# TD – Graphes

*3ème partie*

## Introduction et objectifs

Dans cette troisième partie du TD consacré aux graphes, l'objectif principal est de se familiariser avec quelques aspects des chaînes de Markov homogènes qui permettent de modéliser des situations variées.

On utilisera les matrices via la bibliothèque `numpy` (importée avec l'alias `np`) et, plus particulièrement, le module `linalg` pour les éventuelles « manipulations » du domaine de l'algèbre linéaire (comme l'obtention des valeurs propres d'une matrice carrée).

## Matrices stochastiques

### Exercice N°1

Ecrire une fonction Python `IsMatStoch` qui reçoit en argument un tableau `numpy` `A` et renvoie `True` ou `False` selon que `A` correspond ou non à une matrice stochastique.

Si `A` n'est pas du type `np.array`, on génèrera une erreur de type et si `A` n'est pas carrée, on génèrera une erreur de valeur.

### Exercice N°2

Ecrire une fonction Python `MatStoch_eig` qui reçoit en argument un tableau `numpy A` et renvoie la liste de ses valeurs propres triée dans l'ordre décroissant.

La fonction :

- vérifiera que le tableau `A` correspond bien à une matrice stochastique (et renverra un message d'erreur sinon).
- Utilisera la fonction `eigvals` du module `linalg` de la bibliothèque `numpy`.

## Chaînes de Markov homogènes

### Exercice N°3 – On a le temps !

Note : cet exercice est une application directe du cours et ne requiert pas de manipulations informatiques.

Dans un pays imaginaire les habitants s'intéressent au temps qu'il fait et, classiquement, à celui qu'il fera… en s'appuyant sur les données passées. Ils ont observé que :

- la probabilité que le temps reste inchangé est de 0,6.
- s'il pleut, la probabilité qu'il fasse beau le lendemain est de 0,1 mais qu'il neigera avec une probabilité de 0,3.
- s'il fait beau, la probabilité qu'il pleuve le lendemain est de 0,2 et qu'il neigera avec une probabilité de 0,2 également.
- s'il neige, la probabilité qu'il fasse beau le lendemain est de 0,4.

L'espace des états est noté $\mathscr{E} = \{P\ ;N\ ;B\}$.

1. Traduire la situation proposée à l'aide d'un graphe probabiliste et en donner la matrice de transition *T*.
2. Vérifier que 1 est valeur propre simple et dominante de *T*.
3. En modélisant l'évolution du temps qu'il fait (fera ? ☺) à l'aide d'une chaîne de Markov homogène, déterminer la distribution de probabilités limite en résolvant un système. Donner alors une interprétation du résultat.

### Exercice N°4 – Sauts de puce.

On considère *N* points (numérotés de 1 à *N*) sur une droite ( $N \geq 4$ ).

Une puce se trouve initialement en l'un de ces points et, à tout instant (0, 1, 2, …, *n*) se déplace en sautant d'un point à un autre aléatoirement : vers la gauche avec une probabilité *p* et vers la droite avec une probabilité *q*. Dès que la puce atteint le point 1 ou le point *N*, elle s'arrête de sauter et reste donc en ce point.

La position initiale de la puce est donnée par la matrice ligne $\pi_0 = \begin{pmatrix} 0 & 0 & \cdots & 1 & \cdots & 0 & 0 \end{pmatrix}$ ne comportant qu'un seul 1 (les autres coefficients étant nuls) en position $i_0 \in [\![1, N]\!]$.

1. Représenter la situation à l'aide d'un graphe probabiliste et donner sa matrice de transition (notée *T*).
2. Démontrer que 1 est valeur propre de *T* d'ordre de multiplicité égal à 2.

3. On prend $N = 4$ et $p = q = \dfrac{1}{2}$.
   A l'aide de la fonction `matrix_power` de `numpy.linalg`, calculer $T^{10}$, $T^{50}$ et $T^{100}$.
   Conjecturer $T^{\infty} = \lim\limits_{n \to +\infty} T^n$.

4. On cherche à « valider » la conjecture précédente en écrivant un script complet.
   Les variables principales du programme seront :

   - Le nombre de points `N`.
   - Les probabilités `p` et `q`.

   Les valeurs initiales de `T` et `p` seront fournies par l'utilisateur.

   Les étapes du script seront :

   a. Construction de la matrice $T$.
   b. Détermination des valeurs propres et des vecteurs propres de $T$ à l'aide de la fonction `eig` du module `numpy.linalg`.
   c. Construction de $\lim\limits_{n \to +\infty} D^n$ où $D$ est la matrice diagonale comportant les valeurs propres de $T$ sur la diagonale ($T = P \times D \times P^{-1}$ où $P$ est la matrice de passage).
   d. Construction et affichage de $\lim\limits_{n \to +\infty} T^n = P \times \lim\limits_{n \to +\infty} D^n \times P^{-1}$.

5. Donner une interprétation du résultat obtenu.

### Exercice N°5 – Séquences de piles.

On jette une pièce de monnaie bien équilibrée $N$ fois. On note PILE et FACE les résultats possibles de chaque lancer.

On cherche la probabilité d'obtenir au moins une séquence d'au moins $n$ PILE consécutifs (avec $1 \leq n \leq N$).

Dans un premier temps, on suppose que l'on a : $N = 100$ et $n = 5$.

1. Ecrire un script Python permettant d'effectuer *ns* (variable `ns` dont la valeur sera saisie par l'utilisateur) simulations de 100 lancers et calculant le pourcentage de ces simulations où au moins une séquence d'au moins 5 PILE a été observée.

2. On considère l'ensemble des états $\mathscr{E}$ suivant $\mathscr{E} = \{\varnothing ; P ; PP ; PPP ; PPPP ; PPPPP\}$ où :

   - $\varnothing$ correspond à l'état : « la partie n'a pas commencé ou (on n'a pas encore obtenu une séquence d'au moins 5 PILE et on vient d'obtenir 1 FACE) ».
   - $P$ correspond à l'état : « on n'a pas encore obtenu une séquence d'au moins 5 PILE et on vient d'obtenir une séquence de 1 PILE ».

- *PP* correspond à l'état : « on n'a pas encore obtenu une séquence d'au moins 5 PILE et on vient d'obtenir une séquence de 2 PILE ».
- *PPP* correspond à l'état : « on n'a pas encore obtenu une séquence d'au moins 5 PILE et on vient d'obtenir une séquence de 3 PILE ».
- *PPPP* correspond à l'état : « on n'a pas encore obtenu une séquence d'au moins 5 PILE et on vient d'obtenir une séquence de 4 PILE ».
- *PPPPP* correspond à l'état : « on a obtenu au moins une séquence d'au moins 5 PILE ».

a. Vérifier que l'on a bien affaire à une chaîne de Markov homogène et en donner la matrice de transition.

b. En reprenant les notations du cours, on a : $\pi_0 = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$.
Calculer alors $\pi_{100}$ et donner la probabilité cherchée (on complètera le script précédent de sorte qu'il affiche le pourcentage calculé à la question 1 et la valeur théorique obtenue grâce à $\pi_{100}$).

On généralise la situation précédente. *N* et *n* sont désormais des paramètres du problème.

1. Donner la matrice de transition de la chaîne de Markov.
2. Modifier le script précédent pour traiter le cas général.