Planches HEC

Sujet E 42

Exercice avec préparation 1

Dans cet exercice, n désigne un entier supérieur ou égal à 1.

1. **Question de cours** :
Que peut-on dire du degré de la somme et du produit de deux polynômes ?

Soit $\mathbb{R}_n[X]$ l'espace vectoriel des polynômes à coefficients réels de degré inférieur ou égal à n, et f l'application qui à tout polynôme $P \in \mathbb{R}_n[X]$, fait correspondre le polynôme $f(P)$ défini par :

$$f(P)(X) \;=\; n\,X\,P(X) + X(1-X)P'(X), \text{ où } P' \text{ désigne la dérivée du polynôme } P$$

2. ***a)*** Montrer que f est un endomorphisme de $\mathbb{R}_n[X]$ et écrire sa matrice M dans la base canonique de $\mathbb{R}_n[X]$.

b) L'endomorphisme f est-il bijectif ? Quel est son rang ?

c) La matrice M est-elle diagonalisable ?

3. Pour tout $k \in [\![0, n]\!]$, on note H_k le polynôme de $\mathbb{R}_n[X]$ défini par : $H_k(X) = X^k(1-X)^{n-k}$.

a) Calculer pour tout $k \in [\![0, n]\!]$, $f(H_k)(X)$. Donner une traduction matricielle de ce résultat.

b) En déduire que $\mathscr{B} = (H_0, H_1, \ldots, H_n)$ est une base de $\mathbb{R}_n[X]$.

c) Trouver les coordonnées du polynôme $(X+1)^n$ dans la base $\mathscr{B}$.

Exercice sans préparation 1

On considère une urne contenant b boules blanches et r boules rouges.

1. La fonction **Python** suivante permet de simuler des tirages dans cette urne.

```python
def X(b, r):
    V = False # Le booléen "faux"
    for k in range(3):
        V = V or (rd.randint(1, b + r + 1) <= b)
    if V:
        return 1
    else:
        return 2
```

Que retourne la fonction X et quelle loi simule-t-elle ?

2. De quelle valeur théorique la valeur affichée après l'exécution des instructions suivantes fournit-elle une approximation ?

```
import numpy as np
R = []
for k in range(10000):
    R.append(X(5, 5))
print(np.mean(R))
```

Réponses de l'exercice avec préparation 1 :

1. $\deg(P+Q) \leqslant \max(\deg(P), \deg(Q))$ et $\deg(PQ) = \deg(P) + \deg(Q)$.

2. ***a)*** f est linéaire et

- $f(1) = nX$
- pour tout $k \in [\![1, n]\!]$, $f(X^k) = kX^k + (n-k)X^{k+1}$. En particulier, $f(X^n) = nX^n$.

Donc
$$\mathrm{Im}(f) = \mathrm{Vect}\left(f(1), \dots, f(X^n)\right) \subset \mathbb{R}_n[X]$$
et f est bien un endomorphisme de $\mathbb{R}_n[X]$.

$$M = \begin{pmatrix} 0 & \cdots & \cdots & \cdots & 0 \\ n & 1 & & (0) & \\ & n-1 & \ddots & & \\ & & \ddots & \ddots & \\ & (0) & & 1 & n \end{pmatrix}$$

b) La matrice M est triangulaire inférieure donc ses coefficients diagonaux sont exactement ses valeurs propres. On en déduit que $\mathrm{Sp}(M) = \{0, 1 \dots, n\}$. Ainsi, M possède $n+1$ valeurs propres distinctes et on en déduit que :

- M est diagonalisable
- pour tout $k \in \{0, 1 \dots, n\}$, $\dim(E_k(M)) = 1$

En particulier, f n'est pas bijectif (0 est valeur propre de M donc M n'est pas inversible) et $\dim(\ker(f)) = 1$. Par théorème du rang : $n+1 = 1 + \mathrm{rg}(f)$ et donc $\mathrm{rg}(f) = n$.

Commentaire

On pouvait aussi faire un calcul de rang directement sur la matrice M.
La matrice M comporte une ligne de 0 donc n'est pas inversible, donc f n'est pas bijective.

$$\mathrm{rg}(M) = \mathrm{rg}\left(\begin{pmatrix} n & 1 & & (0) & \\ & n-1 & \ddots & & \\ & & \ddots & \ddots & \\ & (0) & & 1 & n \end{pmatrix}\right)$$

Les $n+1$ colonnes de cette matrice forment une famille liée de $\mathscr{M}_{n,1}(\mathbb{R})$ par dimension, or les n premières colonnes forment une famille libre car étagée. On en déduit que la dernière colonne est une combinaison linéaire des n premières. Ainsi,

$$\mathrm{rg}(M) = \mathrm{rg}\left(\begin{pmatrix} n & 1 & & (0) \\ & n-1 & \ddots & \\ & & \ddots & n-1 \\ & (0) & & 1 \end{pmatrix}\right)$$

Cette dernière matrice est carrée d'ordre n, est triangulaire supérieure et tous ses coefficients diagonaux sont non nuls donc elle est inversible. On en déduit que $\mathrm{rg}(f) = \mathrm{rg}(M) = n$.
On peut aussi tout de suite remarquer que les n lignes sont échelonnées donc forment une famille libre et donc $\mathrm{rg}(M) = n$.

c) cf question précédente

3. a) Pour tout $k \in [\![0, n]\!]$, $f(H_k)(X) = kH_k(X)$. Notons $V_k = \mathrm{Mat}_{\mathrm{bc}}(H_k(X))$ la matrice représentative du polynôme $H_k(X)$ dans la base canonique de $\mathbb{R}_n[X]$. L'égalité précédente se réécrit :

$$\forall k \in [\![0, n]\!],\ MV_k = kV_k$$

b) La famille $(V_0, V_1, \dots, V_n)$ est une famille constituée de $n+1$ vecteurs propres de M associés à $n+1$ valeurs propres distinctes donc est libre. Par isomorphisme de représentation matricielle, la famille $\mathscr{B} = (H_0, H_1, \dots, H_n)$ est libre et donc est une base de $\mathbb{R}_n[X]$ par argument de dimension.

c) On a $(X+1)^n = ((1-X)+2X)^n = \sum\limits_{k=0}^{n} \binom{n}{k}(2X)^k(1-X)^{n-k} = \sum\limits_{k=0}^{n} 2^k \binom{n}{k} H_k(X)$.

Réponses de l'exercice sans préparation 1 :

1. La fonction **Python** renvoie une simulation de la variable aléatoire $X_{b,r}$ définie de la manière suivante :

- On effectue 3 tirages avec remise dans une urne contenant b boules blanches et r boules rouges.
- Si au cours des 3 tirages, on obtient au moins une fois une boule blanche, alors $X_{b,r}$ prend la valeur 1, sinon $X_{b,r}$ prend la valeur 2.

$\mathbb{P}([X_{b,r} = 1]) = 1 - \left(\dfrac{r}{b+r}\right)^3$ et $\mathbb{P}([X_{b,r} = 2]) = \left(\dfrac{r}{b+r}\right)^3$

2. Le script renvoie une approximation de $\mathbb{E}(X_{5,5})$ d'après la loi faible des grands nombres (cette variable aléatoire est finie donc admet une espérance et une variance)
Pour calculer cette espérance, on peut écrire $X_{5,5} = 1 + Y$ où $Y \hookrightarrow \mathcal{B}\left(\frac{1}{8}\right)$. Ainsi,

$$\mathbb{E}(X_{5,5}) = 1 + \frac{1}{8} = \frac{9}{8}$$

Sujet E 82

Exercice avec préparation 2

On suppose que toutes les variables aléatoires qui interviennent dans l'exercice sont définies sur un même espace probabilisé $(\Omega, \mathscr{A}, \mathbb{P})$.

1. Question de cours : Loi uniforme sur un intervalle $[a, b]$; définition, propriétés.

2. Pour tout x réel, on note $\lfloor x \rfloor$ la partie entière de x.

a) Pour n entier de $\mathbb{N}^*$, montrer que pour tout x réel, on a : $\lim\limits_{n\to+\infty} \dfrac{\lfloor nx \rfloor}{n} = x$.

b) Établir pour tout $(x, y) \in \mathbb{R}^2$ l'équivalence suivante : $\lfloor y \rfloor \leqslant x \iff y < \lfloor x \rfloor + 1$.

c) Soit α et β deux réels vérifiant $0 \leqslant \alpha \leqslant \beta \leqslant 1$ et soit $N_n(\alpha, \beta)$ le nombre d'entiers k qui vérifient $\alpha < \dfrac{k}{n} \leqslant \beta$. Exprimer $N_n(\alpha, \beta)$ en fonction de $\lfloor n\alpha \rfloor$ et $\lfloor n\beta \rfloor$.

3. Pour tout entier $n \geqslant 1$, on note Y_n la variable aléatoire discrète dont la loi est donnée par :

$$\forall k \in [\![0, n-1]\!],\ \mathbb{P}\left(\left[Y_n = \frac{k}{n}\right]\right) = \frac{1}{n}.$$

Soit Z une variable aléatoire suivant la loi uniforme sur l'intervalle $[0, 1]$. Pour tout entier $n \geqslant 1$, on définit la variable aléatoire Z_n par : $Z_n = \dfrac{\lfloor nZ \rfloor}{n}$. Soit α et β deux réels vérifiant $0 \leqslant \alpha \leqslant \beta \leqslant 1$.

a) Écrire une fonction **Python** `partieEntiere(x)` prenant en entrée un réel x positif et renvoyant sa partie entière inférieure. On n'utilisera pas la fonction `np.floor`.

b) Écrire une fonction **Python** `simulY(n)` (resp. `simulZ(n)`) prenant en entrée un entier $n \geqslant 1$ et simulant la variable aléatoire Y_n (resp. Z_n).

c) Montrer que $\lim\limits_{n\to+\infty} \mathbb{P}([\alpha < Y_n \leqslant \beta]) = \beta - \alpha$.

d) Comparer les fonctions de répartition respectives de Y_n et Z_n. Conclusion.

Exercice sans préparation 2

Soit x réel et $M(x)$ la matrice de $\mathscr{M}_2(\mathbb{R})$ définie par : $M(x) = \begin{pmatrix} x & -1 \\ 2x & 2x \end{pmatrix}$.

Pour quelles valeurs de x la matrice $M(x)$ est-elle diagonalisable ?

Réponses de l'exercice avec préparation 2 :

1. $X \hookrightarrow \mathcal{U}([a,b])$ si la fonction

$$f : t \mapsto \begin{cases} 0 & \text{si } t < a \\ \frac{1}{b-a} & \text{si } a \leqslant t \leqslant b \\ 0 & \text{si } t > b \end{cases}$$

est une densité de X. Sa fonction de répartition est dans ce cas :

$$F_X : t \mapsto \begin{cases} 0 & \text{si } t < a \\ \frac{t-a}{b-a} & \text{si } a \leqslant t \leqslant b \\ 1 & \text{si } t > b \end{cases}$$

et

$$\mathbb{E}(X) = \frac{a+b}{2}, \qquad \mathbb{V}(X) = \frac{(b-a)^2}{12}$$

2. Pour tout x réel, on note $\lfloor x \rfloor$ la partie entière de x.

a) Soit $x \in \mathbb{R}$. Pour tout entier n non nul, on a : $\lfloor nx \rfloor \leqslant nx < \lfloor nx \rfloor + 1$ donc $nx - 1 < \lfloor nx \rfloor \leqslant nx$ donc $x - \frac{1}{n} < \frac{\lfloor nx \rfloor}{n} \leqslant x$. Par théorème d'encadrement :

$$\frac{\lfloor nx \rfloor}{n} \underset{n \to +\infty}{\longrightarrow} x$$

b) Si $\lfloor y \rfloor \leqslant x$, alors $\lfloor y \rfloor \leqslant \lfloor x \rfloor$ (par argument de maximalité de la partie entière) et donc $y < \lfloor y \rfloor + 1 \leqslant \lfloor x \rfloor + 1$.
Si $y < \lfloor x \rfloor + 1$, alors $\lfloor y \rfloor < \lfloor x \rfloor + 1$ (car $\lfloor y \rfloor \leqslant y$) et donc $\lfloor y \rfloor \leqslant \lfloor x \rfloor \leqslant x$.

c) On remarque que, pour tout $x > 0$, le nombre d'entiers k vérifiant $0 < k \leqslant x$ est $\lfloor x \rfloor$.
Or,

$$\begin{aligned} \{k \in \mathbb{N} \mid \alpha < \frac{k}{n} \leqslant \beta\} &= \{k \in \mathbb{N} \mid n\alpha < k \leqslant n\beta\} \\ &= \{k \in \mathbb{N} \mid 0 < k \leqslant n\beta\} \setminus \{k \in \mathbb{N} \mid 0 < k \leqslant n\alpha\} \end{aligned}$$

En passant aux cardinaux, on en déduit que $N_n(\alpha, \beta) = \lfloor n\beta \rfloor - \lfloor n\alpha \rfloor$.

3. ***a)*** On propose la fonction **Python** suivante, utilisant la propriété : $\lfloor x \rfloor \leqslant x < \lfloor x \rfloor + 1$.

```python
def partieEntiere(x):
    k = 0
    while k+1 <= x:
        k = k+1
    return k
```

b) On propose les fonctions **Python** suivantes :

```python
def simulY(n):
    k = rd.randint(0,n)
    Y = k/n
    return Y
```

```python
def simulZ(n):
    U = rd.random()
    Z = partieEntiere(n*U)/n
    return Z
```

c) On remarque que

$$[\alpha < Y_n \leqslant \beta] = \bigcup_{n\alpha < k \leqslant n\beta} \left[Y_n = \frac{k}{n} \right]$$

d'où

$$\mathbb{P}([\alpha < Y_n \leqslant \beta]) = \sum_{n\alpha < k \leqslant n\beta} \mathbb{P}\left(\left[Y_n = \frac{k}{n}\right]\right) = \sum_{n\alpha < k \leqslant n\beta} \frac{1}{n} = \frac{N(\alpha,\beta)}{n} = \frac{\lfloor n\beta \rfloor - \lfloor n\alpha \rfloor}{n} \underset{n\to+\infty}{\longrightarrow} \beta - \alpha$$

d) D'une part :

$$F_{Y_n}(x) = \begin{cases} 0 & \text{si } x < 0 \\ \frac{\lfloor nx \rfloor}{n} + \frac{1}{n} & \text{si } 0 \leqslant x < 1 \\ 1 & \text{si } x \geqslant 1 \end{cases}$$

D'autre part,

$$\begin{aligned} [Z_n \leqslant x] &= \left[\frac{\lfloor nZ \rfloor}{n} \leqslant x\right] = [\lfloor nZ \rfloor \leqslant nx] \\ &= [nZ \leqslant \lfloor nx \rfloor + 1] = \left[Z \leqslant \frac{\lfloor nx \rfloor}{n} + \frac{1}{n}\right] \end{aligned}$$

Or, $\frac{\lfloor nx \rfloor}{n} + \frac{1}{n} \in [0,1]$.
On en déduit que $F_{Y_n} = F_{Z_n}$ donc Y_n et Z_n ont même loi.

Réponses de l'exercice sans préparation 2 :
Soit $x \in \mathbb{R}$. Soit $\lambda \in \mathbb{R}$.

$$\begin{aligned} \lambda \text{ est valeur propre de } M(x) &\iff \det(M(x) - \lambda I_2) = 0 \\ &\iff (x-\lambda)(2x-\lambda) + 2x = 0 \\ &\iff \lambda^2 - 3x\lambda + 2x(x+1) \end{aligned}$$

On note $\Delta(x) = (-3x)^2 - 8x(x+1) = x(x-8)$. Il y a alors trois cas à distinguer :

- Si $x \in]0,8[$, alors $\Delta(x) < 0$. La matrice $M(x)$ ne possède pas de valeurs propres et donc n'est pas diagonalisable.
- Si $x = 0$ ou $x = 8$, alors $\Delta(x) = 0$. La matrice $M(x)$ possède alors une unique valeur propre et donc n'est pas diagonalisable car n'est pas un multiple de l'identité (raisonnement par l'absurde classique).
- Si $x < 0$ ou $x > 8$, alors $\Delta(x) > 0$. La matrice $M(x)$ possède alors deux valeurs propres distinctes et donc est diagonalisable.

Sujet Maths appliquées 8

Exercice avec préparation 3

On tire une main de 5 cartes d'un paquet de 32 cartes.
On considère alors les événements A et B suivants :

$$A : \text{« L'as de pique est dans la main » et } B : \text{« Au moins un as est dans la main »}$$

1. Cours : coefficients binomiaux ; interprétation ensembliste.

2. ***a)*** Calculer la probabilité $\mathbb{P}(A)$.

b) Sachant que $\dfrac{27 \times 26 \times 25 \times 24}{32 \times 31 \times 30 \times 29} < \dfrac{1}{2}$, montrer que : $\mathbb{P}(B) > 2\mathbb{P}(A)$.

On note maintenant :

- X la variable aléatoire égale au nombre d'as d'une main de cinq cartes,
- Y la variable aléatoire égale au nombre d'as d'une main contenant l'as de pique,
- Z la variable aléatoire égale au nombre d'as d'une main contenant au moins un as.

On se propose de comparer les espérances $\mathbb{E}(Y)$ et $\mathbb{E}(Z)$.

3. Etude empirique.

On suppose qu'on dispose d'une fonction **Python** `ma_main()` qui ne prend aucun argument et qui renvoie une main déterminée de manière aléatoire. Une telle main sera représentée par un tableau de 5 couples `(val, coul)` où `val` est élément de la liste `[7, 8, 9, 10, 'valet', 'dame', 'roi', 'as']` et `coul` de la liste `['trefle', 'carreau', 'coeur', 'pique']`.

a) Ecrire le code d'une fonction **Python** `as_de_pique(m)` qui prend en argument une main `m` de cinq cartes, et qui renvoie `True` si la main contient l'as de pique, `False` sinon.

b) Ecrire le code d'une fonction **Python** `nbre_as(m)` qui prend en argument une main `m` de cinq cartes et qui renvoie le nombre d'as contenus dans la main.

c) Compléter la première boucle `for` dans le code de la fonction `estimation(N)` suivante, qui prend en entrée un entier `N` et réalise une estimation des espérances $\mathbb{E}(Y)$ et $\mathbb{E}(Z)$.

```
import numpy as np

def estimation(N)
    a,b = 0,0
    xA,xB = np.zeros(5),np.zeros(5)
    for i in range(N):
        main = ma_main()
        ........................
    sY = 0; sZ = 0
    for k in range(1,5):
        sY += k*xA[k]
        sZ += k*xB[k]
    if a*b != 0:
        return sY/a, sZ/b
```

La première boucle `for` comptabilise, pour `N` mains aléatoires : dans `a` (resp. `b`) le nombre de celles qui contiennent l'as de pique (resp. au moins un as) et au rang n $(0 \leqslant n \leqslant 4)$ de la liste `xA` (resp. `xB`) le nombre de celles contenant n as et appartenant à A (resp. à B).

4. Etude théorique.

a) Pour $k = 2$, $k = 3$ et $k = 4$, comparer les probabilités conditionnelles $\mathbb{P}_A(X = k)$ et $\mathbb{P}_B(X = k)$.

b) Comparer les espérance $\mathbb{E}(Y)$ et $\mathbb{E}(Z)$.

Exercice sans préparation 3
Etudier le comportement asymptotique, quand t tend vers $+\infty$, des solutions de l'équation différentielle :

$$y'''(t) + 3y''(t) + 2y'(t) = 0, \quad t \in \mathbb{R}$$

Réponses de l'exercice avec préparation 3 :

1. Soient n et k deux entiers naturels tels que $k \leqslant n$. On pose :

$$\binom{n}{k} = \frac{n!}{k!(n-k)!}$$

Interprétation ensembliste : $\binom{n}{k}$ est le nombre de parties à k éléments d'un ensemble à n éléments.

2. a) Par équiprobabilité :

$$\mathbb{P}(A) = \frac{\binom{31}{4}}{\binom{32}{5}} = \frac{\frac{31!}{4! \times \cancel{27!}}}{\frac{32!}{5! \times \cancel{27!}}} = \frac{5! \times 31!}{4! \times 32!} = \frac{5}{32}$$

b) On remarque que $\overline{B}$: « Aucun as n'est dans la main ». Par équiprobabilité :

$$\mathbb{P}(\overline{B}) = \frac{\binom{28}{5}}{\binom{32}{5}} = \frac{\frac{28!}{\cancel{5!} \times 23!}}{\frac{32!}{\cancel{5!} \times 27!}} = \frac{\cancel{28} \times 27 \times 26 \times 25 \times 24}{32 \times 31 \times 30 \times 29 \times \cancel{28}} < \frac{1}{2} \qquad \text{(d'après l'énoncé)}$$

On en déduit que $\mathbb{P}(B) > \frac{1}{2}$. Or, $2\mathbb{P}(A) = \frac{5}{16} < \frac{1}{2}$.

$$\boxed{\text{On a bien : } \mathbb{P}(B) > 2\mathbb{P}(A).}$$

3. a) On propose la fonction suivante :

```python
def as_de_pique(m):
    for k in range(len(m)):
        val, coul = m[k]
        if val == 'as' and coul == 'pique':
            return True
    return False
```

b) On propose la fonction suivante :

```python
def nbre_as(m):
    compteur = 0
    for k in range(len(m)):
        val, coul = m[k]
        if val == 'as':
            compteur += 1
    return compteur
```

c) On propose la fonction suivante :

```
import numpy as np

def estimation(N)
    a,b = 0,0
    xA,xB = np.zeros(5),np.zeros(5)
    for i in range(N):
        main = ma_main()
        n = nbre_as(main)
        if n >= 1:
            b += 1
            xB[n] += 1
            if as_de_pique(main):
                a += 1
                xA[n] += 1
    sY = 0; sZ = 0
    for k in range(1,5):
        sY += k*xA[k]
        sZ += k*xB[k]
    if a*b != 0:
        return sY/a, sZ/b
```

4. a) Soit $k \in \{2, 3, 4\}$. Tout d'abord,

$$\mathbb{P}_A([X = k]) = \frac{\mathbb{P}(A \cap [X = k])}{\mathbb{P}(A)} \qquad \text{et} \qquad \mathbb{P}_B([X = k]) = \frac{\mathbb{P}(B \cap [X = k])}{\mathbb{P}(B)}$$

Ensuite, par équiprobabilité :

$$\mathbb{P}(A \cap [X = k]) = \frac{\binom{3}{k-1} \times \binom{28}{5-k}}{\binom{32}{5}}$$

(Il faut choisir $k - 1$ as autres que l'as de pique puis 5-k cartes qui ne sont pas des as)

et

$$\mathbb{P}(B \cap [X = k]) = \frac{\binom{4}{k} \times \binom{28}{5-k}}{\binom{32}{5}}$$

(Il faut choisir k as puis 5-k cartes qui ne sont pas des as)

Par la formule du capitaine, on a : $\binom{4}{k} = \frac{4}{k}\binom{3}{k-1}$ et donc

$$\mathbb{P}(B \cap [X = k]) = \frac{4}{k}\mathbb{P}(A \cap [X = k])$$

Puisque $k \geqslant 2$, on en déduit que :

$$\mathbb{P}(B \cap [X = k]) \leqslant 2\mathbb{P}(A \cap [X = k])$$

D'après la question ***2.b)***, $\mathbb{P}(B) > 2\mathbb{P}(A)$.

$$\boxed{\text{On peut alors conclure que : } \mathbb{P}_B([X = k]) < \mathbb{P}_A([X = k]).}$$

b) Tout d'abord, remarquons que $Y(\Omega) = Z(\Omega) = [\![1, 4]\!]$.
Ensuite, il est immédiat par définition de Y et Z que :

$$\forall k \in [\![1, 4]\!],\ \mathbb{P}([Y = k]) = \mathbb{P}_A([X = k]) \qquad \text{et} \qquad \mathbb{P}([Z = k]) = \mathbb{P}_B([X = k])$$

Puisque la famille $([X=k])_{k\in[\![1,4]\!]}$ est un sce, on a :

$$\mathbb{P}_A([X=1]) = 1 - \sum_{k=2}^{4} \mathbb{P}_A([X=k]) \qquad \text{et} \qquad \mathbb{P}_B([X=1]) = 1 - \sum_{k=2}^{4} \mathbb{P}_B([X=k])$$

D'où :

$$\mathbb{E}(Y) = \sum_{k=1}^{4} k\mathbb{P}_A([X=k]) = \mathbb{P}_A([X=1]) + \sum_{k=2}^{4} k\mathbb{P}_A([X=k]) = 1 + \sum_{k=2}^{4} (k-1)\mathbb{P}_A([X=k])$$

et

$$\mathbb{E}(Z) = \sum_{k=1}^{4} k\mathbb{P}_B([X=k]) = \mathbb{P}_B([X=1]) + \sum_{k=2}^{4} k\mathbb{P}_B([X=k]) = 1 + \sum_{k=2}^{4} (k-1)\mathbb{P}_B([X=k])$$

$$\boxed{\text{D'après la question } \textbf{\textit{4.a)}}, \ \mathbb{E}(Z) < \mathbb{E}(Y).}$$

Commentaire

Les variables aléatoires X, Y et Z ne sont pas définies sur le même espace probabilisé, mais cela n'empêche pas de comparer $\mathbb{E}(Y)$ et $\mathbb{E}(Z)$.

Commentaire

Pour les plus rapides, il peut être demandé de coder la fonction `ma_main()`. Voici une possibilité :

```python
liste_val = [7, 8, 9, 10, 'valet', 'dame', 'roi', 'as']
liste_coul = ['trefle', 'carreau', 'coeur', 'pique']

def ma_main():
    liste_main = []
    while len(liste_main) < 5:
        i = rd.randint(0,8)
        j = rd.randint(0,4)
        val, coul = liste_val[i], liste_coul[j]
        if not (val, coul) in liste_main:
            liste_main.append((val, coul))
    return liste_main
```

Réponses de l'exercice sans préparation 3 : Proposons deux méthodes.

- Première méthode : on traduit l'équation différentielle sous forme d'un système différentiel linéaire.
  On pose $X : t \mapsto \begin{pmatrix} y(y) \\ y'(t) \\ y''(t) \end{pmatrix}$, de sorte que l'équation soit équivalente à $X' = AX$ où $A = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -2 & -3 \end{pmatrix}$.
  Soit $\lambda \in \mathbb{R}$.

$$\begin{aligned}
\mathrm{rg}(A-\lambda I_3) &= \mathrm{rg}\left(\begin{pmatrix} -\lambda & 1 & 0 \\ 0 & -\lambda & 1 \\ 0 & -2 & -3-\lambda \end{pmatrix}\right) \\
&= \mathrm{rg}\left(\begin{pmatrix} -\lambda & 1 & 0 \\ 0 & -2 & -3-\lambda \\ 0 & -\lambda & 1 \end{pmatrix}\right) && L_2 \leftrightarrow L_3 \\
&= \mathrm{rg}\left(\begin{pmatrix} -\lambda & 1 & 0 \\ 0 & -2 & -3-\lambda \\ 0 & 0 & \lambda^2+3\lambda+2 \end{pmatrix}\right) && L_3 \leftarrow 2L_3 - \lambda L_2
\end{aligned}$$

D'où

$$\begin{aligned} \lambda \in \mathrm{Sp}(A) &\iff -\lambda = 0 \qquad \text{ou} \qquad \lambda^2 + 3\lambda + 2 = 0 \\ &\iff \lambda = 0 \qquad \text{ou} \qquad (\lambda+1)(\lambda+2) = 0 \end{aligned}$$

et donc les valeurs propres de A sont : 0, -1 et -2. La matrice A est carrée de taille 3 et possède 3 valeurs propres distinctes donc A est diagonalisable. Toutes les valeurs propres sont négatives ou nulles donc il existe un réel c tel que $\lim\limits_{t \to +\infty} y(t) = c$.

- Deuxième méthode : on procède à un changement de fonction inconnue.

  On pose $z = y'$. La fonction z vérifie l'équation différentielle $z'' + 3z' + 2z = 0$. Il s'agit d'une équation différentielle linéaire homogène d'ordre 2 à coefficients constants. Son équation caractéristique est : $x^2 + 3x + 2 = 0$, dont les solutions sont -1 et -2.

  Ainsi, il existe $(\lambda, \mu) \in \mathbb{R}^2$ tel que :

  $$\forall t \in \mathbb{R},\ z(t) = \lambda \mathrm{e}^{-t} + \mu \mathrm{e}^{-2t}$$

  La fonction y étant une primitive de z, il existe $(a, b, c) \in \mathbb{R}^3$ tel que :

  $$\forall t \in \mathbb{R},\ y(t) = a\mathrm{e}^{-t} + b\mathrm{e}^{-2t} + c$$

  $$\boxed{\text{On retrouve : } \lim_{t \to +\infty} y(t) = c.}$$