Exercice I

nb : les questions étant déciminées le long du sujet, elles n'ont pas ou peu de rapport entre elles.

1. On note $\mathscr{H}_n$ l'ensemble des matrices M dans $\mathscr{M}_n(\mathbb{R})$ telles que
- tous les coefficients de M sont dans $\{-1;1\}$
- les vecteurs colonnes de la matrice M sont orthogonaux 2 à 2 pour le produit scalaire $<X;Y>=\sum x_i y_i$

Ainsi on peut constater que $\begin{pmatrix}1 & 1\\ 1 & -1\end{pmatrix} \in \mathscr{H}_2$ et $\begin{pmatrix}1 & 1 & -1 & 1\\ -1 & -1 & -1 & 1\\ 1 & -1 & -1 & -1\\ -1 & 1 & -1 & -1\end{pmatrix} \in \mathscr{H}_4$

Écrire une fonction en `python` qui, lorsqu'elle prend en entrée la liste des colonnes d'une matrice M, de taille n, renvoie 1 si la matrice est dans $\mathscr{H}_n$ et 0 sinon.

2. On note U_n le nombre de valeurs distinctes prises par les variables aléatoires $X_1, \dots X_n$ toutes indépendantes et de même loi uniforme sur $[\![0;l]\!]$. Si les k_i sont les valeurs prises par les V.A. X_i alors U_n prend la valeur $|S|$ où $S = \{k1; \dots k_n\}$

On se propose de simuler en `python` la V.A. U_n pour $n = 10$ et $l = 25$.

(a) Écrire une fonction `simulU` qui renvoie une réalisation de U_{10}. On pourra utiliser la fonction `random.randint` où l'instruction `random.randint(1,25)` fournit un nombre aléatoire dans $[\![1;25]\!]$ uniformément

(b) Écrire une fonction `espU` qui renvoie une approximation de l'espérance de U_{10}. Quel théorème utilisez-vous pour justifier que le résultat de cette fonction est une approximation de $E[U_{10}]$?

Exercice II

Si A est une matrice stochastique strictement positive, on a établi dans la partie précédente la convergence de la suite $(\mu_n)_{n\in\mathbb{N}}$ associée à la matrice A. Ceci fournit un algorithme de calcul de la probabilité invariante par A. On en propose une implémentation en langage Python. On sera très attentif à la rédaction et notamment à l'indentation du code.

Un vecteur x de $\mathbb{R}^p$ sera représenté en Python par une liste de flottants. Par exemple, le vecteur $x = (1,2,3)$ de $\mathbb{R}^3$ sera représenté par la liste `[1,2,3]`. De même, une matrice A sera représentée par une liste dont les éléments sont les lignes de la matrice. Par exemple, la matrice $A = \begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\end{pmatrix}$ sera représentée par la liste `[[1,2,3],[4,5,6]]`.

1. Définir sous python la matrice $A = \begin{pmatrix}1 & 2 & 3\\ 4 & 5 & 6\\ 7 & 8 & 9\\ 10 & 11 & 12\end{pmatrix}$.
2. Donner les valeurs renvoyées lorsque l'on exécute `len(A)`, `A[1]` et `A[2][1]`.
3. Écrire une fonction `difference` qui prend en arguments deux vecteurs x et y de même taille et renvoie le vecteur $x - y$. Par exemple si $x = (5,2)$ et $y = (3,7)$, `difference(x,y)` renverra `[2,-5]`.
4. Écrire une fonction `norme` qui prend en argument un vecteur $x = (x_1, \dots, x_p)$ et renvoie sa norme infinie $\|x\|_\infty = \max\{|x_i| \ tq \ \ i \in [\![1;p]\!]\}$ (on pourra utiliser librement la fonction `abs` qui renvoie la valeur absolue d'un nombre, mais on s'interdit l'utilisation de la fonction `max` déjà implémentée dans Python).
5. Écrire une fonction `itere` qui prend en arguments un vecteur ligne x et une matrice carrée de même taille que x et qui renvoie le vecteur xA. Par exemple si $x = (1,1)$ et $A = \begin{pmatrix}1 & 2\\ 4 & 5\end{pmatrix}$, on a $xA = (5,7)$ et donc `itere(x,A)` renverra `[5,7]`.
6. On admet que si A est une matrice strictement positive, la suite de vecteurs lignes de $\mathbb{R}^p$ associée $(\mu_n)_{n\in\mathbb{N}}$ définie par la relation : $\forall n \in \mathbb{N},\ \mu_{n+1} = \mu_n A$ converge vers un vecteur μ_∞ indépendant du choix de μ_0 vecteur stochastique.

Écrire une fonction `probaInvariante` qui prend en arguments une matrice stochastique strictement positive A de $\mathscr{M}_p(\mathbb{R})$ et un réel $\varepsilon > 0$ et qui renvoie le premier terme μ_k de la suite $(\mu_n)_{n\in\mathbb{N}}$ avec $\mu_0 = \left(\frac{1}{p}, \frac{1}{p}, \dots, \frac{1}{p}\right)$ tel que $\|\mu_k - \mu_{k-1}\|_\infty \leqslant \varepsilon$. On ne demandera pas à l'algorithme de vérifier que la matrice passée en argument est bien stochastique et strictement positive.

Par exemple, si $A = \begin{pmatrix}\frac{1}{2} & \frac{1}{2}\\ \frac{1}{4} & \frac{3}{4}\end{pmatrix}$ et $\varepsilon = 10^{-6}$, `probaInvariante(A,eps)` renverra `[0.3333339691, 0.666666030]`.

Exercice III

nb : les questions étant déciminées le long du sujet, elles n'ont pas ou peu de rapport entre elles.

On a préalablement montrer que $\left|\int_0^1 e^{-x^2}dx - \sum_{k=0}^{n} \frac{(-1)^k}{(2k+1)k!}\right| = |I - S_n| \leqslant \frac{1}{(2n+3)(n+1)!}$

1. Écrire une fonction récursive `factorielle` qui prend en argument un entier naturel n et renvoie l'entier $n!$
2. en déduire un script qui détermine un entier N, tel que $|I - S_N| \leqslant 10^{-6}$

Un peu plus loin, pour tout entier $i \in [\![0; n]\!]$ on définit le polynôme $l_i(X) = \prod_{\substack{k=0 \\ k \neq i}}^{n} \frac{X - x_k}{x_i - x_k} \in \mathbb{R}_n[X]$.

Si $y_0, \dots y_n$ sont des réels, le polynôme $P = \sum_{i=0}^{n} y_i l_i(X)$ est l'unique polynôme de $\mathbb{R}_n[X]$ vérifiant $\forall i,\ P(x_i) = y_i$

3. Écrire en langage `python`, une fonction `lagrange` qui prend en arguments x une liste de points d'interpolations x_i, y une liste d'ordonnées y_i de même longueur, a un réel et qui renvoie la valeur de P en a.
Par exemple, si `x=[-1,0,1]` et `y=[4,0,4]`, on montre que $P = 4X^2$ et ainsi $P(3) = 36$. Aussi `lagrange(x,y,3)` renverra `36`.