Devoir libre n°04

2TSI. Mathématiques

À rendre le jeudi 23 novembre 2017 au plus tard

Les deux exercices sont indépendants et peuvent être faits dans n'importe quel ordre.

Exercice 01

Inspiré d'une planche d'oral du Concours Centrale Supélec Mathématiques II en 2016.

Dans 1), 2), 3) et 4)d), on écrit des programmes en Python.

Une particule P se déplace sur une surface comportant quatre positions successives, A_0 (position 0) qui est un puits, A_1 (position 1) et A_2 (position 2) deux positions intermédiaires et A_3, (position 3) un second puits. À l'instant $t = n$:
• si P est dans un puits, elle y reste avec une probabilité de 1.
• si P est en A_1, P va en A_0 avec la probabilité p, en A_2 avec la probabilité $1 - p$.
• si P est en A_2, P va en A_1 avec la probabilité p, en A_3 avec la probabilité $1 - p$.
On note x_n la position de la particule P à $t = n$, $x_n(\Omega) = [\![0, 3]\!]$.

1. Écrire une fonction $Position(x, p)$ en Python qui donne x_{n+1} en fonction de x_n.
   Indication : pour $Position(x, p)$, on remarque que $x = 1$ a pour image 0 (probabilité p) et 2 (probabilité $q = 1 - p$). On peut introduire $va = binom(1, p)$ et $X = va.rvs(size = 1)$. Ainsi si $X[0] = 1$, on tape $return(0)$ et sinon $return(2)$. Idem pour $x = 2$.
2. Écrire une fonction $Pos(n, p, x0)$ en Python renvoyant x_n en fonction de x_0 et p.
3. Écrire une fonction en Python qui donne avec quelle fréquence la particule se trouve dans les quatre positions au bout de N essais, en fonction de la position initiale x_0. On suppose ici $p = \dfrac{1}{2}$.
   Indication : on pourra utiliser $L.count(i)$ qui donne le nombre d'occurences de i dans la liste L.
4. Soit, pour tout $n \in \mathbb{N}$,
$$X_n = \begin{pmatrix} P(x_n = 0) \\ P(x_n = 1) \\ P(x_n = 2) \\ P(x_n = 3) \end{pmatrix}.$$
   (a) Trouver $A \in \mathcal{M}_4(\mathbb{R})$, indépendante de n, telle que $X_{n+1} = AX_n$.
   Indication : on pourra poser $A_{i,n}$ l'événement : « être en position i à l'instant n », et utiliser la formule des probabilités totales.

   On suppose dorénavant dans toute la suite $p = \dfrac{1}{2}$.

   (b) Montrer que A est diagonalisable dans $\mathcal{M}_4(\mathbb{R})$ et la réduire. On prendra une base de vecteurs propres associés à des valeurs propres croissantes.
   (c) En utilisant la question précédente, calculer A^n pour tout $n \in \mathbb{N}$ et en déduire $\lim\limits_{n \to +\infty} X_n$.
   (d) Écrire un programme Python d'argument l'entier n et qui fournit A^n.
   Indication :ici, la question est un peu ouverte, commencez par aller voir dans le Whoswho les deux méthodes pour écrire une matrice, calculer ses puissances ou la diagonaliser. Le module usuel est numpy mais en utilisant sympy avec l'ordinateur, ce sera plus ergonomique. Donc vous pouvez faire simplement un programme qui donne la puissance $n^{\text{ème}}$, ce qui est quasiment immédiat ou alors diagonaliser, inverser la matrice de passage et refaire la question **4)c)**. *Vous pouvez aussi visualiser $\lim\limits_{n \to +\infty} X_n$. Bref, régalez vous !*

Exercice 02

Inspiré d'un exercice posé aux Concours Commun Mines-Ponts en 2016.

Un centre téléphonique a une liste de n contacts, chacun ayant une probabilité p d'être appelé et de répondre à l'appel. On note X et Y les variables aléatoires représentant le nombre de personnes répondant à l'appel au premier et au second tour respectivement. À chaque tour, on appelle les n contacts qui ne répondent donc pas forcément. Si une personne répond à un tour, on ne la contacte plus.

1. Déterminer la loi de X.
2. Déterminer, pour tout $(i, k) \in \mathbb{N}^2$,

$$P_{X=i}(Y = k).$$

   En déduire la loi de Y.
3. Déterminer la loi de $X + Y$ et déterminer $E(X + Y)$ de deux manières.

   Indication : pour déterminer la loi de Y et la loi de $X + Y$, on écrira $P(Y = k)$ et $P(X + Y = j)$ sous forme de somme et on fera apparaître la formule du binôme de Newton.