

EXERCICE 1

1. On tire une boule dans une urne contenant des boules blanches et des boules noires.

Ecrire une fonction `tirage` qui prend en paramètres le nombre de boules blanches `nb_b` et le nombre de boules noires `nb_n` et qui renvoie 0 si une boule blanche a été tirée et 1 sinon.

2. On considère l'expérience aléatoire suivante :

Initialement l'urne contient une boule blanche et une boule noire. On effectue n tirages dans cette urne. A l'issue de chaque tirage, on remet la boule tirée dans l'urne et on y ajoute une boule de la même couleur que la boule tirée.

- (a) Ecrire une fonction `listeTirages` qui prend en argument le nombre n de tirages, qui simule l'expérience aléatoire et renvoie la liste des boules tirées (0 représentant une boule blanche et 1 une boule noire).
- (b) Ecrire une fonction `estimeProba` qui prend en argument le nombre de simulations `nb_simul` et le nombre de tirages par simulation n , et qui renvoie une estimation de la probabilité de n'obtenir que des boules blanches au cours de n tirages.
- (c) Conjecturer la probabilité de n'obtenir que des boules blanches suite aux n tirages. On pourra d'abord examiner les cas $n = 3$, $n = 4$ et $n = 9$.

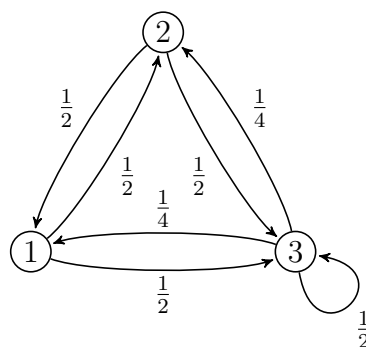
3. Démontrer la conjecture émise à la question 2 (c).

EXERCICE 2

Un groupe de chercheurs s'intéresse au comportement d'une souris lorsqu'elle est placée dans un labyrinthe. Ce labyrinthe comporte trois croisements numérotés 1, 2 et 3, ainsi que des chemins les reliant. Les flèches indiquent les déplacements possibles, et les nombres portés le long des flèches indiquent les probabilités respectives que la souris choisisse ces déplacements.

À l'instant initial $n = 0$, on place la souris sur le croisement numéro 1.

On représente la situation par le graphe ci-dessous :



Pour tout $n \in \mathbb{N}$, on définit les événements :

- A_n : «la souris se trouve au croisement numéro 1 après le $n^{\text{ième}}$ déplacement» ;
- B_n : «la souris se trouve au croisement numéro 2 après le $n^{\text{ième}}$ déplacement» ;
- C_n : «la souris se trouve au croisement numéro 3 après le $n^{\text{ième}}$ déplacement».

Pour tout entier naturel n , on note aussi Y_n le vecteur colonne défini par $Y_n = \begin{pmatrix} P(A_n) \\ P(B_n) \\ P(C_n) \end{pmatrix}$.

1. Déterminer Y_0 et Y_1 .

2. On note A la matrice de $\mathcal{M}_3(\mathbb{R})$ définie par $A = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 0 & \frac{1}{4} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \end{pmatrix}$.

(a) Démontrer que, pour tout entier naturel n , $Y_{n+1} = AY_n$.

(b) En déduire que, pour tout entier naturel n , $Y_n = A^n Y_0$.

3. Importer le module `numpy` avec l'alias `np`.

(a) Créer les matrices A et Y_0 .

(b) Écrire une fonction Python `probas`, qui prend en argument un entier naturel n , et qui renvoie la matrice Y_n .

(c) Écrire une fonction Python `choix`, qui prend en argument un entier naturel s compris entre 1 et 3, et qui renvoie un entier naturel compris entre 1 et 3 correspondant au numéro du sommet choisi au hasard par la souris, lorsqu'elle se trouve au sommet numéro s .

(d) En déduire une fonction Python `position`, qui prend en argument un entier naturel n , et qui renvoie la position de la souris après n déplacements.

(e) Proposer une fonction Python permettant de comparer les probabilités obtenues grâce à la fonction `probas` et les fréquences empiriques que l'on peut déduire de la fonction `position`.

Corrigé de la question 3. exercice 1

Soit B_i l'événement : "La boule tirée au $i^{\text{ième}}$ tirage est blanche."

On cherche à déterminer la probabilité $P(B_1 \cap B_2 \cap \dots \cap B_n)$.

On utilise la formule des probabilités composées :

$$P(B_1 \cap B_2 \cap \dots \cap B_n) = P(B_1) \times P_{B_1}(B_2) \times P_{B_1 \cap B_2}(B_3) \times \dots \times P_{B_1 \cap B_2 \cap \dots \cap B_{n-1}}(B_n)$$

Or, par définition de l'expérience, pour tout entier naturel k tel que $2 \leq k \leq n$, on a :

$$P_{B_1 \cap B_2 \cap \dots \cap B_{k-1}}(B_k) = \frac{k}{k+1} \text{ et } P(B_1) = \frac{1}{2}$$

En effet, avant le $k^{\text{ième}}$ tirage on a effectué $k-1$ tirages qui ont tous donné une boule blanche, on a ainsi ajouté $k-1$ boules blanches avant le $k^{\text{ième}}$ tirage. Lors du $k^{\text{ième}}$ tirage on se retrouve donc avec une urne composée de $1 + (k-1) = k$ boules blanches et d'une seule boule noire (donc $k+1$ boules au total).

Ainsi

$$P(B_1 \cap B_2 \cap \dots \cap B_n) = \frac{1}{2} \times \frac{2}{3} \times \frac{3}{4} \times \dots \times \frac{n}{n+1} = \frac{1}{n+1}.$$

Corrigé des questions 1 et 2 de l'exercice 2

$$1. Y_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \text{ et } Y_1 = \begin{pmatrix} 0 \\ \frac{1}{2} \\ \frac{1}{2} \end{pmatrix}.$$

2. (a) (A_n, B_n, C_n) forme un système complet d'événements. Donc d'après la formule des probabilités totales :

$$P(A_{n+1}) = P(A_n) \times P_{A_n}(A_{n+1}) + P(B_n) \times P_{B_n}(A_{n+1}) + P(C_n) \times P_{C_n}(A_{n+1})$$

On obtient ainsi :

$$P(A_{n+1}) = \frac{1}{2} \times P(B_n) + \frac{1}{4} \times P(C_n)$$

De même on obtient :

$$P(B_{n+1}) = \frac{1}{2} \times P(A_n) + \frac{1}{4} \times P(C_n)$$

et

$$P(C_{n+1}) = \frac{1}{2} \times P(A_n) + \frac{1}{2} \times P(B_n) + \frac{1}{2} \times P(C_n)$$

Par définition de Y_n on obtient bien $Y_{n+1} = AY_n$

- (b) Démontrons l'égalité par récurrence sur l'entier n :

Initialisation : On a bien $Y_0 = A^0 Y_0$ puisque $A^0 = I_3$.

Hérédité : Soit $n \in \mathbb{N}$. Supposons que pour cet entier n on ait $Y_n = A^n Y_0$. Sous cette hypothèse, montrons que $Y_{n+1} = A^{n+1} Y_0$.

$$Y_{n+1} = AY_n = A \times A^n Y_0 = A^{n+1} Y_0$$

L'hérédité est vérifiée.

Par le principe de récurrence, $\forall n \in \mathbb{N}^*, Y_n = A^n Y_0$.