COLLE SEMAINE 21 - SUJET 1 -

Exercice 1 : Correction

① a) *La matrice A est-elle diagonalisable ?* La matrice est triangulaire donc

$$\mathrm{Sp}(A) = \{1, \tfrac{1}{2}, \cdots, \tfrac{1}{n+1}\}$$

A est une matrice d'ordre n qui admet n valeur propres distinctes donc elle est diagonalisable.

b) *Déduire du résultat précédent que la suite $(A^k)_{k\in\mathbb{N}}$ converge :* On écrit que A est diagonalisable, donc il existe une matrice de passage P, inversible, telle que : $A = PDP^{-1}$ où

$$D = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1/2 & 0 & \cdots & 0 \\ 0 & 0 & 1/3 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 1/(n+1) \end{pmatrix}.$$

Une récurrence (à écrire) prouve que : $\forall n \in \mathbb{N}$, $A^n = PD^nP^{-1}$ avec

$$\lim_{n\to\infty} D^n = \begin{pmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & \cdots & 0 \\ 0 & 0 & 0 & \cdots & 0 \end{pmatrix} \text{ car } \lim_{n\to\infty} q^n = 0 \text{ si } 0 < q < 1$$

c) *En utilisant la fonction **`linalg.eig`** de **`numpy`**, déterminons le vecteur propre associé à la valeur propre 1 de A :* On peut par exemple écrire :

```
def matriceA(n):
    A = np.zeros((n+1,n+1))
    for i in range(n+1):
       for j in range(i,n+1):
          A[i,j]=1/(j+1)
    return np.matrix(A)

n = 10
A = matriceA(n)
vp,P = np.linalg.eig(A)
X1 = P[:,0]
```

On obtient que $X1$ est la matrice colonne représentant le premier vecteur de la base canonique. Ce qui était évident à la lecture de A car A ayant $n+1$ valeurs propres distinctes, la dimension de chaque valeur propre vaut 1. En particulier $\dim(E_1) = 1$ et :

$$A \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix}$$

Soit $\boxed{E_1 = \mathrm{Vect}\{(1, 0, \cdots, 0)\}}$

② *Écrivons une fonction qui prend en paramètres deux entiers k et n et renvoie une liste contenant le résultat de k tirages :*

```
def simulTirages(n,k):
    L = [n]
    u = n
    for i in range(k):
       t = rdm.randint(0,u)
       L.append(t)
       u = t
    return L
```

③ a) *Pour tout $j \in [\![0, n]\!]$, écrivons $\mathbb{P}(X_{k+1} = j)$ en fonction de certains des nombres $\mathbb{P}(X_k = i)$ pour i dans $[\![0, n]\!]$:* On considère le système complet d'événements : $\{(X_k = i), i \in [\![0, n]\!]\}$. Alors, d'après la Formule des probabilités totales :

$$\mathbb{P}(X_{k+1} = j) = \sum_{i=0}^{n} \mathbb{P}_{(X_k=i)}(X_{k+1} = j)\mathbb{P}(X_k = i) = \sum_{i=j}^{n} \frac{1}{i+1}\mathbb{P}(X_k = i)$$

b) *En déduire la relation $W_{k+1} = AW_k$ puis une expression de W_k en fonction de A et de W_0 :* La relation matricielle demandée découle directement de la définition du produit matriciel et, par récurrence :

$$\forall k \in \mathbb{N},\ W_k = A^k W_0 \text{ avec } W_0 = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

c) *Écrivons une fonction en Python qui prend en paramètres deux entiers k et n et engendre le vecteur W_0 et renvoie le vecteur W_k (en utilisant* **`matriceA(n)`***) :*

```
def vecteurW(n,k):
    A = matriceA(n)
    W = np.matrix(np.zeros(n+1)).T # Transposition
    W[n]=1
    for i in range(k):
       W = A*W
    return W

n3 = 100
k3 = 20
W = vecteurW(n3,k3)
```

④ a) *Déterminons la matrice $B \in \mathcal{M}_{1,n+1}(\mathbb{R})$ telle que $BW_k = \mathbb{E}(X_k)$:* Il suffit de poser

$$B = \begin{pmatrix} 0 & 1 & 2 & \cdots & n \end{pmatrix}$$

b) *Calculons le produit BA en fonction de B :*

$$BA = \begin{pmatrix} 0 & 1 & 2 & \cdots & n \end{pmatrix} \cdot \begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ 0 & \frac{1}{2} & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & 0 & \frac{1}{3} & \cdots & \frac{1}{n+1} \\ \vdots & \vdots & \ddots & \ddots & \vdots \\ 0 & 0 & \cdots & 0 & \frac{1}{n+1} \end{pmatrix} = \begin{pmatrix} 0 & \frac{1}{2} & \frac{2}{2} & \cdots & \frac{n}{2} \end{pmatrix} = \frac{1}{2}B$$

c) *Pour tout entier naturel k, exprimons $\mathbb{E}(X_{k+1})$ en fonction de $\mathbb{E}(X_k)$:*

$$\mathbb{E}(X_{k+1}) = BW_{k+1} = BAW_k = \frac{1}{2}BW_k = \frac{1}{2}\mathbb{E}(X_k)$$

d) On reconnaît une suite géométrique de raison 1/2.

Conclusion : $\boxed{\mathbb{E}(X_k) = \left(\frac{1}{2}\right)^k \mathbb{E}(X_0) = \frac{n}{2^k}}$

Vérifions son adéquation avec les résultats empiriques obtenus précédemment :
J'ai écrit pour ça la fonction suivante :

```
def confrontation(n,kmax,m):
    L = np.zeros(kmax+1)
    for i in range(m):
       L += np.array(simulTirages(n,kmax))
    E1 = L/m # moyenne de chacun des Xk pour k = 0...kmax
    X = np.arange(kmax+1)
    plt.plot(X,E1,'ro')
    plt.plot(X,[n/2**k for k in range(kmax+1)],'b')
    plt.show()
    return E1
```

En tapant : `E1 = confrontation(100,20,1000)`, on obtient :