^\star,\mathbf{P}(D_n)=u_n-u_{n-1}}.$$

5.2. Soit R l'événement « la population de bactéries finit par s'éteindre » : $R=\cup_{n=1}^{+\infty}D_n$. Comme les événements D_n

sont deux à deux incompatibles, alors

$$\mathbf{P}(\mathrm{R}) = \sum_{n=1}^{+\infty} \mathbf{P}(\mathrm{D}_n) = \sum_{n=1}^{+\infty} (u_n - u_{n-1}).$$

Il s'agit de la somme d'une série télescopique. Pour tout $n \in \mathbf{N}^\star$, notons S_n la somme partielle de rang n : $S_n = \sum_{k=1}^{n}(u_k - u_{k-1}) = u_n - u_0$. Et $u_0 = \mathbf{P}(X_0 = 0) = 0$ donc $S_n = u_n$. Comme $\lim_{n\to+\infty} u_n = 1/2$, on en déduit que $\lim_{n\to+\infty} S_n = 1/2$. Ainsi

$$\boxed{\mathbf{P}(\mathrm{R}) = 1/2}.$$

La probabilité que la population de bactéries s'éteigne vaut $1/2$.

■■ *Solution (Exercice 1.18)*

1 — On remarque que pour tout n, $X_n(\Omega) = [\![0\,,\,n]\!]$ puisque c'est le nombre de parties gagnées par A en n parties.

✓ Nous avons $X_1(\Omega) = \{0,1\}$ et la probabilité que A gagne la première partie est de $\frac{1}{2}$. Donc $X_1 \hookrightarrow \mathcal{B}(1/2)$ (ou encore $\mathcal{U}([\![0\,,\,1]\!])$).

✓ $X_2(\Omega) = \{0,1,2\}$ De plus, $\mathbf{P}(X_2 = 0) = P(B_1 \cap B_2) = P(B_1) \times \mathbf{P}(B_1|B_2) = \frac{1}{2} \times \frac{2}{3}$ donc $\boxed{\mathbf{P}(X_2 = 0) = \frac{1}{3}}$.

On fait comme avant pour $\mathbf{P}(X_2 = 2)$: $\mathbf{P}(X_2 = 2) = \mathbf{P}(A_1 \cap A_2) = \frac{1}{2} \times \frac{2}{3}$. Donc $\boxed{\mathbf{P}(X_2 = 2) = \frac{1}{3}}$.

d'où $\mathbf{P}(X_2 = 1) = 1 - \frac{1}{3} - \frac{1}{3} \iff \boxed{\mathbf{P}(X_2 = 1) = \frac{1}{3}}$ Conclusions : $\boxed{X_2 \hookrightarrow \mathcal{U}[\![0\,,\,2]\!]}$.

1.1.
```
import random as rd
def simulX(n):
    #a designe le capital de A, b celui de B, et x le nombre de parties gagnées par A
    #gagnees par A
    a,b,x = 1,1,0
    for k in range(n): # on effectue n parties
        u = rd.random()
        if u < a/(a+b) :
            a,b,x = a+1,b,x+1
        else:
            a,b,x=a,b+1,x
    return x
```

2 — **2.1.** Comme d'habitude, le principe est d'utiliser la loi faible des grands nombres pour obtenir une valeur approchée de chaque $\mathbf{P}(X_n = k)$, donc on simule un grand nombre m de fois la variable X_n, on accumule les résultats, puis on fait la moyenne en divisant par m.

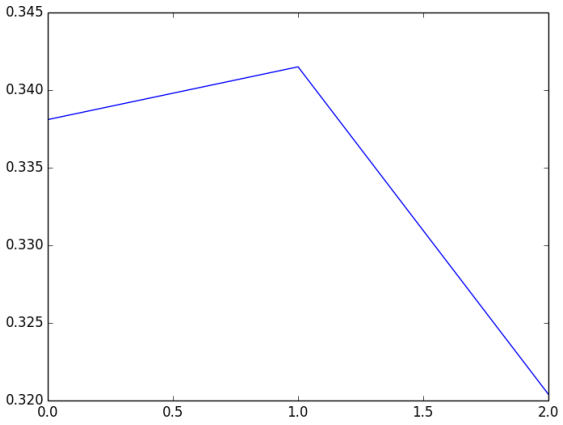
```
def loiX(n):
    m = 10000
    l = (n+1)*[0]
    for _ in range(m):
        l[simulX(n)]+=1
    return [l[i]/m for i in range(n+1)]
```

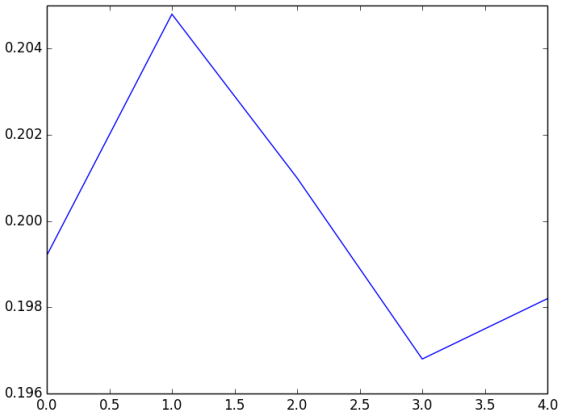
On peut vérifier les résultats sur la loi de X_n pour $n = 1$ ou $n = 2$. Par exemple pour $n = 2$, on obtient : [0.3462, 0.3272, 0.3266]. Dans la liste `loiX`, on trouve bien chaque cellule $\simeq 0,33333...$.

2.2. Pour obtenir une représentation graphique on peut demander `graphe(loiX(2))`.

```
import matplotlib.pyplot as plt
def graphe(loi):
    lx=[i for i in range(len(loi))]
    plt.plot(lx,loi)
```

Commençons par une exécution avec $n = 2$. drait des probabilités égales à $\frac{1}{5} = 0,2$. On obtient :





Autre exemple : pour $n = 4$, on se doute qu'on vou-

2.3. Contrairement à ce que laisse présager les dessins, la variabilité est faible, notre conjecture est la suivante :

$\boxed{X_n \hookrightarrow \mathcal{U}[\![0\,,\,n]\!] \text{ donc avec } \forall k \in [\![0\,,\,n]\!], \quad \mathbf{P}(X_n = k) = \frac{1}{n+1}}$

2.4. Posons $\mathcal{P}(n)$ la propriété : $\forall k \in [\![0\,,\,n]\!], \quad \mathbf{P}(X_n = k) = \frac{1}{n+1}$. On va la montrer par récurrence pour tout $n \in \mathbf{N}$.

■ ***Hérédité.*** En 0 partie, la probabilité que A en gagne 0 est de 1, et $\frac{1}{0+1} = 1$.

De façon plus intéressante, pour $n = 1$, on a bien trouvé que $\mathbf{P}(X_1 = 0) = \mathbf{P}(X_1 = 1) = \frac{1}{2} = \frac{1}{1+1}$.
Donc on a bien $\mathcal{P}(0)$ et $\mathcal{P}(1)$ vraies.

■ ***Initialisation.*** Supposons $\mathcal{P}(n-1)$ vraie pour un $n \geqslant 1$ quelconque fixé, et montrons $\mathcal{P}(n)$.

Soit $k \in [\![0\,,\,n]\!], \quad \mathbf{P}(X_n = k) = \sum_{j=0}^{n-1} \mathbf{P}((X_{n-1} = j) \cap (X_n = k))$ d' après la formule des probabilités totales avec le système complet d'événements induit par X_{n-1}.
Or si $k \geqslant 1$, pour tout $j \neq k-1$ et $j \neq k$, $\quad \mathbf{P}((X_{n-1} = j) \cap (X_n = k)) = 0$.
Donc $\mathbf{P}(X_n = k) = \mathbf{P}((X_{n-1} = k-1) \cap (X_n = k)) + \mathbf{P}((X_{n-1} = k) \cap (X_n = k))$, et de manière équivalente avec des probabilités conditionnelles :

$$\mathbf{P}(X_n = k) = \mathbf{P}(X_n = k | X_{n-1} = k-1)\,\mathbf{P}(X_{n-1} = k-1) + \mathbf{P}(X_n = k | X_{n-1} = k)\,\mathbf{P}(X_{n-1} = k).$$

On peut appliquer l'hypothèse de récurrence aux deux parties si $1 \leqslant k \leqslant n-1$. Nous traiterons les autres cas à part.
Sachant $X_{n-1} = k$, le joueur A a gagné k fois et le joueur B un nombre $n-k-1$ de victoires. Les capitaux des deux joueurs sont donc à cet instant $1+k$ pour A et $1+(n-k-1)$ pour B. Donc :

$$\mathbf{P}(X_n = k | X_{n-1} = k-1) = \frac{1+(k-1)}{2+(n-1)} = \frac{k}{n+1}.$$

De-même, on trouve :

$$\mathbf{P}(X_n = k | X_{n-1} = k) = \frac{n-k}{n+1}.$$

D'après l'hypothèse de récurrence, on trouve toujours pour $1 \leqslant k \leqslant n-1$:

$$\mathbf{P}(X_n = k) = \frac{k}{n+1}\frac{1}{n} + \frac{n-k}{n+1}\frac{1}{n} = \frac{n}{n+1}\frac{1}{n} = \frac{1}{n+1}.$$

Enfin on traite le cas $k = 0$:

$$\mathbf{P}(X_n = 0) = \mathbf{P}(X_n = 0 | X_{n-1} = 0)\,\mathbf{P}(X_{n-1} = 0) = \frac{n}{n+1}\frac{1}{n} = \frac{1}{n+1}.$$

Le cas $k = n$ est similaire (sauf que c'est l'autre joueur qui a gagné à chaque coup).
Conclusion : $\boxed{X_n \text{ suit une loi uniforme sur } [\![0\,,\,n]\!]}$.

3 — Ici, on pose $n = 2p+1, \quad p \in \mathbf{N}$. Remporter le tournoi c'est donc gagner au moins $p+1$. Notons G_A l'événement « A remporte le tournoi après n combats ».

$$\mathbf{P}(G_A) = P(X_{2p+1} \geqslant p+1) = \sum_{k=p+1}^{2p+1} \mathbf{P}(X_{2p+1} = k) = (p+1) \times \frac{1}{2p+2} \quad \Longleftrightarrow \quad \boxed{\mathbf{P}(G_A) = \frac{1}{2}}.$$

Partie I— Deuxième cas : les capitaux initiaux sont égaux à a et 1..

Désormais $\alpha_0 = a$ et $\beta_0 = 1$.

4 — $X_3(\Omega) = [\![0\,,\,3]\!]$.

✓ D'après la formule des probabilités composées : $\mathbf{P}(X_3 = 0) = \frac{1}{a+1} \times \frac{2}{a+2} \times \frac{3}{a+3}$.

✓ D'après la formule des probabilités composées : $\mathbf{P}(X_n = 3) = \frac{a}{a+1} \times \frac{a+1}{a+2} \times \frac{a+2}{a+3} = \frac{a}{a+3}$.

✓ L'ensemble $\{X_3 = 1\}$ signifie que A a gagné une fois en trois coups, donc soit au premier tour, soit au second, soit au troisième (et a perdu aux autres coups). Par additivité dénombrable et probabilités composées, on obtient :

$$\mathbf{P}(X_3 = 1) = \frac{a}{a+1} \times \frac{1}{a+2} \times \frac{2}{a+3} + \frac{1}{a+1} \times \frac{a+1}{a+2} \times \frac{2}{a+3} + \frac{1}{a+1} \times \frac{2}{a+2} \times \frac{a+2}{a+3}.$$

✓ Enfin, on obtient $\mathbf{P}(X_3 = 2)$ en écrivant $\mathbf{P}(X_3 = 2) = 1 - \mathbf{P}(X_3 = 0) - \mathbf{P}(X_3 = 1) - \mathbf{P}(X_3 = 3)$.

4.1. $X_n(\Omega) = [\![0\,,\,n]\!]$. Pour $k \in [\![0\,,\,n]\!]$, l'évènement $\{X_n = k\}$ est « k victoires de A, $n-k$ victoires de B ».
Il faut donc choisir les k victoires de A parmi les n parties (les victoires de B sont alors imposées), le capital de A utilisé dans les probabilités conditionnelles passe de a à $a+k-1$, pour B le capital utilisé passe de 1 à $1+n-k-1 = n-k$.
On a donc, par additivité dénombrable : $\mathbf{P}(X_n = k) = \frac{\binom{n}{k}\, a\ldots(a+k-1)\times 1\ldots(n-k)}{(a+1)\ldots(a+n)}$ (c'est une généralisation du calcul de $\mathbf{P}(X_3 = 1)$ fait plus haut, sauf que $\binom{n}{k}$ termes apparaissent au lieu de $\binom{3}{1}$ comme avant).

C'est aussi :

$$\begin{aligned}\mathbf{P}(X_n = k) &= \frac{n!}{k!\ \ (n-k)!} \times \frac{(a+k-1)!}{(a-1)!} \times (n-k)! \times \frac{a!}{(a+n)!} \\ &= \frac{(a+k-1)!}{k!(a-1)!} \times \frac{n!\ \ a!}{(a+n)!} \\ &= \frac{\binom{a+k-1}{a-1}}{\binom{a+n}{a}}.\end{aligned}$$

La probabilité que le joueur A perde le tournoi est $\mathbf{P}(X_n < \frac{n}{2})$.
Si $n = 2p+1$, c'est $\mathbf{P}(X_n \leqslant p) = \sum_{k=0}^{p} \mathbf{P}(X_n = k) = \frac{1}{\binom{a+n}{a}} \sum_{k=0}^{p} \binom{a-1+k}{a-1} = \frac{\binom{a+p}{a}}{\binom{a+n}{a}}$.
Si $n = 2p$, c'est $\mathbf{P}(X_n \leqslant p-1) = \sum_{k=0}^{p-1} \mathbf{P}(X_n = k) = \frac{1}{\binom{a+n}{a}} \sum_{k=0}^{p-1} \binom{a-1+k}{a-1} = \frac{\binom{a+p-1}{a}}{\binom{a+n}{a}}$.

4.2. L' espérance existe puisque la variable est finie.
$\mathbf{E}(X_n) = \sum_{k=0}^{n} k\mathbf{P}(X_n = k) = \sum_{k=1}^{n} k \frac{\binom{a+k-1}{a-1}}{\binom{a+n}{a}} = \frac{1}{\binom{a+n}{a}} \sum_{k=1}^{n} k \frac{(a+k-1)!}{(a-1)!k!} = \frac{1}{\binom{a+n}{a}} \sum_{k=1}^{n} \frac{(a+k-1)!}{(a-1)!(k-1)!}$.
En posant $j = k-1$, on obtient $\mathbf{E}(X) = \frac{1}{\binom{a+n}{a}} \sum_{j=0}^{n-1} \frac{(a+j)!}{(a-1)!j!} = \frac{a}{\binom{a+n}{a}} \sum_{j=0}^{n-1} \frac{(a+j)!}{a!j!} = \frac{a}{\binom{a+n}{a}} \sum_{j=0}^{n-1} \binom{a+j}{a}$
Or d'après l'indication de l'énoncé, $\sum_{j=0}^{n-1} \binom{a+j}{a} = \binom{a+n-1+1}{a+1} = \binom{a+n}{a+1}$
Donc $\mathbf{E}(X_n) = a\frac{a!\ n!}{(a+n)!} \times \frac{(a+n)!}{(a+1)!\ \ (n-1)!} \iff \boxed{\mathbf{E}(X_n) = \frac{a}{a+1} \times n}$.

■■ Solution (Exercice 1.19)

1 — L'évènement $\{X = 1\}$ est l'évènement « obtenir un un puis un élément de $\{1,2,3\}$ lors du lancé de deux dés équilibrés ».
Par indépendance,

$$\mathbf{P}(X = 1) = \frac{1}{6} \times \frac{3}{6} = \boxed{\frac{1}{12}}.$$

Le même raisonnement s'applique pour $\mathbf{P}(X = 7) = \frac{1}{12}$. De plus, l'évènement « D_i est lancé » pour $i \in [\![2\,,\,6]\!]$ est « obtenir i » lors d'un lancé de dé équilibré à 6 faces. Donc

$$\forall k \in [\![2\,,\,6]\!], \qquad \boxed{\mathbf{P}(X = k) = \frac{1}{6}}.$$

2 — python

```
import random as rd
#on retourne le résultat sous forme d'une liste [X,Y]
```

```
#Nous aurons besoin de savoir simuler une loi géométrique

def Geo(p):
    n=1
    while rd.random()>p:
        n=n+1
    return n

def simuXY():
    a=rd.randint(1,6)
    if a==1:
        b=rd.randint(1,6)
        if 1<=b<=3:
            X=1
            #Simu de Y
            Y=Geo(0)
            return [X,Y]
        else:
            X=7
            #Simu de Y
            Y=Geo(1)
            return [X,Y]
    else:
        X=a
        #Simu de Y
        Y=Geo((a-1)/6)
        return [X,Y]
```

3 — Appliquons la formule des probabilités totales au système complet au $\{X=k\}_{k\in[\![1\,,\,7]\!]}$. Alors

$$\mathbf{P}(B_1)=\sum_{\ell=1}^{7}\mathbf{P}(B_1|X=\ell)\,\mathbf{P}(X=\ell)\,.$$

On obtient alors en comptant le nombre de faces blanches des autres dés :

$$\begin{aligned}\mathbf{P}(B_1)&=\mathbf{P}(B_1|X=1)\,\mathbf{P}(X=1)+\mathbf{P}(B_1|X=7)\,\mathbf{P}(X=7)+\sum_{\ell=2}^{6}\mathbf{P}(B_1|X=\ell)\,\mathbf{P}(X=\ell)\\&=\frac{1}{12}(0+1)+\sum_{\ell=2}^{6}\frac{\ell-1}{6}\frac{1}{6}\\&=\frac{1}{12}+\frac{1}{6}\sum_{\ell=1}^{5}\ell=\frac{1}{12}+\frac{1}{6}\frac{5\times6}{2\times6}.\end{aligned}$$

D'où l'on tire : $\boxed{\frac{1}{2}.}$ Par indépendance des lancers, nous avons :

$$\boxed{\mathbf{P}(B_k)=\frac{1}{2^k}}.$$

4 — Sachant que le dé i est sélectionné, pour $i\in[\![1\,,\,7]\!]$, Y suit une loi $\mathcal{G}((i-1)/6)$.
En revanche, si $i=1$: $\mathbf{P}(Y=j|X=1)=0$ puisqu'il n'y a aucune face blanche sur le dé 1.
On en déduit alors l'espérance conditionnelle :

$$\mathbf{E}(Y|X=i)=\begin{cases}\frac{6}{i-1} & \text{si } i\in[\![2\,,\,7]\!]\\ 0 & \text{si } i=1\end{cases}.$$

■■ *Solution (Exercice 1.20)* On considère deux variables aléatoires indépendantes : U et V suivant, chacune, la loi uniforme sur $[0;1]$.

1 — $U^2(\Omega) \subset [0,1]$ donc $F_{U^2}(x) = 0$ si $x < 0$ et $F_{U^2}(x) = 1$ si $x > 10$. Soit $x \in [0,1]$:

$$\begin{aligned} F_{U^2}(x) &= \mathbf{P}(U^2 \leq x) \\ &= \mathbf{P}(-\sqrt{x} \leq U \leq \sqrt{x}) \\ &= F_U(\sqrt{x}) - F_U(-\sqrt{x}) \\ &= \sqrt{x} - 0 \text{ car } \sqrt{x} \in [0,1] \text{ et } -\sqrt{x} \leq 0) \end{aligned}$$

Bilan : $\boxed{F_{U^2}(x) = \begin{cases} 0 & \text{si } x < 0 \\ \sqrt{x} & \text{si } 0 \leq x \leq 1 \\ 1 & \text{si } x > 1 \end{cases}}$ Vérifions qu'il s'agit bien d'une fonction de répartition de variable aléatoire réelle à densité. $\lim\limits_{\substack{x\to 0 \\ x<0}} F_{U^2}(x) = 0 = F_{U^2}(0)$ et $\lim\limits_{\substack{x\to 0 \\ x>0}} F_{U^2}(x) = 0 = F_{U^2}(0)$: F_{U^2} est continue en 0. $\lim\limits_{\substack{x\to 1 \\ x<1}} F_{U^2}(x) = 1 = F_{U^2}(1)$ et $\lim\limits_{\substack{x\to 1 \\ x>1}} F_{U^2}(x) = 1 = F_{U^2}(1)$: F_{U^2} est continue en 1.

La fonction F_{U^2} est donc continue sur $\mathbf{R}$, de classe $\mathcal{C}^1$ sur $\mathbf{R}\backslash\{0,1\}$: $\boxed{U^2 \text{ admet donc une densité}}$.

La fonction f définie par : $f(x) = \begin{cases} F'_{U^2}(x) & \text{si } x \in \mathbf{R}\backslash\{0,1\} \\ 0 & \text{sinon} \end{cases}$ est une densité de U^2.

$$\boxed{f(x) = \begin{cases} \frac{1}{2\sqrt{x}} & \text{si } 0 < x < 1 \\ 0 & \text{sinon} \end{cases}}$$

C'est également une densité de V^2 puisque V suit la même loi que U.

2 — **2.1.** Comme U^2 et V^2 sont indépendantes (car U et V sont indépendantes) et admettent chacune f comme densité, on en déduit que la variable aléatoire $Z = U^2 + V^2$ admet une densité h définie par le produit de convolution des deux :

$$h(x) = f \star f(x) = \int_{-\infty}^{+\infty} f(x-t)f(t)\,\mathrm{d}t.$$

2.2. python Programme permettant de simuler Z et d'estimer $\mathbf{P}(Z \leq 1)$:

```
import random as rd

def simuleZ():
    return rd.random()**2 + rd.random()**2

def freq_empirique(N=10000):
    c = 0
    for k in range(N):
        if simuleZ() <= 1:
            c += 1
    return c/N # estimation de P(Z <= 1)
```

On en déduit que $\mathbf{P}(Z \leq 1) \approx 0,78$.

3 — **3.1.** Soit $0 < x \leq 1$; $h(x) = \int_{-\infty}^{+\infty} f(x-t)f(t)\,\mathrm{d}t$. Or $f(t)f(x-t) = \begin{cases} \frac{1}{2}\frac{1}{\sqrt{x-t}} \times \frac{1}{2}\frac{1}{\sqrt{t}} & \text{si } 0 < t < 1 \text{ et } 0 < x-t < 1 \\ 0 & \text{sinon} \end{cases}$.

$\begin{cases} 0 < t < 1 \\ 0 < x-t < 1 \end{cases} \iff \begin{cases} 0 < t < 1 \\ x-1 < t < x \end{cases} \iff \max(0, x-1) < t < \min(1, x)$. Ici $x \in]0,1]$, donc $\max(0, x-1) < t < \min(1, x) \iff 0 < t < x$. Ainsi on a bien :

$$\boxed{h(x) = \frac{1}{4}\int_0^x \frac{1}{\sqrt{x-t}}\frac{1}{\sqrt{t}}\,\mathrm{d}t, \quad \forall x \in]0,1]}.$$

3.2. Soit $0 < x \leq 1$. La fonction $t \mapsto \frac{1}{\sqrt{x-t}}\frac{1}{\sqrt{t}}$ est définie, continue sur $]0,x[$, donc l'intégrale précédente est impropre en 0 et en x. Changement de variable : $y = \frac{t}{x}$, la fonction $\varphi : t \mapsto \frac{t}{x}$ étant de classe $\mathcal{C}^1$, strictement croissante sur $]0,x[$ et $\varphi(]0,x[) =]0,1[$; on a alors $\mathrm{d}t = x\,\mathrm{d}y$, $\sqrt{x-t} = \sqrt{x-xy} = \sqrt{x}\sqrt{1-y}$ et $\sqrt{t} = \sqrt{x}\sqrt{y}$ donc $h(x) = \frac{1}{4}\int_0^1 \frac{1}{\sqrt{x}}\frac{1}{\sqrt{1-y}}\frac{1}{\sqrt{x}}\frac{1}{\sqrt{y}}x\,\mathrm{d}y$ soit

$$h(x) = \frac{1}{4}\int_0^1 \frac{1}{\sqrt{1-y}}\frac{1}{\sqrt{y}}\,\mathrm{d}y$$

3.3. On pose ensuite $y = \sin^2(u)$ avec $u \in]0,\pi/2[$ ($u \mapsto \sin^2(u)$ étant strictement croissante et de classe C^1 sur $]0,\pi/2[$ à valeurs dans $]0,1[$) $\mathrm{d}y = 2\cos(u)\sin(u)\,\mathrm{d}u$, $\sqrt{1-y} = \sqrt{1-\sin^2(u)} = \sqrt{\cos^2(u)} = \cos(u)$ car $\cos(u) > 0$ sur $]0,\pi/2[$ et $\sqrt{y} = \sqrt{\sin^2(u)} = \sin(u)$ car $\sin(u) > 0$ sur $]0,\pi/2[$. On a donc
$h(x) = \frac{1}{4}\displaystyle\int_0^{\pi/2} \frac{1}{\cos(u)}\frac{1}{\sin(u)} 2\cos(u)\sin(u)\,\mathrm{d}u = \frac{1}{2}\int_0^{\pi/2} 1\,\mathrm{d}u$ ce qui donne :

$$\boxed{h(x) = \frac{\pi}{4}}.$$

3.4. Comme Z a pour densité h, alors $\mathbf{P}(Z \leq 1) = \displaystyle\int_{-\infty}^{1} h(x)\,\mathrm{d}x$. $Z(\Omega) \subset [0,2]$ donc $h = 0$ sur $]-\infty,0]$; de plus $h(x) = \pi/4$ sur $]0,1]$. On en déduit que $\mathbf{P}(Z \leq 1) = \displaystyle\int_0^1 \pi/4\,\mathrm{d}x$ autrement dit

$$\boxed{\mathbf{P}(Z \leq 1) = \pi/4}$$

Cette estimation est cohérente avec ce que nous avions trouvé précédemment. Comme U et V suivent une loi uniforme sur $[0,1]$, le couple (U,V) représente le choix d'un point M(U, V) au hasard sur le carré $[0,1]^2$; $Z = U^2 + V^2 = OM^2$ (distance au carré entre M et le point $O(0,0)$). L'événement $(Z \leq 1)$ signifie que $OM^2 \leq 1$, autrement dit le point M est situé dans le quart de disque de centre O et de rayon 1. Intuitivement, la probabilité qu'il appartienne à ce quart de disque est l'aire de ce quart de disque (c'est-à-dire $\pi/4$) divisé par l'aire totale du carré. Comme l'aire du carré vaut 1, la probabilité de choisir un point au hasard dans le quart de disque est égale $\pi/4$, ce que confirme le calcul précédent.

4 — On considère une suite de variables aléatoires de Bernoulli $(Y_n)_{n\geq 1}$ mutuellement indépendantes et de même paramètre $p = \frac{\pi}{4}$, et on note $S_n = \frac{Y_1+\ldots+Y_n}{n}$ pour tout entier $n \geq 1$.

4.1. Soit $\varepsilon > 0$; inégalité de Bienaymé-Tchebychev : $\mathbf{P}\left(\left|S_n - \mathbf{E}(S_n)\right| \geq \varepsilon\right) \leq \frac{\mathbf{Var}(S_n)}{\varepsilon^2}$. Pour tout $i \in [\![1;n]\!]$, $Y_i \hookrightarrow \mathbf{B}(p)$ donc $\mathbf{E}(Y_i) = p = \pi/4$ et $\mathbf{Var}(Y_i) = p(1-p) = \pi/4(1-\pi/4)$.

Par suite, $\mathbf{E}(S_n) = \frac{1}{n}\displaystyle\sum_{i=1}^{n} \mathbf{E}(Y_n) = \frac{np}{n} = p$ (linéarité de l'espérance) donc $\mathbf{E}(S_n) = \pi/4$.

$\mathbf{Var}(S_n) = \mathbf{Var}\left(\frac{1}{n}\displaystyle\sum_{i=1}^{n} Y_i\right) = \frac{1}{n^2}\sum_{i=1}^{n} \mathbf{Var}(Y_i)$ (car les Y_i sont indépendantes) et par conséquent $\mathbf{Var}(S_n) = \frac{np(1-p)}{n^2} = \frac{p(1-p)}{n} = \frac{\pi/4(1-\pi/4)}{n}$. D'après l'inégalité de Bienaymé-Tchebychev, on a donc : $\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| \geq \varepsilon\right) \leq \frac{\pi/4(1-\pi/4)}{n\varepsilon^2}$.
La fonction $x \mapsto x(1-x)$ admet un maximum en $x = 1/2$ qui vaut $1/4$. En majorant $\pi/4(1-\pi/4)$ par $1/4$, on obtient :

$$\boxed{\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| \geq \varepsilon\right) \leq \frac{1}{4n\varepsilon^2}}.$$

4.2. Dès lors, il suffit que $\frac{1}{4n\varepsilon^2} \leq 0,05$ pour avoir $\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| \geq \varepsilon\right) \leq 0,05$. Or $\frac{1}{4n\varepsilon^2} \leq 0,05 \iff \frac{1}{4n\varepsilon^2} \leq \frac{5}{100} \iff 4n\varepsilon^2 \geq 20 \iff n \geq \frac{5}{\varepsilon^2}$.
Donc pour $n \geq \frac{5}{\varepsilon^2}$, $\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| \geq \varepsilon\right) \leq 0,05$ soit $\mathbf{P}\left(\left|S_n - \frac{\pi}{4}\right| < \varepsilon\right) \geq 0,95$.
Ainsi, pour $n \geq \frac{5}{\varepsilon^2}$, on a : $\mathbf{P}\left(\frac{\pi}{4} \in [S_n - \varepsilon, S_n + \varepsilon]\right) \geq 0,95$.

4.3. En choisissant $\varepsilon = 10^{-2}$, on obtient un intervalle de longueur 2×10^{-2}. $\frac{5}{\varepsilon^2} = 50000$ donc on conclut que :

$$\boxed{\text{si } n \geq 50000,\ \left[S_n - 10^{-2}, S_n + 10^{-2}\right] \text{ est un intervalle de confiance de } p \text{ à } 95\%}$$

.

```
def simuleY():
    if simuleZ() <= 1:
        return 1
    return 0
def intervalle_confiance(eps):
    n = 1
    S = simuleY() # Y1
    while 1/(4*n*eps**2) > 0.05:
        n += 1
        S += simuleY() # Y1+...+Yn
```

```
    S = S/n
    return [S - eps,S + eps]
```

$[0.77274, 0.79274]$ est un intervalle de confiance de $\pi/4$ de longueur 2×10^{-2}.

5 — *Une première alternative* : Si U suit une loi uniforme sur $[0,1]$, alors $\mathbf{P}(\tan(U) \leq 1) = \pi/4$ (car U prend ses valeurs dans $[0,1]$ donc $\text{Arctan}(\tan(U)) = U$). Il suffit alors de reprendre le programme précédent en remplaçant la fonction `simuleY` par la suivante :

```
import random as rd
def simuleY(): # simule la loi de Bernoulli B(pi/4)
    u = rd.random()
    if np.tan(u) <= 1: # probabilité pi/4
        return 1
    return 0
```

Une deuxième alternative D'après la formule du transfert, si U suit une loi uniforme sur $[0,1]$, alors sous réserve d'existence : $\mathbf{E}(\varphi(U)) = \displaystyle\int_0^1 \varphi(t) f_U(t)\,dt = \int_0^1 \varphi(t)\,dt$. On choisit alors une fonction φ de telle sorte que $\displaystyle\int_0^1 \varphi(t)\,dt = \pi/4$. Par exemple la fonction $\varphi : t \mapsto \frac{1}{1+t^2}$ vérifie : $\displaystyle\int_0^1 \varphi(t)\,dt = [\text{Arctan}(t)]_0^1 = \pi/4$. On a donc $\mathbf{E}\left(\frac{1}{1+U^2}\right) = \frac{\pi}{4}$ si U suit une loi uniforme sur $[0,1]$. On définit alors un n-échantillon $(Y_1, \dots, Y_n)$ de la variable $Y = \frac{1}{1+U^2}$: les variables aléatoires $Y_1, \dots, Y_n$ sont indépendantes et de même loi que Y. On pose $M_n = \frac{Y_1+\dots+Y_n}{n}$ (moyenne empirique), $S_n = \sqrt{\frac{1}{n}\sum_{i=1}^n (Y_i - M_n)^2} = \sqrt{\frac{1}{n}\left(\sum_{i=1}^n Y_i^2\right) - M_n^2}$ (écart-type empirique). Dès lors, en appliquant le théorème de la limite centrée, on sait que pour n suffisamment grand, l'intervalle $\left[M_n - 1.96\frac{S_n}{\sqrt{n}}, M_n + 1.96\frac{S_n}{\sqrt{n}}\right]$ est un intervalle de confiance de la moyenne $\mu = \pi/4$.

```
def Y():
    u = random()
    return 1/(1 + u**2)

def moyenne(X):
    n = len(X)
    S = 0
    for x in X:
        S += x
    return S/n

def ecart_type(X):
    n = len(X)
    S = 0
    for x in X:
        S += x**2
    return sqrt(S/n - moyenne(X)**2)

def intervalle_confiance2():
    n = 30 # pour appliquer le TCL, on prend n>= 30
    echantillon = [Y() for i in range(n)]
    sy = ecart_type(echantillon)
    while 1.96 * sy/sqrt(n) > 0.01:
        n += 1
        echantillon.append(Y())
        sy = ecart_type(echantillon)
    M = moyenne(echantillon)
    return n,[M - 1.96*sy/sqrt(n),M + 1.96*sy/sqrt(n)]
```

`(979, [0.7736673136565477, 0.7936563396780032])` : il a fallu 979 itérations pour obtenir l'intervalle de confiance $[0.7736673136565477, 0.7936563396780032]$ qui est d'amplitude inférieure à 10^{-2}.

■■ *Solution (Exercice 1.21)*

1 — 1.1. Puisque $Y \hookrightarrow \mathcal{U}([0,1[)$, la variable X est presque-sûrement à valeurs dans $\mathbf{R}^+$ puisque $\lambda > 0$. Donc

$$\mathbf{P}(X \leqslant x) = \begin{cases} 0 & \text{si } x < 0, \\ \mathbf{P}(\ln(1-Y) \geqslant -\lambda x) = \mathbf{P}(Y \leqslant 1 - e^{-\lambda x}) = 1 - e^{-\lambda x} & \text{sinon.} \end{cases}$$

La dernière étape est obtenue en se souvenant que la fonction de répartition de $\mathcal{U}([0,1[)$ vaut l'identité sur $[0,1[$. En conclusion

$$\boxed{F_X(x) = \begin{cases} 0 & \text{si } x < 0, \\ 1 - e^{-\lambda x} & \text{sinon.} \end{cases}}$$

1.2. On reconnaît la fonction de répartition de $\mathcal{E}(\lambda)$. Donc $\boxed{X \hookrightarrow \mathcal{E}(\lambda)}$.

1.3.
```
import numpy as np
import random as rd

def Simux(lamba):
    return -np.log(rd.random())/lamba
```
Une simulation donne : 3.67859527763.

2 — 2.1. Nous avons par hypothèse $1 - \mathbf{P}(T \leqslant 0) = 1$ donc $\mathbf{P}(T \leqslant 0) = 0 = F(0)$. Donc $\boxed{F|_{]-\infty;0]} = 0}$, puisque F est une fonction positive croissante.

2.2. Commençons par reformuler l'hypothèse. Elle signifie que, pour $t \in [0;+\infty[$:

$$\frac{\mathbf{P}(\{t < T < t+h\} \cap \{T > t\})}{h\mathbf{P}(T > t)} = \frac{\mathbf{P}(t < T < t+h)}{\mathbf{P}(T > t)} = \frac{F(t+h) - F(t)}{h(1-F(t))} \underset{h \to 0^+}{\longrightarrow} \lambda.$$

Comme $1 - F(t)$ ne dépend pas de h, cette limite s'écrit aussi :

$$\boxed{\frac{F(t+h) - F(t)}{h} \underset{h \to 0^+}{\longrightarrow} (1 - F(t))\lambda.}$$

C'est exactement la définition de la dérivabilité de F en $t \in [0;+\infty[$, ceci étant vrai pour tout t, nous avons : $\boxed{\text{F dérivable sur } [0;+\infty[.}$

2.3. La question précédente donne en plus que F vérifie :

$$\boxed{F'(t) = \lambda(1 - F(t)), \quad \forall t \in [0;+\infty[}.$$

2.4. On résout immédiatement l'équation différentielle : d'une part, d'après la formule du cours, l'ensemble des solutions homogènes est

$$\left\{ t \in [0;+\infty[\longmapsto Ce^{-\lambda t},\ C \in \mathbf{R} \right\}.$$

Comme le second membre est constant, on peut gagner du temps chercher une solution particulière sous forme d'une constante. En conclusion, l'ensemble des solutions est

$$\boxed{\left\{ t \in [0;+\infty[\longmapsto Ce^{-\lambda t} + 1,\ C \in \mathbf{R} \right\}.}$$

Mais comme $F(0) = 0$, il vient comme condition $C + 1 = 0$ donc $C = -1$, et finalement :

$$\boxed{F(t) = \begin{cases} 0 & \text{si } t < 0, \\ 1 - e^{-\lambda t} & \text{sinon.} \end{cases}}$$

2.5. Puisque la fonction de répartition détermine la loi, on déduit que $\boxed{T \hookrightarrow \mathcal{E}(\lambda)}$ d'après la partie précédente, et d'après le cours : $\boxed{\mathbf{E}(T) = \frac{1}{\lambda},\ \mathbf{Var}(T) = \frac{1}{\lambda^2}.}$

■■ *Solution (Exercice 1.23)* On considère la matrice :

$$A = \begin{pmatrix} 3 & 1 & 1 \\ 1 & 0 & 1 \\ -3 & 0 & -1 \end{pmatrix}.$$

1 — **1.1.** On rappelle que deux vecteur u et v sont colinéaires si et seulement si $\text{Rg}(u,v) < 2$.

```
def colineaires(u, v):
    a = np.array([u, v])
    return np.linalg.matrix_rank(a) < 2 #retourne un booléen
```

Nous pouvons alors tester :

```
>>> colineaires([1,1], [2,2])
True
```

1.2. Le vecteur u est vecteur propre de A si et seulement si u est non nul et Au est colinéaire à u. Il suffit alors de tester la condition $u \neq 0$ et la colinéarité entre le produit Au et u :

```
def vecteurs_propres(u):
    return u != [0,0,0] and colineaires(np.dot(A,u), u)
```

2 — **2.1.** On pose $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$;

$$AX = -X \iff (A+I)X = 0 \iff \begin{cases} 4x+y+z &= 0 \\ x+y+z &= 0 \\ -3x &= 0 \end{cases} \iff \begin{cases} x &= 0 \\ z &= -y \end{cases}$$ Le système admet d'autres solutions que $(0,0,0)$ donc -1 est valeur propre de A et

$$\boxed{E_{-1}(A) = \text{Vect}\left(\begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}\right).}$$

$$AX = X \iff \begin{cases} 2x+y+z &= 0 \\ x-y+z &= 0 \\ -3x-2z &= 0 \end{cases} \iff \begin{cases} y+2x+z &= 0 \\ 3x+2z &= 0 \\ -3x-2z &= 0 \end{cases} \iff \begin{cases} y &= -1/2x \\ z &= -3/2x \end{cases}$$

De même, on en déduit que 1 est valeur propre de A et

$$\boxed{E_1(A) = \text{Vect}\left(\begin{pmatrix} -2 \\ 1 \\ 3 \end{pmatrix}\right).}$$

$$AX = 2X \iff \begin{cases} x+y+z &= 0 \\ x-2y+z &= 0 \\ -3x-3z &= 0 \end{cases} \iff \begin{cases} y+x+z &= 0 \\ 3x+3z &= 0 \\ -3x-3z &= 0 \end{cases} \iff \begin{cases} y &= 0 \\ z &= -x \end{cases}$$ Ainsi 2 est bien valeur propre de A et

$$\boxed{E_2(A) = \text{Vect}\left\{\begin{pmatrix} 1 \\ 0 \\ -1 \end{pmatrix}\right\}.}$$

Comme A ne peut avoir plus de trois valeurs propres, on en déduit que $\boxed{\text{Spec}(A) = \{-1,1,2\}}$.

2.2. A est carrée d'ordre 3 et possède 3 valeurs propres distinctes donc $\boxed{\text{A est diagonalisable}}$.

3 — Soient $X_1, \cdots, X_n$, n variables aléatoires indépendantes suivant la loi de Bernoulli de paramètre $p \in]0;1[$. On note : $M_n = \frac{1}{n}\sum_{k=1}^{n} X_k$ et $M_n^\star = \frac{M_n - p}{\sqrt{\frac{p(1-p)}{n}}}$.

3.1. X_k suit la loi de Bernoulli $\mathcal{B}(p)$ donc $\mathbf{E}(X_k) = p$ et $\mathbf{Var}(X_k) = p(1-p)$ pour tout $k \in [\![1\,,\,n]\!]$. On en déduit, par linéarité de l'espérance que $\mathbf{E}(M_n) = \frac{1}{n}\sum_{k=1}^{n}\mathbf{E}(X_k) = \frac{1}{n}\sum_{k=1}^{n} p = p$. De plus, comme les variables aléatoires $X_1, \ldots, X_n$ sont indépendantes, on a :

$\mathbf{Var}(M_n) = \frac{1}{n^2}\sum_{k=1}^{n}\mathbf{Var}(X_k) = \frac{p(1-p)}{n}$ et par conséquent $\sigma_{M_n} = \sqrt{\frac{p(1-p)}{n}}$. Ainsi $M_n^\star = \frac{M_n - \mathbf{E}(M_n)}{\sigma(M_n)}$: $M_n^\star$ correspond à la variable centrée réduite associée à M_n. Comme les les variables aléatoires $X_1, \ldots, X_n$ sont indépendantes et de même loi, on peut appliquer le théorème central limite : on en déduit que

$\boxed{M_n^\star \text{ suit approximativement la loi normale } \mathcal{N}(0,1)}$. Ainsi, pour $\alpha \in \mathbf{R}_+^\star$, $\mathbf{P}([-\alpha < M_n^\star < \alpha]) \approx \Phi(\alpha) - \Phi(-\alpha)$ où Φ désigne la fonction de répartition de la loi normale centrée réduite. Sachant qu'on a $\Phi(-\alpha) = 1 - \Phi(\alpha)$ et $P(-\alpha \leq X \leq \alpha) = P(-\alpha < X < \alpha)$ pour toute variable X à densité, on peut aussi écrire que

$$\boxed{\mathbf{P}([-\alpha \leq M_n^\star \leq \alpha]) \approx 2\Phi(\alpha) - 1}.$$

3.2.

$$\mathbf{P}\left(p \in \left[M_n - \frac{1}{\sqrt{n}}; M_n + \frac{1}{\sqrt{n}}\right]\right) = \mathbf{P}\left(-\frac{1}{\sqrt{n}} \leq M_n - p \leq \frac{1}{\sqrt{n}}\right)$$
$$= \mathbf{P}\left(-\frac{1}{\sqrt{p(1-p)}} \leq M_n^\star \leq \frac{1}{\sqrt{p(1-p)}}\right)$$

Comme $\sqrt{p(1-p)} \leq \sqrt{1/4} = 1/2$ alors $-\frac{1}{\sqrt{p(1-p)}} \leq -2$ et $2 \leq \frac{1}{\sqrt{p(1-p)}}$.[11]

Donc : $\mathbf{P}\left(-\frac{1}{\sqrt{p(1-p)}} \leq M_n^\star \leq \frac{1}{\sqrt{p(1-p)}}\right) \geq \mathrm{P}\left(-2 \leq M_n^\star \leq 2\right)$.

Or $\mathbf{P}\left(-2 \leq M_n^\star \leq 2\right) = 2\Phi(2) - 1 \geq 2\Phi(1,96) - 1 = 0,95$ car la fonction Φ est croissante sur $\mathbf{R}$. Dès lors, on en déduit que $\mathbf{P}\left(-\frac{1}{\sqrt{p(1-p)}} \leq M_n^\star \leq \frac{1}{\sqrt{p(1-p)}}\right) \geq 0,95$.

$\mathbf{P}\left(p \in \left[M_n - \frac{1}{\sqrt{n}}; M_n + \frac{1}{\sqrt{n}}\right]\right) \geq 95\%$ ce qui signifie que

$$\boxed{\left[M_n - \tfrac{1}{\sqrt{n}}, M_n + \tfrac{1}{\sqrt{n}}\right] \text{ est un intervalle de confiance de } p \text{ au seuil de } 95\%}.$$

4 — 4.1. On note N_V (respectivement p) le nombre (respectivement la proportion) de vecteurs propres de A qui appartiennent à $[\![-5\,,\,5]\!]^3$.

Comme $\#\,[\![-5\,,\,5]\!]^3 = 11^3$, alors $p = \frac{N_V}{11^3}$ soit $N_V = p \times 11^3$. On considère l'épreuve de Bernoulli qui consiste à choisir au hasard un vecteur de $[\![-5\,,\,5]\!]^3$, puis à renvoyer 1 si le vecteur en question est un vecteur propre de la matrice A (la probabilité de succès est notre paramètre de Bernoulli).

On réalise $n = 10000$ fois dans des conditions indépendantes cette expérience (ce qui est réalisé dans la boucle `for`).

On note $X_k = 1$ si le k-ième vecteur tiré est vecteur propre de A, $X_k = 0$ sinon.

Les variables aléatoires X_k sont indépendantes et de même loi $\mathcal{B}(p)$.

On sait alors que la variable aléatoire $M_n = \frac{1}{n}\sum_{k=1}^{n} X_k$ est un estimateur sans biais de p.

Le nombre `nb/n` en sortie de boucle correspond à une réalisation de la variable M_n et donne une estimation de p.

En multipliant par 11^3 et en arrondissant à l'entier le plus proche (car $N_V \in \mathbf{N}$)[12], on obtient donc une estimation de N_V.

4.2. On a vu précédemment que $\mathbf{P}\left(M_n - \frac{1}{\sqrt{n}} \leq p \leq M_n + \frac{1}{\sqrt{n}}\right) \geq 0,95$.

Cela équivaut à : $\mathbf{P}\left(11^3 M_n - \frac{11^3}{\sqrt{n}} \leq N_V \leq 11^3 M_n + \frac{11^3}{\sqrt{n}}\right) \geq 0,95$.

Si on choisit n tel que $\frac{11^3}{\sqrt{n}} \leq 0,5$, on aura donc

$$\mathbf{P}\left(11^3 M_n - 0,5 \leq N_V \leq 11^3 M_n + 0,5\right) \geq 0,95.$$

Par ailleurs, l'entier N_n le plus proche de $11^3 M_n$ vérifie

$$11^3 M_n - 0,5 \leq N_n \leq 11^3 M_n + 0,5.$$

On en déduit que l'écart entre N_n et N_V est inférieur ou égal à 1 (avec une probabilité d'au moins 95%).

Or $\frac{11^3}{\sqrt{n}} \leq 0,5 \iff \sqrt{n} \geq 2 * 11^3 \iff n \geq 4 \times 11^6$.

Donc en choisissant $n \geq 4 \times 11^6$ (soit $n \geq 7086244$), la valeur affichée `round(nb/n*11**3)` donne une estimation de N_V à 95%.

4.3. On reprend le programme du début de la question, en remplaçant n par 7086244. On obtient $\boxed{22}$ après exécution.

Calculons la valeur exacte de N_V afin de la comparer à 22 :

D'après l'étude réalisée en seconde question, les vecteurs propres de A à coefficients entiers sont de la forme $(0, k, -k)$ ou $(-2k, k, 3k)$ ou $(k, 0, -k)$ avec $k \in \mathbb{Z}^\star$.

Comme il y a 10 entiers non nuls compris entre -5 et 5, on dénombre :

- ✓ 10 vecteurs propres $(0, k, -k)$ éléments de $[\![-5\,,\,5]\!]^3$
- ✓ 10 vecteurs propres $(k, 0, -k)$ éléments de $[\![-5\,,\,5]\!]^3$

Reste à dénombrer ceux qui sont de la forme $(-2k, k, 3k)$ avec $k \neq 0$.

11. On « remontre » ici, dans ce cas particulier, que l'on peut épaissir tout intervalle de confiance de seuil $1-\alpha$, la version épaissie reste un intervalle de confiance de seuil $1-\alpha$.

12. Attention, cette fonction n'est pas la partie entière, qui est `int()` dans Python

Il faut que l'on ait : $\begin{cases} 0 < |2k| \leq 5 \\ 0 < |k| \leq 5 \\ 0 < |3k| \leq 5 \end{cases}$

Les seuls entiers k qui conviennent sont -1 et 1.

Il y a donc 2 vecteurs propres de la forme $(-2k, k, 3k)$ qui appartiennent à $[\![-5\,,\,5]\!]^3$.

On en déduit que $\boxed{N_V = 10 + 10 + 2 = 22}$ ce qui correspond à la valeur obtenue par estimation dans la question précédente.

CHAPITRE 2

Banque G2E — Mathématiques

1 Algèbre & Géométrie

[Sol 3.1] **Exercice 2.1** Soit $n \geqslant 2$, on note $E = \mathbf{R}_{n-1}[X]$. Soient a et b deux réels distincts.

1 — Soit $F_a = \{P \in E \text{ tel que } P(a) = 0\}$. Montrer que F_a est un sous-espace vectoriel de E. En déterminer une base et sa dimension.

2 — Soit $F = \{P \in E\,, \quad P(a) = P(b) = 0\}$. Montrer que F est un sous-espace vectoriel de E.

3 — On définit $u : P \in E \longmapsto u(P)(X) = P(a)X + P(b)$.

3.1. Montrer que $u \in \mathcal{L}(E)$.

3.2. Déterminer $\mathrm{Ker}(u)$ et $\mathrm{Im}(u)$.

[Sol 3.2] **Exercice 2.2** A tout polynôme P de $\mathbf{R}_{n-1}[X]$, on associe le polynôme F(P) défini par $F(P) = Q$ où

$$Q(X) = P(X) + \frac{1-X}{n}P'(X)$$

1 — Montrer que F est un endomorphisme de $\mathbf{R}_{n-1}[X]$.

2 — Pour k entier entre 1 et n, on pose $P_k = X^{n-k}$ et $Q_k = F(P_k)$. Exprimer Q_k en fonction de P_k et P_{k-1}.

3 — Donner la matrice de F relativement à la base canonique de $\mathbf{R}_{n-1}[X]$

4 — L'endomorphisme F est-il diagonalisable ?

[Sol 2.3] **Exercice 2.3** Soit $E = \left\{(u_n)_{n\in\mathbf{N}},\ \forall n \in \mathbf{N} \quad u_{n+3} = u_{n+2} + \frac{1}{4}u_{n+1} - \frac{1}{4}u_n\right\}$.

1 — Montrer que E est un $\mathbf{R}$-espace vectoriel.

2 — Soit

$$f \left| \begin{array}{ccc} E & \longrightarrow & \mathbf{R}^3 \\ (u_n)_{n\in\mathbf{N}} & \longmapsto & (u_0, u_1, u_2) \end{array} \right. .$$

Montrer que f est un isomorphisme. En déduire la dimension de E.

3 — Donner trois suites géométriques de E.

4 — En déduire E.

[Sol 2.4] **Exercice 2.4** A tout polynôme P de $\mathbf{R}_n[X]$, on associe le polynôme F(P) défini par $F(P) = Q$ où

$$Q(X) = XP(X+1) - (X-1)P(X)$$

1 — Donner degré et coefficient de $F(X^k)$, pour k dans $[\![0\,,\ n]\!]$.

2 — Montrer que F est un endomorphisme de $\mathbf{R}_n[X]$.

3 — Est-il diagonalisable ?

[Sol 2.5] **Exercice 2.5** On définit l'ensemble $\mathcal{C}$ comme l'ensemble des fonctions continues sur $[0;\infty[$. On définit l'application T qui à toute fonction f de $\mathcal{C}$ associe $F = T(f)$, avec :

$$F(x) = \begin{cases} f(0) & \text{si } x = 0 \\ \frac{1}{x}\int_0^x f(t)\,\mathrm{d}t & \text{si } x > 0 \end{cases}$$

1 — Démontrer que T est un endomorphisme de C.

2 — Démontrer que T est injectif.

3 — L'application T est-elle bijective ?

2 Analyse

[Sol 2.6] ■ **Exercice 2.6** Soit $n \in \mathbf{N}^*$, et f_n définie sur $\mathbf{R}^-$ par : $f_n(x) = 1 + x - \frac{e^x}{n}$ pour tout $x \in \mathbf{R}^-$.

1 — Montrer qu'il existe un unique $x_n \in \mathbf{R}^-$ tel que : $f_n(x_n) = 0$.

2 — **2.1.** Montrer que la suite $(x_n)_{n\in\mathbf{N}^*}$ est décroissante et converge vers une limite ℓ.
2.2. Déterminer ℓ.
2.3. Donner un équivalent de $x_n - \ell$ quand $n \to +\infty$.

[Sol 2.7] ■ **Exercice 2.7** oit f la fonction définie par $f(x) = e^x - e^{-x}$.

1 — Étudier f. Montrer que f définit une bijection de $\mathbf{R}$ sur J un intervalle que l'on déterminera. *On note g la bijection réciproque dans la suite.*

2 — Sans la calculer, montrer que g est dérivable et que $g'(x) = \frac{1}{\sqrt{x^2+4}}$ pour tout $x \in$ J.

3 — Déterminer maintenant l'expression de g et retrouver l'expression de $g'(x)$ pour tout x de la question précédente.

3 Probabilités & Statistiques

[Sol 2.8] ■ **Exercice 2.8** Achille et Hector ont chacun une pièce. Achille la lance en premier, puis Hector, puis de nouveau Achille *etc.*, le gagnant est celui qui obtient pile en premier.

1 — On suppose que les deux pièces amènent pile avec la même probabilité $p \in]0,1[$
1.1. Quelle est la probabilité que Achille gagne lors de son n-ième lancer ?
1.2. Quelle est la probabilité que Achille gagne ?
1.3. Le jeu est-il équitable ?

2 — On suppose maintenant que la pièce d'Achille amène pile avec la probabilité $p_1 \in]0,1[$ et celle d'Hector amène pile avec la probabilité $p_2 \in]0,1[$. Pouvez-vous donner une condition sur p_1 et p_2 pour que le jeu soit équitable ?

[Sol 2.9] ■ **Exercice 2.9** On définit une suite de variables aléatoires $(X_k)_{k\in[\![0\,,\,n]\!]}$ avec $n \in \mathbf{N}$. Ces variables sont indépendantes et suivent toutes la même loi de Bernoulli de paramètre p.
On pose, pour tout k de $[\![0\,,\,n-1]\!]$, $Y_k = X_k + X_{k+1}$.

1 — Calculer $\mathbf{Cov}(Y_k, Y_{k+1})$.

2 — Justifier que Y_k et Y_{k+j} sont indépendantes si $j > 1$.

3 — On note $Z = \sum_{k=0}^{n-1} Y_k$. On admet que $\mathbf{Var}(Z) = \sum_{k=0}^{n-1} \mathbf{Var}(Y_k) + \sum_{0\leq i<j\leq n-1} \mathbf{Cov}(Y_i, Y_j)$.
Donner la variance de Z.

[Sol 2.10] ■ **Exercice 2.10** On dispose d'une urne comportant $2n$ boules de couleurs différentes. Parmi elles, n sont numérotées de 1 à n, et les n autres portent le numéro 0. On effectue dans cette urne un tirage simultané de n boules.

1 — Pour i entier entre 0 et n, on note X_i la variable indicatrice de l'événement « la boule numéro i est dans la poignée ». Donner la loi de X_i.

2 — Pour $i \neq j$, donner la loi du couple (X_i, X_j) ainsi que la covariance de X_i et X_j.

[Sol 2.11] ■ **Exercice 2.11** Soit $(X_n)_{n\in\mathbf{N}^*}$ une suite de variables aléatoires indépendantes telles que $\mathbf{P}(X_n = -1) = p$ et $\mathbf{P}(X_n = 1) = 1 - p$ avec $p \in [0,1]$. On pose $Z_n = \prod_{i=1}^{n} X_i$.

1 — Calculer l'espérance de Z_n.

2 — Déduire de la question précédente la loi de Z_n.

3 — Donner une condition sur p pour que Z_1 et Z_2 soient indépendantes.

[Sol 2.12] ■ **Exercice 2.12** On définit la variable X qui correspond au tirage au hasard d'un entier entre 1 et n. Puis si $\{X = k\}$ est réalisé, on tire au hasard un entier entre 1 et k et on note Y le résultat.

1 — **1.1.** Déterminer la loi de X.
1.2. Déterminer la loi de Y.

2 — Calculer l'espérance de Y.

3 — Calculer la covariance de (X, Y).

[Sol 2.13] ■ **Exercice 2.13** Une urne contient n boules numérotées de 0 à $n-1$. On effectue trois tirages successifs avec remise. On note X, Y et Z les résultats obtenus successivement aux trois tirages.

1 — Calculer $\mathbf{P}(X + Y = Z)$.

2 — Calculer $\mathbf{P}(X + Y + Z = n - 1)$.

[Sol 2.14] ■ **Exercice 2.14** soit T une variable aléatoire suivant une loi normale centrée réduite, de fonction de répartition Φ, et a un réel, ainsi que $X = |T| + a$

1 — Donner la fonction de répartition de X en fonction de Φ.

2 — La variable aléatoire X est-elle à densité ? Si oui, en donner une densité .

3 — La variable aléatoire X a-t-elle une espérance ? Si oui, la donner.

[Sol 2.15] ■ **Exercice 2.15** ***Transformée de Laplace d'une variable aléatoire réelle à densité*** Soit X une variable aléatoire de densité f, on pose, pour t réel et lorsque c'est possible,

$$L_X(t) = \mathbf{E}\left(e^{-tX}\right) = \int_{-\infty}^{+\infty} e^{-xt} f(x)\,dx.$$

1 — Déterminer $L_X(0)$.

2 — Soient a et b deux réels tels que $a < b$ et $L_X(a)$ et $L_X(b)$ existent. Montrer que pour tout t compris entre a et b, $L_X(t)$ existe. Que peut-on en déduire sur le domaine de définition de L_X ?

3 — On suppose que X et Y sont deux variables aléatoires indépendantes . Montrer que si t appartient à l'intersection des domaines de définition de L_X et L_Y, alors $L_{X+Y}(t)$ existe et satisfait : $L_{X+Y}(t) = L_X(t)L_Y(t)$.

4 — Montrer que si X suit une loi normale centrée réduite, alors : $\forall t \in \mathbf{R}, \quad L_X(t) = e^{\frac{t^2}{2}}$.

4 Solutions

■■ *Solution (Exercice 2.6)*

1 — f_n est dérivable et $f_n'(x) = 1 - \frac{e^x}{n} \geqslant 0 \iff e^x \leqslant n \iff x \leqslant \ln n$, donc toujours vrai sur $\mathbf{R}^-$.
Donc f_n strictement croissante sur $\mathbf{R}^-$, $\lim_{x\to-\infty} f_n(x) = -\infty$, $f_n(0) = 1 - \frac{1}{n}$ et f_n continue sur $\mathbf{R}^-$.
Alors par le théorème des valeurs intermédiaires, il existe un unique $x_n \in \mathbf{R}^-$ tel que $f_n(x_n) = 0$.

2 — $f_{n+1}(x_n) = 1 + x_n - \frac{e^{x_n}}{n+1} = \frac{e^{x_n}}{n} - \frac{e^{x_n}}{n+1}$ car $f_n(x_n) = 0$.
Donc $f_{n+1}(x_n) = 1 + x_n - \frac{e^{x_n}}{n+1} = e^{x_n}\left(\frac{1}{n} - \frac{1}{n+1}\right) > 0 = f_{n+1}(x_{n+1})$. Or f_n strictement croissante sur $\mathbf{R}^-$ pour tout $n \in \mathbf{N}^*$ donc $\boxed{x_n > x_{n+1}, \text{ la suite } (x_n) \text{ est décroissante.}}$
De plus (x_n) est minorée par -1 (à vérifier en calculs $f_n(-1)$) donc $\boxed{(x_n) \text{ converge vers } \ell \in [-1, 0[}$.

2.1. On a donc l'encadrement : $e^{-1} \leqslant e^{x_n} \leqslant 1 \iff \frac{e^{-1}}{n} \leqslant \frac{e^{x_n}}{n} \leqslant \frac{1}{n}$, et par théorème d'encadrement : $\frac{e^{x_n}}{n} \to 0$ quand $n \to +\infty$.
Alors par passage à la limite dans $f_n(x_n) = 0$ on obtient $1 + \ell - 0 = 0 \iff \boxed{\ell = -1}$

2.2. $x_n - \ell = x_n + 1 = \frac{e^{x_n}}{n} \sim \boxed{\frac{e^{-1}}{n}}$.

Solution (Exercice 2.11)

1 — Par mutuelle indépendance des X_i, on a $\mathbf{E}\left(\prod_{i=1}^n X_i\right) = \prod_{i=1}^n \mathbf{E}(X_i) \iff \mathbf{E}(Z_n) = (\mathbf{E}(X_1))^n$. Or $\mathbf{E}(X_1) = \mathbf{P}(X_1 = 1) - \mathbf{P}(X_1 = -1) = 1 - p - p = 1 - 2p$, d'où $\boxed{\mathbf{E}(Z_n) = (1-2p)^n.}$

2 — On a deux égalités : $\mathbf{P}(Z_n = 1) - \mathbf{P}(Z_n = -1) = \mathbf{E}(Z_n)$ et $\mathbf{P}(Z_n = 1) + \mathbf{P}(Z_n = -1) = 1$, d'où le système :
$$\begin{cases} \mathbf{P}(Z_n = 1) - \mathbf{P}(Z_n = -1) = (1-2p)^n \\ \mathbf{P}(Z_n = 1) + \mathbf{P}(Z_n = -1) = 1 \end{cases} \Leftrightarrow \begin{cases} 2\mathbf{P}(Z_n = 1) = (1-2p)^n + 1 \\ 2\mathbf{P}(Z_n = -1) = 1 - (1-2p)^n \end{cases} \Leftrightarrow \begin{cases} \mathbf{P}(Z_n = 1) = \frac{1+(1-2p)^n}{2} \\ \mathbf{P}(Z_n = -1) = \frac{1-(1-2p)^n}{2} \end{cases}.$$

3 — Nous avons nécessairement : $\mathbf{P}(Z_1 = 1 \cap Z_2 = 1) = \mathbf{P}(Z_1 = 1)\mathbf{P}(Z_2 = 1)$ $(\bigstar)$.
Or $\mathbf{P}(Z_1 = 1 \cap Z_2 = 1) = \mathbf{P}(X_1 = 1 \cap X_1X_2 = 1) = \mathbf{P}(X_1 = 1 \cap X_2 = 1) = \mathbf{P}(X_1 = 1)\mathbf{P}(X_2 = 1) = (1-p)^2$
et $\mathbf{P}(Z_1 = 1) = \mathbf{P}(X_1 = 1) = 1 - p$.

$\mathbf{P}(Z_2 = 1) = \mathbf{P}(X_1 = 1 \cap X_2 = 1) + \mathbf{P}(X_1 = -1 \cap X_2 = -1) = (1-p)^2 - p^2$.
D'où : $(\bigstar) \iff (1-p)((1-p)^2 - p^2) = (1-p)^2 \iff (1-p)^2 - p^2 = 1 - p \iff p(2p-1) = 0 \iff$
$\boxed{p = 0 \text{ ou } p = \frac{1}{2}}$
Il reste à montrer réciproquement que ces deux valeurs fonctionnent.

Solution (Exercice 2.13)

1 — On applique la formule des probabilités totales au système complet $\{X = k, X = \ell\}_{(k,\ell)\in[\![0\,,\,n]\!]^2}$. Nous avons alors

$$\begin{aligned} \mathbf{P}(X + Y = Z) &= \sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1} \mathbf{P}(Z = k + \ell, X = k, Y = \ell) \\ &= \sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1-\ell} \mathbf{P}(Z = k+\ell)\mathbf{P}(X = k)\mathbf{P}(Y = \ell) = \sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1-\ell}\left(\frac{1}{n}\right)^3 = \left(\frac{1}{n}\right)^3 \sum_{\ell=0}^{n-1}(n-\ell) \\ &= \left(\frac{1}{n}\right)^3 \sum_{\ell=1}^{n} \ell = \left(\frac{1}{n}\right)^3 \frac{n(n+1)}{2} = \boxed{\frac{n+1}{2n^2}} \end{aligned}$$

car dans les termes précédents, on ne garde que les couples (k, ℓ) vérifiant $0 \leqslant k + \ell \leqslant n - 1$ *i.e.* $k \leqslant n - 1 - \ell$, et par indépendance de X, Y, Z.

2 — Toujours grâce à la formule des probabilités totales appliquée au même système complet, nous avons

$$\begin{aligned} \mathbf{P}(X + Y + Z = n - 1) &= \sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1} \mathbf{P}(Z = n - 1 - k - \ell, X = k, Y = \ell) \\ &= \sum_{\ell=0}^{n-1}\sum_{k=0}^{n-1-\ell}\left(\frac{1}{n}\right)^3 = \left(\frac{1}{n}\right)^3 \sum_{\ell=0}^{n-\ell}(n-\ell) = \boxed{\frac{n+1}{2n^2}}. \end{aligned}$$

Solution (Exercice 2.14)

1 — Soit $t \in \mathbf{R}$. Alors

$$\mathbf{P}(X \leqslant t) = \mathbf{P}(|T| \leqslant t - a) = \begin{cases} 0 & \text{si } t - a \leqslant 0, \\ \mathbf{P}(-(t-a) \leqslant T \leqslant t - a) = \Phi(t-a) - \Phi(-(t-a)) = 2\Phi(t-a) - 1 & \text{si } t - a > 0. \end{cases}$$

2 — La fonction Φ est de classe C^∞, en particulier, par composition, la fonction F_X (répartition de X) est continue et de classe C^1 sauf en un nombre fini de points. Donc X est à densité, et une densité est donnée par [1] :

$$\boxed{f_X(t) = \frac{2}{\sqrt{2\pi}} e^{-\frac{(t-a)^2}{2}} \mathbb{1}_{[a,\infty[}(t)}.$$

1. Attention, on ne peut pas utiliser le théorème du cours fournissant la densité d'une expression affine, à cause de la valeur absolue.

3 — On en déduit alors aisément l'espérance en étudiant (seconde forme provenant du changement de variable C^1 bijectif $s \mapsto s - a$) :

$$\int_a^{\infty} s\mathrm{e}^{-\frac{(s-a)^2}{2}}\,\mathrm{d}s = \int_0^{\infty} (t+a)\mathrm{e}^{-\frac{t^2}{2}}\,\mathrm{d}t.$$

Nous avons par ailleurs, pour $A > 0$:

$$\int_a^{A} s\mathrm{e}^{-\frac{(s-a)^2}{2}}\,\mathrm{d}s = \int_0^{A} t\mathrm{e}^{-\frac{t^2}{2}}\,\mathrm{d}t + a\int_0^{A} \mathrm{e}^{-\frac{t^2}{2}}\,\mathrm{d}t = \left[\mathrm{e}^{-\frac{t^2}{2}}\right]_0^{A} + a\int_0^{A} \mathrm{e}^{-\frac{t^2}{2}}\,\mathrm{d}t \underset{A\to\infty}{\longrightarrow} \boxed{1 + \frac{a\sqrt{2\pi}}{2}}.$$

La fin du calcul conclut à l'existence de l'espérance.

CHAPITRE 3

Banque G2E — Informatique

1 Énoncés

[Sol 3.1]

Exercice 3.1 Nous nous intéressons à une gamme de DVD. Pour les reconnaître, ils possèdent tous un code composé des trois éléments suivants :

1. Un numéro compris entre 1 et 9999.
2. Une chaîne de caractère constitué de trois lettres.
3. Un numéro final : 1 si le dvd vient d'Amérique 2 s'il vient d'Europe ou d'Afrique et 3 pour l'Asie.

Par exemple, le premier DVD d'Amérique sera : `1 AAA 1`. Le second sera `2 AAA 1` jusqu'à `9999 AAA 1`. Le suivant sera `1 AAB 1` puis `2 AAB 1` jusqu'à `9999 AAZ 1` Puis ensuite 1 ABA 1 et ainsi de suite. Les lettres sont augmentées lorsque l'on atteint 9999.

1 — Par quel objet Python proposez-vous de représenter le code d'un DVD ?
On suppose dans la suite que l'ensemble des DVD d'une vidéothèque est représenté par une liste `listeDVD` contenant tous les codes des DVD de la vidéothèque.
Dans une chaîne de caractère, la fonction `ord(lettre)` permet de transformer une chaîne de caractère formée d'une seule lettre en un entier : `ord('A')` renvoie `65`, `ord('B')` `66` et ainsi de suite jusqu'à `ord('Z')` qui renvoie `90`.

2 — Écrire une fonction qui reçoit en argument le nom sous forme de chaîne de caractères de la région et qui renvoie le code du premier DVD de la région (les noms étant `amerique`, `europe_afrique`, `asie`).

3 — **3.1.** Écrire une fonction qui reçoit en argument la liste des codes de tous les DVD et le code d'une région et qui renvoie la liste des codes des DVD de cette région.
3.2. Écrire une fonction qui reçoit en argument deux chaînes de caractères de longueur trois formées avec les lettres de `A` à `Z` et qui renvoie `True` si la première chaîne est plus grande ou égale à la seconde, `False` sinon (plus grand signifiant au-deLà dans l'ordre alphabétique : par exemple la chaîne `ABA` est plus grande que la chaîne `AAB`).
3.3. Écrire une fonction qui reçoit en argument la liste des codes de tous les DVD et le code d'une région, et qui renvoie le code du DVD le plus ancien de cette région (plus ancien : celui de code le plus petit).

[Sol 3.2]

Exercice 3.2 Nous utilisons le système décimal (base 10) dans nos activités quotidiennes. Ce système est basé sur une logique à dix symboles, de 0 à 9, avec une unité supérieure (dizaine, centaine, etc.) à chaque fois que dix unités sont comptabilisées. C'est un système positionnel, c'est-à-dire que l'endroit où se trouve le symbole définit sa valeur. Ainsi, le 2 de 523 n'a pas la même valeur que le 2 de 132. En fait 523 est l'abréviation de $5 \times 100 + 2 \times 10 + 3$. On peut selon ce principe imaginer une infinité de systèmes numériques fondés sur des bases différentes. En informatique, outre la base 10, on utilise très fréquemment le système binaire (base 2) puisque la logique booléenne est à la base de l'électronique numérique. Deux symboles suffisent : 0 et 1. Cette unité élémentaire ne pouvant prendre que les valeurs 0 et 1 s'appelle un bit (de l'anglais binary digit). Une suite de huit bits s'appelle un octet. On utilise aussi très souvent le système hexadécimal (base 16) du fait de sa simplicité d'utilisation et de représentation pour les mots machines (il est bien plus simple d'utilisation que le binaire). Il faut alors six symboles supplémentaires : A, B, C, D, E et F. Le tableau ci-dessous montre la représentation des nombres de 0 à 15 dans les bases 10, 2 et 16 :

Décimal	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Binaire	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111
Hexadécimal	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F

1 — Écrire une fonction `Exa` convertissant une base décimale en base hexadécimale. Par exemple, la fonction renverra `B` si on execute `hexa(11)`.

2 — Écrire une fonction convertissant une base binaire en base décimale du type 0111 rend 7.

3 — À partir d'une liste de bases binaires, donner une fonction qui la convertit en bases hexadécimales par exemple 1100001001011000 donne C258.

4 — ***(Compression)*** Donner une fonction qui compresse une liste de 1 et de 0 comme ceci 11101100 donne (3,1),(1,0),(2,1),(2,0).

2 Solutions

Solution (Exercice 3.1)

1 — On représente le code d'un DVD par une liste de trois éléments : un entier de 1 à 9999, puis une chaîne de caractère de longueur 3 parmi les lettres A à Z et enfin un entier de 1 à 3.

2 —
```
def question2(nom):
    if nom=='amerique':
        return [1,'AAA',1]
    elif nom=='europe':
        return [1,'AAA',2]
    elif nom=='asie':
        return [1,'AAA',3]
```

3 — **3.1.**
```
import random as rd
def question3.1(listeDVD,code_region):
    liste_region=[]
    for DVD in listeDVD:
        if DVD[2]==code_region:
            liste_region.append(DVD)
    return liste_region
```

3.2.
```
def question3.2(chaine1,chaine2):
    #cette fonction renvoie True si chaine1>=chaine2
    for i in range(3):
        if ord(chaine1[i])>ord(chaine2[i]):
            return True
        elif ord(chaine1[i])<ord(chaine2[i]):
            return False
    #si on n'est pas sorti de la fonction à cette étape, les deux chaines sont égales. La fonction doit
    return True
```

3.3.
```
def question33(liste_DVD,code_region):
    #on récupère tous les DVD de la région demandée
    liste_region=question2a(liste_DVD,code_region)
    #on récupère ensuite le ou les DVD dont le code lettre est le plus petit
    liste_code_lettre_min=[]
    min_lettre='ZZZ'
    for DVD in liste_region:
        if question2b(min_lettre,DVD[1]):
            if min_lettre==DVD[1]:
                #dans ce cas on ajoute le DVD à la liste
                liste_code_lettre_min.append(DVD)
            else:
                #dans ce cas le DVD est plus ancien que tous les DVD précédents. On réinitialise la list
                liste_code_lettre_min=[DVD]
                min_lettre=DVD[1]
    #parmi tous les DVD ayant le code lettre minimum, il faut choisir celui ayant le numéro le plus peti
    plus_ancien=liste_code_lettre_min[0]
    for DVD in liste_code_lettre_min:
        if DVD[0]<plus_ancien[0]:
            plus_ancien=DVD
    return plus_ancien
```