



CHAPITRE X

DEUX APPLICATIONS À LA RÉDUCTION DES MATRICES

TABLE DES MATIÈRES

1. Récurrence linéaire.	2
1.1. Suites récurrentes linéaires.	2
1.2. Systèmes récurrents linéaires.	3
1.3. Retour aux suites récurrentes, explication du théorème 1.1	5
2. Chaînes de Markov.	7
2.1. Rappels préliminaires sur les graphes.	7
2.1.1. Définitions	7
2.2. Représentation d'un graphe.	8
2.2.1. Représentation via la matrice d'adjacence	8
2.2.2. Représentation via les listes d'adjacence	9
2.3. Propriétés des graphes	11
2.4. Graphes pondérés	12
2.5. Un peu de théorie.	13
2.5.1. Chaîne de Markov, matrice de transition et graphe associé.	13
2.5.2. La relation de Chapman-Kolmogorov.	15
2.5.3. État stationnaire.	15
2.6. Deux exemples de sujets de concours.	16
2.6.1. ECRICOME 1992 Exercice 2	16
2.6.2. EDHEC 2017 Problème	17
2.6.3. Autres idées de sujets de concours avec des chaînes de Markov.	19
2.7. Python et chaînes de Markov (à faire en TP)	19
2.7.1. Simuler une chaîne de Markov à l'aide de sa matrice de transition	19
2.7.2. Évolution d'une maladie.	20
2.7.3. Algorithme du Page Rank de Google.	21
2.8. Un exemple avec 3 sites	21
2.9. Le cas général	22
3. Sujets d'annales en lien avec ce chapitre.	23

Ce texte fait suite au chapitre sur la réduction des matrices. Il vise à présenter des applications de la réduction matricielle : l'étude des suites définies par une relation de récurrence linéaire, l'étude des chaînes de Markov et l'étude des systèmes différentiels. Ces trois applications sont indépendantes et appartiennent à des branches différentes du programme (analyse et probabilités). Leur point commun provient des méthodes que nous allons utiliser qui consistent à traduire faire jouer un rôle important à une matrice et à utiliser nos arguments de diagonalisation/réduction.

Ces trois objets (suites récurrentes, chaînes de Markov et systèmes différentiels) apparaissent très fréquemment dans les sujets de concours.

1. RÉCURRENCE LINÉAIRE.

1.1. Suites récurrentes linéaires.

Définition : Suites récurrentes linéaires.

On appelle suite récurrente linéaire d'ordre k une suite définie par la donnée de ses k premiers termes u_0, u_1, \dots, u_{k-1} et par la relation de récurrence linéaire d'ordre k : pour tout $n \in \mathbb{N}$,

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \dots + a_{k-1} u_{n+k-1},$$

où les réels a_0, a_1, \dots, a_{k-1} sont fixés.

Exemple 1.1.1. Au début de ce cours, nous nous sommes intéressés aux suites récurrentes linéaires d'ordre 2. C'est un cas particulier de la définition générale précédente. Ces suites sont définies par la donnée des **deux** premiers termes et par une relation de récurrence de la forme

$$u_{n+2} = a u_{n+1} + b u_n,$$

avec a et b des réels fixés.

Méthode : Suites récurrentes d'ordre 2.

On rappelle ici la méthode qui sert à trouver le terme général d'une suite récurrente linéaire d'ordre 2. Soit donc une suite $(u_n)_{n \in \mathbb{N}}$ pour laquelle on connaît les deux premiers termes u_0 et u_1 et qui vérifie la relation de récurrence

$$u_{n+2} = a u_{n+1} + b u_n,$$

avec a et b des réels fixés.

On introduit alors l'**équation caractéristique** de la suite. C'est l'équation, d'inconnue $x \in \mathbb{R}$ donnée par

$$x^2 - a x - b = 0$$

(attention aux signes). Trois cas se présentent alors

1. Si l'équation n'a pas de solution, on ne peut pas donner le terme général de la suite (dans le programme de ECE2). Ce cas là ne se présente donc **jamais** dans les questions de concours.
2. Si l'équation a une solution x_0 , alors le terme général de la suite est de la forme

$$u_n = (\lambda + \mu n) x_0^n$$

et on trouve λ et μ grâce aux valeurs initiales de la suite u_0 et u_1 .

3. Si l'équation a deux solutions x_1 et x_2 , alors, le terme général de la suite est de la forme

$$u_n = \lambda x_1^n + \mu x_2^n,$$

et on trouve λ et μ grâce aux valeurs initiales de la suite u_0 et u_1 .

Exemple 1.1.2. Trouver (encore une fois) le terme général de la suite de Fibonacci définie par $u_0 = 0$, $u_1 = 1$ et

$$u_{n+2} = u_{n+1} + u_n.$$

Revenons au cas général d'une suite définie par une relation de récurrence d'ordre $k \geq 2$ quelconque.

Théorème : Terme général d'une suite récurrente linéaire (hors programme).

Soit $(u_n)_{n \in \mathbb{N}}$ une suite définie par ses k premiers termes et par la relation de récurrence d'ordre k ,

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \cdots + a_{k-1} u_{n+k-1},$$

où les réels a_0, a_1, \dots, a_{k-1} sont fixés.

Soit (E) l'équation caractéristique de la suite, c'est-à-dire l'équation d'inconnue x donnée par

$$(E) \quad x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_0 = 0.$$

Supposons que les solutions de cette équation soient les nombres réels x_1, x_2, \dots, x_ℓ de multiplicités respectives $\alpha_1, \alpha_2, \dots, \alpha_\ell$. En d'autres termes, on a trouvé une factorisation du polynôme $x \mapsto x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_0$ de la forme

$$x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_0 = (x - x_1)^{\alpha_1} \cdot (x - x_2)^{\alpha_2} \cdots (x - x_\ell)^{\alpha_\ell}$$

Alors le terme général de $(u_n)_{n \in \mathbb{N}}$ est de la forme

$$u_n = P_1(n)x_1^n + P_2(n)x_2^n + \cdots + P_\ell(n)x_\ell^n,$$

où chaque P_i est un polynôme de degré $\alpha_i - 1$. On trouve les coefficients de P_i avec les termes initiaux de la suite.

Remarque 1.1.3. Dans le paragraphe suivant, nous allons généraliser ce théorème. Une fois interprété comme un cas particulier d'une relation de récurrence plus générale, nous pourrions en donner une (idée de) preuve. Nous aurons aussi une méthode plus simple pour donner le terme général de la suite. En effet, il n'est pas facile en général de trouver les racines d'un polynôme de degré k , dès que $k \geq 3$ (c'est même impossible si $k \geq 5$).

Remarque 1.1.4. Ce théorème n'est pas au programme, il sert simplement de guide pour trouver le terme général d'une suite récurrente. Dans les sujets de concours, il faudra justifier dans chaque cas, l'utilisation de ce théorème (et le sujet vous y aidera, voir l'exercice ci-dessous.)

Exercice type concours.

ERICOME 1989 Exercice 1.

On considère l'ensemble E des suites de nombres réels $(u_n)_{n \in \mathbb{N}}$ vérifiant

$$u_{n+3} = 4u_{n+2} - 4u_{n+1} + u_n.$$

1. Montrer que E est un sous-espace vectoriel de l'espace des suites réelles.
2. Vérifier l'existence et préciser la valeur de trois réels distincts r non nuls tels que la suite de terme général $u_n = r^n$ soit élément de E . Ces trois réels seront notés r_1, r_2, r_3 .
3. Soit u un élément de E . Montrer qu'il existe α, β, γ réels tels que la relation

$$u_n = \alpha r_1^n + \beta r_2^n + \gamma r_3^n$$

soit vérifiée pour $n = 0, n = 1$ et $n = 2$.

4. Montrer alors que pour tout $n \in \mathbb{N}$, $u_n = \alpha r_1^n + \beta r_2^n + \gamma r_3^n$.

1.2. Systèmes récurrents linéaires. On s'intéresse maintenant à une nouvelle façon de construire des suites récurrentes qui est en fait plus générale que la précédente. Il s'agit d'une relation entre des vecteurs $X \in \mathcal{M}_{k,1}(\mathbb{R})$.

Définition : Système récurrent.

Soit A une matrice de $\mathcal{M}_k(\mathbb{R})$. On dit qu'une suite $(X_n)_{n \in \mathbb{N}}$ où $X_n \in \mathcal{M}_{k,1}(\mathbb{R})$ est une suite récurrente si elle est définie par la donnée de son premier terme X_0 et par la relation, pour tout $n \in \mathbb{N}$,

$$X_{n+1} = AX_n$$

En raisonnant par récurrence, on peut immédiatement trouver le terme général d'un système récurrent.

Proposition : Terme général d'un système récurrent.

Soit (X_n) un système récurrent. Alors, pour tout $n \in \mathbb{N}$, on a

$$X_n = A^{n+1}X_0.$$

Ainsi donc, pour pouvoir exprimer le terme général d'un système récurrent linéaire, il suffit de savoir calculer les puissances de la matrice de récurrence A . On rappelle alors la méthode qui sert à calculer des puissances de matrices lorsqu'on l'a diagonalisée.

Méthode : Puissances d'une matrice diagonalisable

Soit A une matrice de $\mathcal{M}_k(\mathbb{R})$. On suppose que A est diagonalisable. Il existe donc une matrice P inversible telle que $P^{-1}AP = D$ soit une matrice diagonale. On rappelle que si

$$D = \begin{pmatrix} d_1 & & \\ & \ddots & \\ & & d_k \end{pmatrix}$$

alors

$$D^n = \begin{pmatrix} d_1^n & & \\ & \ddots & \\ & & d_k^n \end{pmatrix}.$$

On sait donc calculer les puissances d'une matrice diagonalisable. Mais on sait aussi comparer les puissances de deux matrices semblables. On a

$$A^n = PD^nP^{-1}.$$

Il faut donc retenir que si on connaît la matrice de passage P et la matrice diagonale D , on peut exprimer toutes les puissances de la matrice A (et donc le terme général du système linéaire associé à A).

Remarque 1.2.1. Attention, si la matrice A n'est pas diagonalisable, alors l'exercice est plus difficile et le sujet vous guidera pour trouver les puissances. Voir l'exercice suivant.

Exercice type concours.

ECRICOME 1988 Exercice 1.

Soit E un espace vectoriel rapporté à une base $\mathcal{B} = (e_1, e_2, e_3)$. On considère l'endomorphisme f de E dont la matrice relativement à \mathcal{B} est

$$A = \begin{pmatrix} 2 & 0 & 0 \\ 1 & 3 & -1 \\ 1 & 1 & 1 \end{pmatrix}$$

1. Vérifier que A n'admet comme valeur propre que le réel $\lambda = 2$.
2. Déterminer l'ensemble des vecteurs propres associés à $\lambda = 2$.

3. On pose

$$\begin{cases} e'_1 = & e_2 + e_3 \\ e'_2 = e_1 & + e_3 \\ e'_3 = e_1 + e_2 & \end{cases}$$

Montrer que la matrice A' de f relativement à la base $\mathcal{B} = (e'_1, e'_2, e'_3)$ est

$$A' = \begin{pmatrix} 2 & 0 & 2 \\ 0 & 2 & 0 \\ 0 & 0 & 2 \end{pmatrix}$$

4. On écrit $A' = 2I + J$ avec $I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$ et $J = \begin{pmatrix} 0 & 0 & 2 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$. Pour tout $n \in \mathbb{N}$, calculer A'^n puis A^n .

5. **Application.** Déterminer en fonction de n , les valeurs de x_n, y_n, z_n définis par $x_0 = y_0 = z_0 = 1$ et

$$\begin{cases} x_{n+1} = 2x_n \\ y_{n+1} = x_n + 3y_n - z_n \\ z_{n+1} = x_n + y_n + z_n \end{cases}$$

1.3. **Retour aux suites récurrentes, explication du théorème 1.1.** Pour établir un lien entre les suites récurrentes et les systèmes récurrents, nous devons tout d'abord transformer une suite récurrente en un système récurrent. Soit donc $(u_n)_{n \in \mathbb{N}}$ une suite qui vérifie la relation de récurrence d'ordre k ,

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \cdots + a_{k-1} u_{n+k-1},$$

où les réels a_0, a_1, \dots, a_{k-1} sont fixés. Pour chaque entier n , on définit le vecteur $X_n \in \mathcal{M}_{k,1}(\mathbb{R})$ par

$$X_n = \begin{pmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{pmatrix}$$

La relation de récurrence linéaire pour la suite $(u_n)_{n \in \mathbb{N}}$ se réécrit alors comme une relation de récurrence linéaire pour la suite de vecteurs $(X_n)_{n \in \mathbb{N}}$. En effet, on a

$$X_{n+1} = \begin{pmatrix} u_{n+1} \\ u_{n+2} \\ \vdots \\ u_{n+k} \end{pmatrix} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{k-1} \end{pmatrix} \cdot \begin{pmatrix} u_n \\ u_{n+1} \\ \vdots \\ u_{n+k-1} \end{pmatrix}$$

Définition : Matrice compagne d'une suite récurrente linéaire.

Soit $(u_n)_{n \in \mathbb{N}}$ une suite récurrente linéaire d'ordre k définie par la relation de récurrence

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \cdots + a_{k-1} u_{n+k-1}.$$

La matrice

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{k-1} \end{pmatrix}$$

s'appelle la matrice compagne de la suite $(u_n)_{n \in \mathbb{N}}$.

On dit que le système récurrent linéaire $(X_n)_{n \in \mathbb{N}}$ défini par $X_{n+1} = AX_n$ est associé à la suite récurrente $(u_n)_{n \in \mathbb{N}}$.

On obtient ainsi un deuxième moyen de trouver le terme général de la suite $(u_n)_{n \in \mathbb{N}}$.

Méthode : Terme général d'une suite récurrente linéaire.

On considère encore la suite définie par la relation

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \cdots + a_{k-1} u_{n+k-1}$$

et on cherche à donner son terme général.

1. On écrit la matrice compagne de la suite récurrence linéaire :

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{k-1} \end{pmatrix}$$

2. On cherche à diagonaliser cette matrice.
3. On en déduit les puissances de la matrice A .
4. Cela nous permet de donner le terme général du système récurrent linéaire $(X_n)_{n \in \mathbb{N}}$ associé à $(u_n)_{n \in \mathbb{N}}$.
5. On obtient en particulier le terme général de la suite.

Pour conclure ce paragraphe, il nous reste à comprendre le lien entre l'énoncé du théorème 1.1 et la méthode précédente, c'est-à-dire le lien entre les puissances de la matrice compagne et les solutions de l'équation caractéristique. Nous admettrons alors le résultat suivant, qui est hors programme et qui n'apparaît ici que pour montrer la cohérence des deux méthodes.

Théorème : Solutions de l'équation caractéristique et puissances de la matrices compagne (hors programme).

On considère une suite récurrente linéaire d'ordre k définie par la relation

$$u_{n+k} = a_0 u_n + a_1 u_{n+1} + \cdots + a_{k-1} u_{n+k-1},$$

où les réels a_0, a_1, \dots, a_{k-1} sont fixés. Comme précédemment, on associe à cette suite son équation caractéristique

$$(E) \quad x^k - a_{k-1}x^{k-1} - a_{k-2}x^{k-2} - \cdots - a_0 = 0$$

et sa matrice compagne

$$A = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \ddots & \cdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ a_0 & a_1 & a_2 & \cdots & a_{k-1} \end{pmatrix}$$

Alors les valeurs propres de A sont les solutions de l'équation caractéristique.

Remarque 1.3.1. Même si la preuve est trop technique pour être étudiée dans ce contexte général, il est fréquent qu'une question de concours demande de retrouver les valeurs propres d'une matrice compagne. C'est le cas par exemple de l'exercice 3 de la feuille d'exercice 09.

Ce résultat explique bien le lien entre la méthode du paragraphe 1.3 et le théorème 1.1. En effet, si on a réussi à diagonaliser la matrice compagne A , c'est-à-dire qu'on peut trouver une matrice $P \in \mathcal{M}_k(\mathbb{R})$ inversible et une matrice diagonale D telle que

$$A = PDP^{-1},$$

alors les valeurs propres sont les coefficients diagonaux de D . Puis les puissances de A sont données par la relation

$$A^n = PD^nP^{-1}.$$

La matrice D^n est une matrice diagonale dont les coefficients sont les puissances de valeurs propres de A . Ainsi les coefficients de la matrice A^n sont des combinaisons linéaires des puissances des valeurs propres de A . C'est exactement ce qu'affirme le théorème 1.1!

2. CHAÎNES DE MARKOV.

2.1. Rappels préliminaires sur les graphes.

2.1.1. Définitions.

Définition : Graphe.

On appelle *graphe (fini)* la donnée d'un couple $G = (S, A)$ où

- S est un ensemble fini, appelé ensemble des *sommets (ou noeuds)*
- A est un sous-ensemble de S^2 , appelé ensemble des *arêtes*.

Remarque 2.1.1. Il est important de noter qu'un sommet peut être relié à lui-même via une arête.

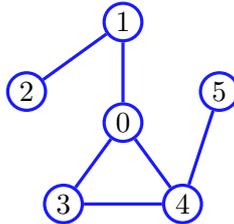
Définition : Graphe non orienté.

Soit $G = (S, A)$ un graphe. On dit que le graphe G est *non-orienté* si

$$\forall (x, y) \in S^2, (x, y) \in A \iff (y, x) \in A$$

Dans le cas contraire, on dit que le graphe G est orienté.

Exemple 2.1.2. Considérons le graphe non orienté

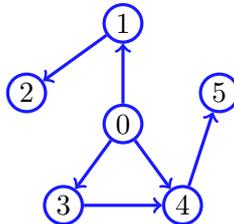


Il est mathématiquement modélisé par

$$S = \{0, 1, 2, 3, 4, 5\}$$

$$A = \{(0, 1), (0, 3), (0, 4), (1, 0), (1, 2), (2, 1), (3, 0), (3, 4), (4, 0), (4, 3), (4, 5), (5, 4)\}$$

Exemple 2.1.3. Considérons le graphe orienté



Il est mathématiquement modélisé par

$$S = \{0, 1, 2, 3, 4, 5\}$$

$$A = \{(0, 1), (0, 3), (0, 4), (1, 2), (3, 4), (4, 5)\}$$

Définition : Voisins, Voisinage, Degré.

Soit $G = (S, A)$ un graphe. Soit $x \in S$ un sommet.

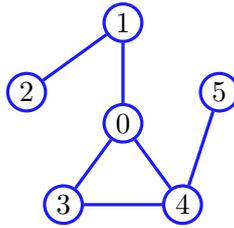
- Soit $y \in S$. On dit que y est un *voisin* de x si $(x, y) \in A$. Autrement dit, y est un voisin de x si x et y sont reliés par une arête partant de x .
- On appelle *voisinage* de x l'ensemble $V(x) \subset S$ des voisins de x , *i.e.*

$$V(x) = \{y \in S \mid (x, y) \in A\}$$

- On appelle *degré de x* le nombre de voisins de x . On note ce nombre $\deg(x)$, *i.e.*

$$\deg(x) = \text{Card}(V(x))$$

Exemple 2.1.4. Reprenons l'exemple du graphe



On a

- $V(0) = \{1, 3, 4\}$ et $\deg(0) = 3$
- $V(1) = \{0, 2\}$ et $\deg(1) = 2$
- $V(2) = \{1\}$ et $\deg(2) = 1$
- $V(3) = \{0, 4\}$ et $\deg(3) = 2$
- $V(4) = \{0, 3, 5\}$ et $\deg(4) = 3$
- $V(5) = \{4\}$ et $\deg(5) = 1$

2.2. Représentation d'un graphe.

2.2.1. Représentation via la matrice d'adjacence.

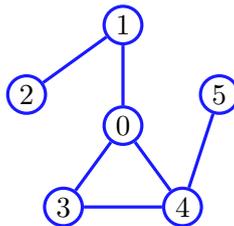
Définition : Matrice d'adjacence.

Soit $n \in \mathbb{N}^*$. Soit $G = (S, A)$ un graphe à n sommets. On note $S = \{s_1, s_2, \dots, s_n\}$. On appelle *matrice d'adjacence de G* la matrice $M = (m_{i,j})_{\substack{1 \leq i \leq n \\ 1 \leq j \leq n}} \in \mathcal{M}_n(\mathbb{R})$ définie par

$$\forall (i, j) \in \llbracket 1, n \rrbracket^2, m_{i,j} = 1_A((s_i, s_j)) = \begin{cases} 1 & \text{si } (s_i, s_j) \in A \\ 0 & \text{sinon} \end{cases}$$

Remarque 2.2.1. Si l'ensemble des sommets est fixé, alors la matrice d'adjacence caractérise le graphe. Ainsi, donner le graphe ou la matrice d'adjacence revient au même.

Exemple 2.2.2. Reprenons l'exemple du graphe



Sa matrice d'adjacence est

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

On remarque que la diagonale contient des 0. Ceci signifie qu'aucun des sommets n'est relié à lui-même.

On définit cette matrice en Python via les commandes

```
1 import numpy as np
2
3 M = np.array
4   ([[0, 1, 0, 1, 1, 0], [1, 0, 1, 0, 0, 0], [0, 1, 0, 0, 0, 0],
    [1, 0, 0, 0, 1, 0], [1, 0, 0, 1, 0, 1], [0, 0, 0, 0, 1, 0]])
```

On remarque que ${}^tM = M$. C'est en fait une propriété générale des graphes non orientés.

Comment calculer le degré d'un sommet à l'aide de la matrice d'adjacence ?

- Mathématiquement : soit $k \in S$.

$$\deg(k) = \sum_{j=1}^6 m_{k+1,j}$$

Attention au décalage de numérotation, les lignes et les colonnes des matrices sont numérotées à partir de 1.

- En Python :

```
1 for k in range(6):
2     deg = np.sum(M[k])
3     print('Le degre du sommet k est deg')
4
```

On prend en compte le fait que Python numérote à partir de 0.

Exercice 2.2.3. Tracer les graphes pour chacune des matrices d'adjacence suivantes. On numérotera les sommets à partir de 1.

1. $M = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}$

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

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

4. $M = \begin{pmatrix} 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 0 & 1 & 0 \end{pmatrix}$

2.2.2. Représentation via les listes d'adjacence.

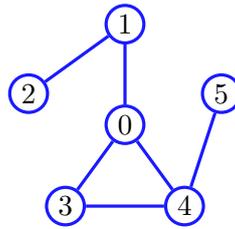
Définition : Liste d'adjacence.

Soit $G = (S, A)$ un graphe. Soit $x \in S$. On appelle *liste d'adjacence* du sommet x la liste des voisins de x . Autrement dit, la liste d'adjacence du sommet x contient les éléments du voisinage de x .

Remarque 2.2.4. L'ordre des éléments dans la liste d'adjacence de x n'a pas d'importance.

Remarque 2.2.5. Si l'ensemble des sommets est fixé, alors la liste des listes d'adjacence caractérise le graphe. Ainsi, donner le graphe ou la liste des listes d'adjacence revient au même.

Exemple 2.2.6. Reprenons l'exemple du graphe



En *Python*, la liste des listes d'adjacence peut être obtenue de plusieurs manières :

- De manière directe en lisant le graphe :

```
1 L = [[1, 3, 4], [0, 2], [1], [0, 4], [0, 3, 5], [4]]
```

- Via la matrice d'adjacence :

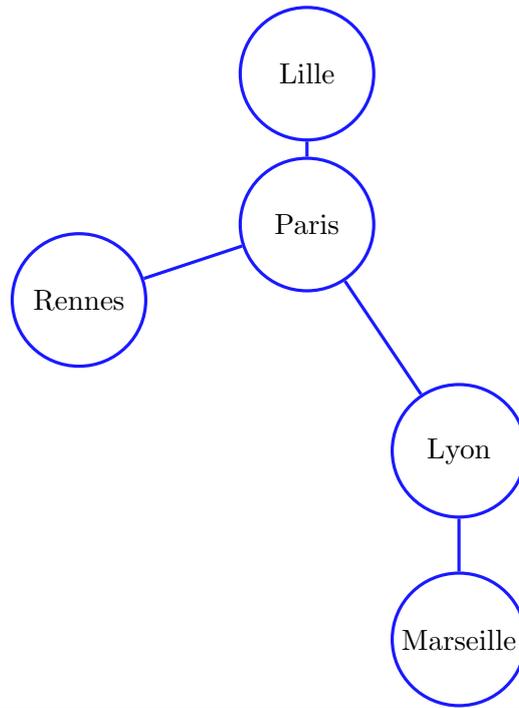
```
1 import numpy as np
2
3 def List_ad_from_Sym_matrix(M):
4     nbS = np.shape(M)[0]
5     L = [[] for k in range(nbS)]
6     for i in range(nbS):
7         for j in range(i+1, nbS):
8             if M[i, j] == 1:
9                 L[i].append(j)
10                L[j].append(i)
11     return L
12
```

Remarque 2.2.7. On a utilisé dans le programme précédent le fait que la matrice d'adjacence est symétrique. Ceci a permis de ne visiter que la partie triangulaire supérieure de la matrice.

Si le graphe est orienté, il faut tester tous les coefficients de la matrice :

```
1 import numpy as np
2
3 def List_ad_from_matrix(M):
4     nbS = np.shape(M)[0]
5     L = [[] for k in range(nbS)]
6     for i in range(nbS):
7         for j in range(nbS):
8             if M[i, j] == 1:
9                 L[i].append(j)
10    return L
11
```

Exemple 2.2.8. On peut aussi rassembler les listes d'adjacence dans un dictionnaire. Exemples d'une partie du réseau ferré français :



```

1 Dictionnaire = ['Paris': ['Rennes', 'Lyon', 'Lille'], 'Rennes': ['Paris'],
2   'Lyon' : ['Paris', 'Marseille'], 'Marseille': ['Lyon'], 'Lille': ['Paris']]
3

```

On accède aux voisins de 'Paris' en rentrant dans la console la commande `Dictionnaire['Paris']`. La console renvoie ['Rennes', 'Lyon', 'Lille'].

2.3. Propriétés des graphes.

Définition : Chemins.

Soit $G = (S, A)$ un graphe. Soient x et y deux sommets de G . Soit $k \in \mathbb{N}^*$.

- On dit qu'il existe un *chemin de longueur k allant de x à y* si on peut relier x à y par k arêtes successives. Plus précisément, si

$$\exists (s_0, s_1, \dots, s_k) \in S^{k+1}, \begin{cases} x = s_0 \\ y = s_k \\ \forall i \in \llbracket 0, k-1 \rrbracket, (s_i, s_{i+1}) \in A \end{cases}$$

Si c'est le cas, on note $x \xrightarrow{k} y$.

- Par convention, on dit qu'il existe un chemin de longueur 0 allant de x à lui-même : $x \xrightarrow{0} x$.
- On dit qu'il existe un *chemin allant de x à y* (on dit aussi que *textitx* mène à *y*) si il existe $k \in \mathbb{N}$ et un chemin de longueur k allant de x à y . Si c'est le cas, on note $x \rightarrow y$.
- Si x mène à y , on note

$$d(x, y) = \min_{k \in \mathbb{N}} \{x \xrightarrow{k} y\}$$

et on appelle ce nombre la *distance de x à y* .

Remarque 2.3.1. La relation \rightarrow est transitive : si $x \rightarrow y$ et $y \rightarrow z$, alors $x \rightarrow z$. Il suffit de concaténer les chemins.

Définition : Graphe connexe.

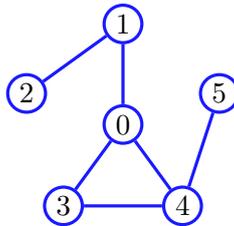
Soit $G = (S, A)$ un graphe. On dit que G est *connexe* si

$$\forall (x, y) \in S^2, x \rightarrow y$$

Autrement dit, G est connexe si tous les sommets sont reliés entre eux par un chemin.

Remarque 2.3.2. Le mot connexe est de la même famille que le mot connecté : G est connexe si tous ses sommets sont connectés.

Exemple 2.3.3. Reprenons l'exemple du graphe



Ce graphe est connexe et on a, par exemple,

- $d(0, 1) = 1$
- $d(0, 2) = 2$
- $d(2, 5) = 4$

2.4. Graphes pondérés.

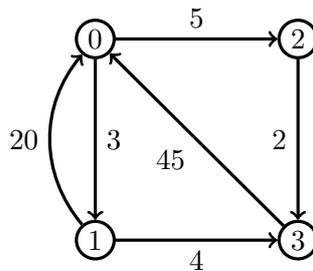
Définition : Graphe pondéré.

On appelle *graphe (fini) pondéré* la donnée d'un triplet $G = (S, A, \nu)$ où

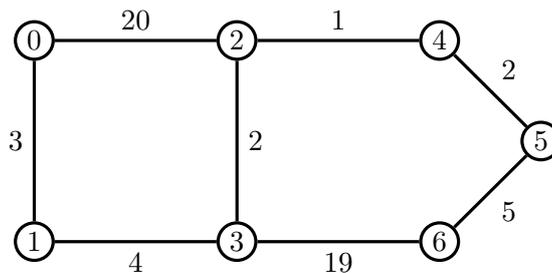
- (S, A) est un graphe
- $\nu : A \rightarrow]0, +\infty[$

Pour toute arête $a \in A$, le réel $\nu(a)$ est appelé *poids* associé à cette arête.

Exemple 2.4.1. Exemple d'un graphe pondéré et orienté.

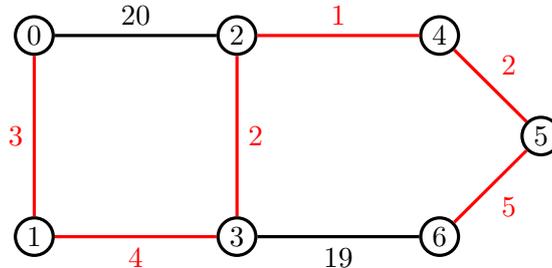


Exemple 2.4.2. Exemple de graphe pondéré et non orienté. Implémentation de l'algorithme de Dijkstra sur ce graphe.



0	1	2	3	4	5	6
0	∞	∞	∞	∞	∞	∞
—	3 ₀	20 ₀	∞	∞	∞	∞
—	—	20 ₀	7 ₁	∞	∞	∞
—	—	9 ₃	—	∞	∞	26 ₃
—	—	—	—	10 ₂	∞	26 ₃
—	—	—	—	—	12 ₄	26 ₃
—	—	—	—	—	—	• ₅

Visualisation du chemin le plus court :



2.5. Un peu de théorie.

2.5.1. Chaîne de Markov, matrice de transition et graphe associé.

Définition : Chaîne de Markov.

On considère une suite de variables aléatoires discrètes $(X_n)_{n \in \mathbb{N}}$ définies sur le même univers Ω et à valeurs dans le même ensemble E (c'est-à-dire que les variables X_n ont toutes le même support), qui sera bien souvent fini. On dit que $(X_n)_{n \in \mathbb{N}}$ est une chaîne de Markov si, pour tout $n \in \mathbb{N}$ et pour toute suite (i_0, \dots, i_n, j) d'éléments de E ,

$$\mathbb{P}_{(X_0=i_0 \cap X_1=i_1 \cap \dots \cap X_n=i_n)}(X_{n+1} = j) = \mathbb{P}_{(X_n=i_n)}(X_{n+1} = j).$$

Autrement dit, dans un chaîne de Markov, le futur X_{n+1} ne dépend que du présent X_n et pas du passé X_0, X_1, \dots, X_{n-1} .

On dit que E est l'ensemble des états de la chaîne de Markov.

Remarque 2.5.1. Dans toute la suite de ce texte, nous allons supposer que la chaîne de Markov est **homogène dans le temps**, c'est-à-dire que la probabilité

$$\mathbb{P}_{(X_n=i_n)}(X_{n+1} = j)$$

ne dépend pas de n . Ce sera toujours le cas dans les sujets de concours et nous supposons toujours que c'est le cas ici.

Si pour l'instant, le contexte semble être complètement probabiliste, il se trouve que l'algèbre linéaire joue un rôle fondamental dans l'étude des chaînes de Markov. On peut en effet encoder toute l'information contenue dans une chaîne de Markov (homogène) dans une matrice :

Définition : Matrice de transition.

Soit $(X_n)_{n \in \mathbb{N}}$ une chaîne de Markov. On suppose que l'ensemble des états de la chaîne est fini et on note

$$E = \{1, 2, \dots, k\}$$

Soit $m_{i,j} = \mathbb{P}_{(X_n=i)}(X_{n+1} = j)$. La matrice

$$M = (m_{i,j})_{i,j \in \llbracket 1,k \rrbracket} = \begin{pmatrix} m_{1,1} & m_{1,2} & \cdots & m_{1,k} \\ m_{2,1} & m_{2,2} & \cdots & m_{2,k} \\ \vdots & \vdots & \cdots & \vdots \\ m_{k,1} & m_{k,2} & \cdots & m_{k,k} \end{pmatrix}$$

s'appelle la matrice de transition de la chaîne de Markov (X_n) .

Remarque 2.5.2. 1. Pour tout i , la somme

$$\sum_{j=1}^k m_{i,j} = 1.$$

En effet, ceci signifie que X_n est bien une loi de probabilité.

2. On peut reformuler cette remarque en disant que 1 est valeur propre de la matrice et que le

vecteur $v = \begin{pmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{pmatrix}$ est propre pour la valeur propre 1.

On peut donc décrire une chaîne de Markov par sa matrice de transition. En fait, il existe encore une autre façon de représenter la même chaîne de Markov, sous la forme d'un graphe.

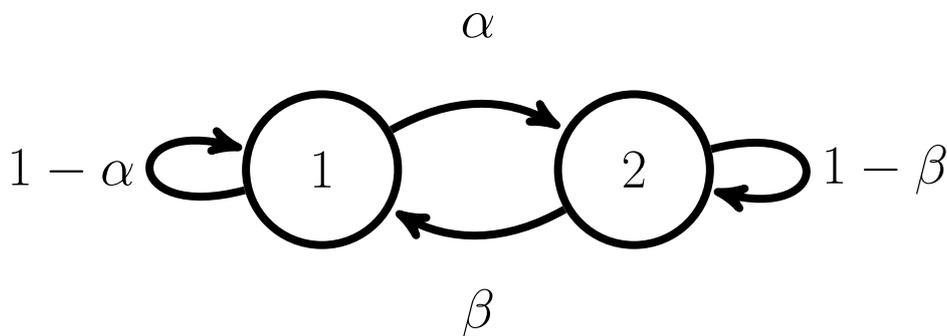
Définition : Graphe associé à une matrice de transition.

À toute matrice de transition M , on peut associer son graphe. Les **sommets** du graphe sont les différents états de la chaîne de Markov. Il y a une **flèche** étiquetée $m_{i,j}$ entre le sommet étiqueté i et le sommet étiqueté j si et seulement si la probabilité de transition de l'état i à l'état j est strictement positive et vaut $m_{i,j}$.

Exemple 2.5.3. la chaîne à deux états est la chaîne de Markov dont la matrice de transition est donnée par

$$M = \begin{pmatrix} 1 - \alpha & \alpha \\ \beta & 1 - \beta \end{pmatrix}$$

Son graphe associé est le graphe suivant.



Exercice 2.5.4. Dans chacun des cas suivants, donner la matrice de transition de la chaîne et le graphe associé.

1. **Modèle de diffusion d'Ehrenfest.** Deux urnes A et B contiennent à elles deux, a boules numérotées de 1 à a . À chaque instant, on choisit un nombre de 1 à a uniformément au hasard. Si ce nombre est i , on change d'urne la boule numéro i . L'ensemble des états est l'ensemble $E = \{0, \dots, a\}$ et la chaîne est dite dans l'état i si l'urne A contient i boules. Par exemple, si la chaîne est dans l'état 0, cela signifie que l'urne A est vide, tandis que si la chaîne est dans l'état a , alors c'est l'urne B qui est vide.
2. **Modèle de la ruine du joueur.** Un joueur A joue contre un joueur B une suite de parties de "pile" ou "face". Les parties sont indépendantes et la probabilité d'obtenir "pile" est de p . La somme totale des fortunes de A et de B est égale à a . À chaque partie, le joueur A mise sur "pile" et s'il gagne, il prend un euro à B et s'il perd, il lui donne un euro. On note X_n la fortune de A après n parties. La suite (X_n) est une chaîne de Markov.

2.5.2. *La relation de Chapman-Kolmogorov.* Dans ce paragraphe, (X_n) est une chaîne de Markov homogène dans le temps, $E = \{1, \dots, k\}$ est l'ensemble des états de la chaîne, M est la matrice de transition. On note à nouveau $M = (m_{i,j})_{i,j \in \llbracket 1, k \rrbracket}$. Pour tout $n \geq 0$ et $i, j \in E$, on note $m_{i,j}^{(n)}$ la probabilité que, partant de l'état i au temps 0, la chaîne soit dans l'état j au moment n . Autrement dit

$$m_{i,j}^{(n)} = \mathbb{P}_{(X_0=i)}(X_n = j).$$

On pose également

$$M^{(n)} = \left(m_{i,j}^{(n)} \right)_{i,j \in \llbracket 1, k \rrbracket}.$$

On dit que $M^{(n)}$ est la matrice de transition en n étapes.

Théorème : Relation de Chapman-Kolmogorov

$$M^{(n)} = M^n$$

Autrement dit, la matrice de transition en n étapes est égale à la puissance n ème de la matrice de transition en 1 étape.

Démonstration. La preuve est hors programme mais pas si difficile. On procède par récurrence sur n en utilisant judicieusement la formule des probabilités totales. Certaines questions difficiles de concours peuvent s'en inspirer.

□

2.5.3. *État stationnaire.* La relation de Chapman-Kolmogorov montre qu'il y a encore une fois, un intérêt à comprendre les puissances d'une certaine matrice (la matrice de transition dans le cas d'une chaîne de Markov). On peut ainsi espérer comprendre le comportement de la chaîne de Markov à long terme. Nous terminons cet aperçu théorique en formalisant ces phénomènes de convergence sur le long terme, sur lesquels nous reviendront au chapitre 14 de ce cours.

Ici encore, on considère une chaîne de Markov homogène $(X_n)_{n \in \mathbb{N}}$ dont l'ensemble des états $E = \llbracket 1, k \rrbracket$ est fini.

Définition : Probabilité initiale.

Soit $(\mu_i)_{i \in E}$ une suite de nombres positifs tels que $\sum_{i=1}^k \mu_i = 1$. Alors on définit une probabilité μ sur l'ensemble E par

$$\mu(i) = \mu_i.$$

On l'appelle probabilité initiale de la chaîne de Markov.

Le but de ce paragraphe est de montrer que la probabilité initiale et la matrice de transition permettent de définir une probabilité sur Ω (l'univers sur lequel sont définies les variables X_n). On pourra ainsi dire à quelle probabilité une certaine trajectoire de la chaîne de Markov est susceptible de se réaliser.

Définition : Probabilité d'une chaîne de Markov

Soit (X_n) une chaîne de Markov, de matrice de transition M et de probabilité initiale μ sur l'ensemble des états E . On dit que \mathbb{P} est la probabilité de la chaîne de Markov si les propriétés suivantes sont réalisées.

1. Pour tout état i , on a $\mathbb{P}(X_0 = i) = \mu_i$.
2. Pour tout couple d'état (i, j) , on a $\mathbb{P}_{X_0=i}(X_1 = j) = m_{i,j}$.

La probabilité \mathbb{P} (et la formule des probabilités totales) permet de trouver la loi des variables X_n (en fonction bien sûr de la probabilité initiale). Par exemple on a, pour tout état j ,

$$\mathbb{P}(X_1 = j) = \sum_{i=1}^k \mathbb{P}_{X_0=i}(X_1 = j)\mathbb{P}(X_0 = i) = \sum_{i=1}^k m_{i,j}\mu_i.$$

Définition : Probabilité stationnaire ou état stable de la chaîne de Markov.

On dit que la probabilité μ est stationnaire si la loi de X_1 est égale à la loi de X_0 . Autrement dit

$$\sum_{i=1}^k m_{i,j}\mu_i = \mu_j.$$

Un résultat célèbre (mais largement hors programme) prétend qu'il existe toujours une probabilité stationnaire¹.

Théorème : Existence d'une probabilité stationnaire.

Toute chaîne de Markov admet une probabilité stationnaire. De plus (sous certaines conditions), quelle que soit la probabilité initiale μ , la loi de (X_n) converge vers une probabilité stationnaire.

Ce théorème est hors programme mais beaucoup de sujets de concours ont pour but de nous faire trouver la probabilité stationnaire (dans certains cas particuliers). Bien sûr, comme il est courant dans ce chapitre, ce résultat est en fait un résultat d'algèbre. En effet, une probabilité stationnaire n'est rien d'autre qu'un vecteur propre pour la transposée de la matrice de transition et pour la valeur propre 1. Attention, même si l'on sait en général que 1 est valeur propre de la matrice (on peut montrer qu'une matrice est semblable à sa transposée), il n'est pas du tout évident que l'on puisse trouver un vecteur propre dont toutes les coordonnées soient positives (ce qui est indispensable pour que ce vecteur soit une loi de probabilité sur l'ensemble des états), c'est ce qui rend d'ailleurs la preuve de ce théorème particulièrement délicate.

2.6. Deux exemples de sujets de concours.

2.6.1. **ECRICOME 1992 Exercice 2.** Un distributeur de jouets distingue trois catégories de jouets :

- T : les jouets traditionnels tels que poupées, peluches,...
- M : les jouets liés à la mode inspiré directement d'un livre, d'un film, d'une émission.
- S : les jouets scientifiques vulgarisant une technique récente.

Il estime que

- Le client qui a acheté un jouet traditionnel une année pour Noël choisira, l'année suivante, un jouet de l'une des trois catégories avec équiprobabilité.

1. Lire par exemple le chapitre 5 de l'excellent livre *Processus stochastique* de D. Foata et A. Fuchs

- Le client qui a acheté un jouet inspiré par la mode choisira l'année suivante
 - Un jouet T avec probabilité $1/4$.
 - Un jouet M avec probabilité $1/4$.
 - Un jouet S avec probabilité $1/2$.
- Le client qui a acheté un jouet scientifique choisira l'année suivante
 - Un jouet T avec probabilité $1/4$.
 - Un jouet M avec probabilité $1/2$.
 - Un jouet S avec probabilité $1/4$.

Le volume des ventes de ce commerçant vient de se composer

- D'une part $p_0 = \frac{45}{100}$ de jouets de la catégorie T.
- D'une part $q_0 = \frac{25}{100}$ de jouets de la catégorie M.
- D'une part $r_0 = \frac{30}{100}$ de jouets de la catégorie S.

On désigne par p_n, q_n, r_n les parts respectives des jouets T, M et S dans les ventes du commerçant le n ème Noël suivant.

1. Montrer que le triplet $(p_{n+1}, q_{n+1}, r_{n+1})$ s'exprime en fonction du triplet (p_n, q_n, r_n) au moyen d'une matrice A que l'on formera.²
2. Soit P la matrice définie par $P = \begin{pmatrix} 3 & 2 & 0 \\ 4 & -1 & 1 \\ 4 & -1 & -1 \end{pmatrix}$
 - a. Montrer à l'aide de la méthode du pivot de Gauss que P est inversible et déterminer P^{-1} .
 - b. Déterminer la matrice $D = P^{-1}AP$.
 - c. Montrer par récurrence sur $n \in \mathbb{N}^*$ que $A^n = PD^nP^{-1}$.
 - d. Calculer A^n pour tout $n \in \mathbb{N}^*$.
3. Exprimer (p_n, q_n, r_n) directement en fonction de n .
4. Quelle parts à long terme les trois catégories de jouets représentent-elles dans la vente si l'attitude des consommateurs reste constante.

2.6.2. EDHEC 2017 Problème.

Partie I : étude d'une variable aléatoire.

Les sommets d'un carré sont numérotés 1,2,3 et 4 de telle façon que les côtés du carré relient le sommet 1 au sommet 2, le sommet 2 au sommet 3, le sommet 3 au sommet 4 et le sommet 4 au sommet 1.

Un mobile se déplace aléatoirement sur les sommets de ce carré selon le protocole suivant :

- Au départ, c'est-à-dire à l'instant 0, le mobile se trouve sur le sommet 1.
- Lorsque le mobile est à un instant donné sur un sommet, il se déplace à l'instant suivant sur l'un quelconque des trois autres sommets, et ceci de façon équiprobable.

Pour tout n de \mathbb{N} , on note X_n la variable aléatoire égale au numéro du sommet sur lequel se situe le mobile à l'instant n . D'après les deux points précédents, on a donc $X_0 = 1$.

1. Donner la loi de X_1 , ainsi que l'espérance de la variable aléatoire X_1 .

On admet pour la suite que la loi de X_2 est donnée par

$$P(X_2 = 1) = \frac{1}{3} \quad P(X_2 = 2) = P(X_2 = 3) = P(X_2 = 4) = \frac{2}{9}.$$

2. J'ai recopié le texte original du sujet qui est **FAUX** dans cet état. Il faut en fait considérer les vecteurs-colonne $\begin{pmatrix} p_n \\ q_n \\ r_n \end{pmatrix}$.

2. Pour tout entier supérieur ou égal à 2, donner, en justifiant, l'ensemble des valeurs prises par X_n .
3. a. Utiliser la formule des probabilités totales pour établir que, pour tout entier n supérieur ou égal à 2, on a

$$P(X_{n+1} = 1) = \frac{1}{3} (P(X_n = 2) + P(X_n = 3) + P(X_n = 4)).$$

b. Vérifier que cette relation reste valable pour $n = 0$ et $n = 1$.

c. Justifier que, pour tout $n \in \mathbb{N}$, on a $P(X_n = 1) + P(X_n = 2) + P(X_n = 3) + P(X_n = 4) = 1$ et en déduire l'égalité, pour tout $n \in \mathbb{N}$,

$$P(X_{n+1} = 1) = -\frac{1}{3}P(X_n = 1) + \frac{1}{3}.$$

d. Établir alors que, pour tout $n \in \mathbb{N}$, $P(X_n = 1) = \frac{1}{4} + \frac{3}{4} \left(-\frac{1}{3}\right)^n$.

4. a. En procédant comme à la question précédente, montrer que l'on a, pour tout $n \in \mathbb{N}$,

$$P(X_{n+1} = 2) = \frac{1}{3} (P(X_n = 1) + P(X_n = 3) + P(X_n = 4)).$$

b. En déduire une relation entre $P(X_{n+1} = 2)$ et $P(X_n = 2)$.

c. Montrer enfin que pour tout $n \in \mathbb{N}$, $P(X_n = 2) = \frac{1}{4} - \frac{1}{4} \left(-\frac{1}{3}\right)^n$.

5. On admet enfin que pour tout entier naturel n , on a

$$P(X_{n+1} = 3) = -\frac{1}{3}P(X_n = 3) + \frac{1}{3}; \quad \text{et} \quad P(X_{n+1} = 4) = -\frac{1}{3}P(X_n = 4) + \frac{1}{3}.$$

En déduire sans calcul que, pour tout $n \in \mathbb{N}$,

$$P(X_n = 3) = P(X_n = 4) = \frac{1}{4} - \frac{1}{4} \left(-\frac{1}{3}\right)^n.$$

6. Déterminer, pour tout entier naturel n , l'espérance $E(X_n)$ de la variable aléatoire X_n .

Partie II : calcul des puissances d'une matrice A .

Pour tout n de \mathbb{N} , on considère la matrice ligne de $\mathcal{M}_{1,4}(\mathbb{R})$

$$U_n = (P(X_n) = 1 \quad P(X_n = 2) \quad P(X_n = 3) \quad P(X_n = 4)).$$

1. a. Montrer, grâce à certains résultats de la partie I. que, si l'on pose

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

on a, pour tout $n \in \mathbb{N}$,

$$U_{n+1} = U_n A.$$

b. Établir par récurrence que, pour tout $n \in \mathbb{N}$,

$$U_n = U_0 A^n.$$

c. En déduire la première ligne de A^n .

2. Expliquer comment choisir la position du mobile au départ pour trouver les trois autres lignes de la matrice A^n , puis écrire ces trois lignes.

Partie III : une autre méthode de calcul des puissances de A .

On considère les matrices I et J suivantes :

$$I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad \text{et} \quad J = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$$

1. Déterminer les réels a et b tels que $A = aI + bJ$.
2.
 - a. Calculer J^2 puis établir que, pour tout entier naturel k non nul, on a $J^k = 4^{k-1}J$.
 - b. À l'aide de la formule du binôme de Newton, en déduire, pour tout entier naturel n non nul, l'expression de A^n comme combinaison linéaire de I et J .
 - c. Vérifier que l'expression trouvée reste valable pour $n = 0$.

Partie IV : informatique.

1. Dans cette question, il était demandé de compléter un script informatique pour simuler la trajectoire des 100 premiers mouvements de la chaîne de Markov. Nous allons revenir dans le paragraphe suivant sur les simulations Python.
2. Après avoir exécuté cinq fois ce script, les réponses concernant le nombre de fois où le mobile est revenu sur le sommet 1 sont $n = 23$, $n = 28$, $n = 23$, $n = 25$ et $n = 26$. En quoi est-ce normal ?

2.6.3. *Autres idées de sujets de concours avec des chaînes de Markov.* Pour des sujets un peu plus difficiles sur le même thème, on pourra par exemple faire **ESSEC** 1995 épreuve 1, exercice 1, **ESSEC** 2008 épreuve 2 ou **HEC** 2021 Problème.

2.7. Python et chaînes de Markov (à faire en TP).

2.7.1. *Simuler une chaîne de Markov à l'aide de sa matrice de transition.* Regardons comment on peut simuler une trajectoire de chaîne de Markov avec sa matrice de transition. On suppose donc qu'on dispose de la matrice de transition

$$M = (m_{i,j}) = (\mathbb{P}_{(X_n=i)}(X_{n+1}=j))_{i,j}.$$

On écrit alors un programme qui simule la trajectoire des n premiers pas de la chaîne (n est à choisir par l'utilisateur) avec la position initiale x_0 (à choisir aussi par l'utilisateur).

```

1 import numpy as np
2
3 def markov_avec_matrice(n,M,x0):
4     X=np.zeros(n)
5     X[0]=x0
6     for k in range(n-1):
7         X[k+1]=np.random.choice(a=list(range(len(M))), p= M[int
            (X[k]),:])
8     return X

```

L'utilisation et la justification de la bonne marche de ce programme demandent quelques explications. La fonction `np.random.choice(a,p)` simule une variable aléatoire avec support dans a et dont les probabilités sont contenues dans p . Les objets a et p doivent être des listes de même taille.

Ici on choisit $a = \llbracket 0, n-1 \rrbracket$ (n est la taille de la matrice de transition). On identifie donc l'ensemble des états de la chaîne à $\llbracket 0, n-1 \rrbracket$. On rappelle que si M est une matrice carrée de taille $n \times n$, alors la commande `len(P)` renvoie l'entier n .

Ensuite, pour choisir les probabilités avec lesquelles on obtient p , il s'agit de savoir en quel état se trouve la chaîne, et utiliser les coefficients de la matrice de transition adaptés. Dans le k ème tour de

boucle, la chaîne se trouve donc dans l'état $X[k]$ et la loi de $X[k+1]$ s'obtient donc en conditionnant par rapport à $X_k = X[k]$. Les valeurs de la loi de X_{k+1} conditionnée à $X_k = X[k]$ sont sur la $X[k]$ ème ligne de la matrice, que l'on extrait donc avec la commande $M[X[k], :]$.

Nous verrons une autre méthode de simulation d'une chaîne de Markov (très similaire) dans le paragraphe suivant. Les deux sont à connaître absolument.

2.7.2. *Évolution d'une maladie.* On modélise l'évolution d'une maladie au sein d'une population au cours du temps en classant les individus en trois groupes :

- Le groupe S des individus sains et non immunisés ;
- Le groupe M des individus malades ;
- Le groupe I des individus immunisés.

On appelle $E = \{S, M, I\}$ l'ensemble des états possibles. On donne la **loi d'évolution** suivante :

- À l'instant 0, tous les individus sont dans l'état S .
- La moitié des individus dans l'état S à l'instant n restent dans le même état à l'instant $n+1$ et l'autre moitié tombe malade ; un dixième des individus immunisés à l'instant n passent dans le groupe S à l'instant $n+1$, les autres restent immunisés. Un malade sur cinq le reste, les autres guérissent et deviennent immunisés.

1. Représenter le modèle précédemment décrit par un graphe ;
2. On s'intéresse à la suite (X_n) de variables aléatoires telle que X_n désigne l'état du système l'instant n , c'est à dire que $X_n(\Omega) = \{1, 2, 3\}$, où $(X_n = 1)$ (resp. 2, 3) correspond à l'état S (resp. M, N) à l'instant n . La suite (X_n) est donc une chaîne de Markov.
 - a. Recopier et compléter le programme ci-dessous pour qu'il simule la suite des états X_0, X_1, \dots, X_n (c'est-à-dire une trajectoire de la chaîne).

```

1 import numpy as np
2
3 def Exemple(n):
4 X=np.zeros(n)
5 X[0]=1
6 for k in range (n-1) :
7     if X[k] == 1 :
8         X[k+1] = np.random.choice(np.array([1,2,3]),.....)
9     elif X[k] == 2 :
10        X[k+1] = .....
11    else :
12        X[k+1] = np.random.choice(np.array([1,2,3]),.....)
13    return X

```

- b. Recopier, exécuter et interpréter les commandes suivantes

```

1 import matplotlib.pyplot as plt
2
3 y = Exemple(1000)
4 plt.plot(y)
5 plt.show()

```

- c. Même question avec les commandes ci-dessous

```

1 U=np.zeros(100)
2 for k in range(100):
3     U[k]=Exemple(1000)[999];
4
5
6 frequ = np.zeros(3)
7 for j in range(3):
8     for k in range(len(U)):
9         if U[k] == j+1 :
10            frequ[j]=frequ[j]+1
11
12 plt.bar(frequ)
13 plt.show()

```

3. Pour $n \in \mathbb{N}$, on note

$$U_n = (P(X_n = 1), P(X_n = 2), P(X_n = 3)).$$

a. Déterminer la matrice de transition de la chaîne de Markov (elle vérifie donc

$$U_{n+1} = U_n A).$$

b. Montrer que, pour tout $n \in \mathbb{N}$, on a $U_n = U_0 A^n$. Préciser U_0 , (c'est la loi initiale).

c. Définir, dans la console, la matrice de transition A et le vecteur de la loi initiale u_0 , et calculer $U_0 A^{1000}$. Comparer la cohérence des résultats avec ceux de la Question 2c.

4. Utiliser le programme du paragraphe précédent pour réécrire une fonction qui simule la trajectoire de la même chaîne de Markov.

2.7.3. *Algorithme du Page Rank de Google.* Nous allons illustrer la méthode utilisée par Google pour classer les pages web par "indice de popularité". L'idée est d'étudier l'évolution des déplacements d'un individu sur internet, et de regarder la probabilité p_i d'être sur un site i au bout d'une très grand nombre de déplacements (on s'intéresse donc à la probabilité stationnaire de la chaîne). On dit alors que p_i est alors appelée "indice de popularité de la page i ". Nous commencerons par regarder un cas très simple (et très édulcoré!) où il n'existerait sur le Web que 3 pages internet (!) puis on étudiera le cas général.

2.8. **Un exemple avec 3 sites.** À l'issue de sa requête, un internaute est susceptible d'aller sur trois sites A , B , et C . On suppose que l'internaute se déplace forcément vers un lien de la page Web où il se trouve.

- le site A contient un lien vers lui-même, un lien vers B , et un lien vers C ;
- le site B contient 5 liens vers lui-même, un vers A et un vers C ;
- le site C comporte un lien vers A , 7 liens vers B et 4 vers C .

À l'instant 0, l'internaute choisit au hasard l'un des trois sites. Par la suite, la probabilité de passer d'un site (à l'instant n) vers un autre (à l'instant $n + 1$) est proportionnelle au nombre de liens du premier site vers le deuxième.

Pour $n \in \mathbb{N}$, on désigne par a_n , b_n et c_n les probabilités des événements A_n (resp. B_n , C_n) définis respectivement par le fait de se trouver sur le site A (resp. B , C) à l'instant n .

Enfin, on note X_n la variable aléatoire définie par

$$X_n(\omega) = \begin{cases} 1, & \text{si } \omega \in A_n \\ 2, & \text{si } \omega \in B_n \\ 3, & \text{si } \omega \in C_n \end{cases}$$

1. En justifiant rigoureusement au moins l'une des égalités, exprimer a_{n+1} , (resp. b_{n+1} , c_{n+1}) en fonction des trois réels a_n , b_n et c_n .
2. Écrire la matrice de transition S .
3. Mémoriser la matrice S dans votre éditeur Python puis compléter et exécuter le programme suivant qui permet d'afficher la trajectoire (X_0, X_1, \dots, X_n) de la chaîne de Markov (X_n) .

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 def navigation(n,P,x0):
5     X=np.zeros(n)
6     X[0]=.....
7     for k in range (n-1):
8         X[k+1]=np.random.choice(a=range(3), p= .....
9     return X
10
11 plt.plot(navigation(100))
12 plt.show()

```

4. Est-ce qu'un site à l'air plus visité que les autres? Moins visité que les autres? Comment l'expliquer?
5. Calculer avec Python les vecteurs de lois U_0, U_1, \dots, U_{10} . (On recopiera et complétera le programme suivant pour définir une matrice E de taille 3×11 dont la i -ème colonne contiendra le vecteur U_{i-1} .

```

1 E=np.zeros((3, 11))
2 E(:, 1)=(1/3)*np.ones((3,1))
3 for k in range (9)
4     E(:, k+2)=.....

```

6. Tracer sur un même graphique les courbes des suites (a_n) , (b_n) et (c_n) . Que remarque-t-on?
7. On appelle "indice de notoriété" d'une page A la valeur $a = \lim a_n$. Donner une valeur approchée.
8. Regarder la valeur exacte des indices de notoriété en calculant la valeur exacte de l'état stable. On admettra que la probabilité invariante est unique.

On rappelle que la mesure invariante est un 1-vecteur propre de la matrice transposée de la matrice de transition. On rappelle que dans la bibliothèque `numpy.linalg`, la fonction `eig` sert à trouver le spectre et les espaces propres de P .

2.9. Le cas général. On modélise l'ensemble des pages du Web par un ensemble fini d'états $E = \{1, 2, \dots, N\}$ (le nombre N est bien très grand, de l'ordre de 10^{13} , mais reste fini).

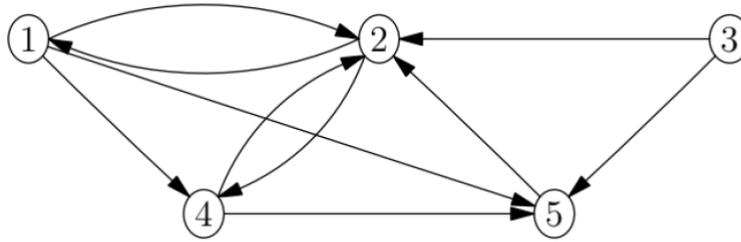
On modélise les déplacements de l'internaute par une chaînes de Markov (X_n) où la variable X_n est le numéro de la page où se situe l'internaute après n déplacements. Après avoir fait des études statistiques des comportements sur le Web, Google part du principe que :

- À l'instant 0, l'internaute choisit une page internet au hasard parmi les N pages ;
- Lorsqu'un internaute est sur une page j :
 - dans 85% des cas, il se déplacera au hasard vers l'une des ℓ_j pages pointées par j (ici, une page a au plus un lien vers une autre).
 - dans 15% des cas, il se déplacera au hasard vers n'importe laquelle des pages du Web.

La matrice de transition $M = (P(i, j))$ est alors définie par

$$P(i, j) = \begin{cases} 0,85 \times \frac{1}{\ell_j} + 0,15 \times \frac{1}{N}, & \text{si } j \text{ pointe vers } i \\ 0,15 \times \frac{1}{N}, & \text{sinon} \end{cases}$$

Nous allons illustrer l'algorithme de PageRank sur un groupe autonome de $N = 5$ pages différentes, ayant des liens pointant les pages les unes vers les autres, selon le diagramme suivant :



1. Taper et expliquer la commande

```

1 import numpy as np
2
3 A=0.15*(1/5)*np.ones((5,5))+0.85*np.array([[0, 1/2, 0, 0, 0], [1/3, 0, 1/2,
    1/2, 0], [0, 0, 0, 0, 0], [1/3, 1/2, 0, 0, 0], [1/3, 0, 1/2, 1/2, 1]])
  
```

2. Déterminer la valeur exacte de la loi stationnaire (à l'aide de la commande eig()).

3. Recopier et exécuter les commandes suivantes. La chaîne converge-t-elle vers l'état stationnaire ?

```

1 import matplotlib.pyplot as plt
2
3 E=np.zeros((5, 11))
4 E(:, 0)=(1/5)*np.ones((5,1))
5 for k in range (10):
6     E(:, k+1)=A np.dot E(:, k)
7
8 plt.bar([E(1,:),E(2,:),E(3,:),E(4,:),E(5,:)])
9 plt.show()
  
```

4. Classer les pages par indice de popularité. Les résultats vous semblent-ils cohérents ? Commenter.

3. SUJETS D'ANNALES EN LIEN AVEC CE CHAPITRE.

Remarque 3.0.1. Le contenu de ce chapitre constitue une thématique récente du programme mais elle est amenée à prendre de l'importance à partir de la session 2023.

1. ECRICOME

- 2022 Exercice 1.

2. EDHEC

-

3. EML

- 2023 (sujet 0) Exercice 3.

4. HEC/ESSEC

-