Devoir surveillé 6

Samedi 1er février 2025 – 4h00

Consignes à lire !

La qualité de la rédaction, le soin porté à la copie, la lisibilité, l'orthographe, la rigueur du vocabulaire ainsi que la clarté des raisonnements sont des critères importants d'évaluation.

Quelques précisions :

- **la copie doit être prise de sorte que la marge se situe à droite de chaque page,**
- **la première page de la copie doit rester vierge et sera réservée aux appréciations,**
- **toutes les pages de la copie devront être numérotées et rangées dans l'ordre de lecture,**
- **les résultats finaux doivent être clairement mis en évidence (soulignés ou encadrés),**
- **les questions d'un même exercice doivent être présentées dans l'ordre du sujet.**
- un aide-mémoire `Python` est donné en fin de sujet.
  Pour toutes les questions `Python` du sujet, on supposera avoir importé les différents modules nécessaires de la sorte :

```
import numpy as np
import numpy.random as rd
import numpy.linalg as al
import matplotlib.pyplot as plt
import pandas as pd
```

L'usage de tout matériel électronique est interdit. Aucun document n'est autorisé.

Si, au cours de l'épreuve, un candidat repère ce qui lui semble être une erreur d'énoncé, il la signalera sur sa copie et poursuivra sa composition en expliquant les raisons des initiatives qu'il sera amené à prendre.

"En dépit de tout ce qu'on peut vous raconter, les mots et les idées peuvent changer le monde."
John Keating, Le cercle des poètes disparus

Exercice 1

Question de cours. Matrices semblables : définition et propriétés.

Pour tout réel t, on considère la matrice $M(t) = \begin{pmatrix} t-1 & t & -t \\ -t-1 & -1 & t+1 \\ -1 & t & 0 \end{pmatrix}$.

Partie I. Réduction des matrices $M(t)$

1. L'ensemble $\{M(t),\ t \in \mathbb{R}\}$ est-il un sous-espace vectoriel de $\mathcal{M}_3(\mathbb{R})$?
2. On pose $A = M(0)$.
   2.a. Déterminer une base de $\ker(A)$.
   2.b. Montrer que -1 est valeur propre de A et déterminer une base de l'espace propre associé, noté $E_{-1}(A)$.
   2.c. En déduire que la matrice A est diagonalisable.
3. Justifier que, pour tout $t \in \mathbb{R}$, t est valeur propre de $M(t)$.

Pour la suite, on considère la matrice $Q = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 1 \\ 1 & 0 & 1 \end{pmatrix}$ et on admet que cette matrice est inversible.

4. Justifier que la famille $\left(\begin{pmatrix} 1 \\ 0 \\ 1 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \right)$ est une base de $\mathcal{M}_{3,1}(\mathbb{R})$, que l'on notera $\mathscr{B}$.
5. Soit $t \in \mathbb{R}$. On note f_t l'endomorphisme de $\mathcal{M}_{3,1}(\mathbb{R})$ canoniquement associé à la matrice $M(t)$. Vérifier que la matrice de f_t dans la base $\mathscr{B}$ est la matrice $\begin{pmatrix} -1 & t & 0 \\ 0 & -1 & 0 \\ 0 & 0 & t \end{pmatrix}$, que l'on notera $T(t)$.
   En déduire une égalité entre $M(t)$, $T(t)$, Q et Q^{-1}.
6. Soit $t \in \mathbb{R}$. Quelles sont les valeurs propres de la matrice $M(t)$?

Partie II. Résolution d'un système différentiel

On considère à présent le système différentiel suivant, noté (S) :

$$\forall t \in \mathbb{R},\ \begin{cases} x'(t) &= (t-1)x(t) &+ ty(t) &- tz(t) \\ y'(t) &= -(t+1)x(t) &- y(t) &+ (t+1)z(t) \\ z'(t) &= -x(t) &+ ty(t) & \end{cases}$$

d'inconnues x, y, z des fonctions de classe $\mathscr{C}^1$ sur $\mathbb{R}$.

7. Soient x, y, z des fonctions de classe $\mathscr{C}^1$ sur $\mathbb{R}$. On pose $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ et $Z = Q^{-1}X$. Vérifier que

$$\big(X \text{ est solution de } (S)\big) \iff \forall t \in \mathbb{R},\ Z'(t) = T(t)Z(t)$$

8. Soit $\beta \in \mathbb{R}$. Résoudre l'équation différentielle $w' + w = \beta t \mathrm{e}^{-t}$, où w est une fonction de classe $\mathscr{C}^1$ sur $\mathbb{R}$. On cherchera une solution particulière sous la forme $t \longmapsto (at^2 + bt + c)\mathrm{e}^{-t}$.
9. Soit w une fonction de classe $\mathscr{C}^1$ sur $\mathbb{R}$. Démontrer :

$$\big(w' - tw = 0\big) \iff \big(\exists \lambda \in \mathbb{R} \ /\ \forall t \in \mathbb{R},\ w(t) = \lambda \mathrm{e}^{\frac{t^2}{2}}\big)$$

10. Déduire des questions précédentes l'ensemble des solutions du système différentiel (S).

Exercice 2

Question de cours. Soient f une fonction définie sur un intervalle I de $\mathbb{R}$ et $a \in I$. Que signifie graphiquement le fait que a soit un point d'inflexion pour la courbe représentative de f ? Comment, dans le cas où f est de classe $\mathscr{C}^2$, le déterminer ?

Partie I

Pour tout couple (p, q) d'entiers naturels, on pose $I(p,q) = \displaystyle\int_0^1 x^p(1-x)^q \, dx$.

1. 1.a. Expliquer rapidement pourquoi cette intégrale est bien définie.
   1.b. Montrer, grâce à une intégration par parties, que l'on a :
   $$\forall (p,q) \in \mathbb{N} \times \mathbb{N}^*, \ I(p,q) = \frac{q}{p+1} I(p+1, q-1)$$
   1.c. En déduire :
   $$\forall (p,q) \in \mathbb{N} \times \mathbb{N}, \ I(p,q) = \frac{p!q!}{(p+q+1)!}$$
2. En déduire, pour tout $p \in \mathbb{N}$, la valeur de $\displaystyle\int_0^1 x^p(1-x)^p dx$.

Partie II

Dans cette partie, on désigne par n un entier naturel quelconque et on pose $\alpha_n = \dfrac{(2n+1)!}{(n!)^2}$. On considère la fonction b_n définie par :

$$\forall x \in \mathbb{R}, \ b_n(x) = \begin{cases} \alpha_n x^n (1-x)^n & \text{si } x \in [0,1] \\ 0 & \text{si } x \notin [0,1] \end{cases}$$

3. Montrer que b_n est une densité.

On considère désormais une suite de variables aléatoires $(X_n)_{n\in\mathbb{N}}$, où X_n admet b_n comme densité.

4. Reconnaître la loi de X_0.
5. 5.a. Utiliser la partie I pour montrer que X_n possède une espérance et que l'on a :
   $$\mathbb{E}(X_n) = \frac{1}{2}$$
   5.b. Toujours en utilisant la partie I, montrer que X_n possède une variance et exprimer $\mathbb{V}(X_n)$ en fonction de n.
   5.c. En déduire que :
   $$\forall \varepsilon > 0, \ \lim_{n\to+\infty} \mathbb{P}\left(\left[\left|X_n - \frac{1}{2}\right| \geqslant \varepsilon\right]\right) = 0$$

Partie III

Dans cette partie, on désigne par n un entier naturel et on se propose d'étudier la suite de fonctions $(f_n)_{n\in\mathbb{N}}$ définie par :

$$\forall x \in \mathbb{R}, \ f_n(x) = \alpha_n \int_0^x t^n(1-t)^n dt, \ \text{où } \alpha_n = \frac{(2n+1)!}{(n!)^2}.$$

On note (C_n) la courbe représentative de f_n dans un repère orthonormé du plan.

6. Déterminer $f_0(x)$ pour tout réel x.
7. 7.a. Donner la valeur de $f_n(1)$.
   7.b. Montrer, grâce au changement de variable $u = 1 - t$ dans l'intégrale définissant $f_n(x)$, que :
   $$\forall x \in \mathbb{R}, \ f_n(x) + f_n(1-x) = 1$$
   7.c. En déduire la valeur de $f_n\left(\dfrac{1}{2}\right)$.
8. 8.a. Montrer que f_n est dérivable sur $\mathbb{R}$ et déterminer, pour tout réel x, l'expression de $f_n'(x)$ en fonction de x et n.
   8.b. Étudier, suivant la parité de n, le signe de $f_n'(x)$ pour tout réel x.
9. 9.a. En utilisant la formule du binôme de Newton, exprimer $f_n(x)$ en fonction de x, pour tout réel x puis en déduire les valeurs de $\lim\limits_{x\to+\infty} f_n(x)$ et de $\lim\limits_{x\to-\infty} f_n(x)$ selon que n est pair ou impair.
   9.b. Dresser le tableau de variations de f_n (toujours en distinguant les cas n pair et n impair).
10. Dans cette question, n désigne un entier naturel supérieur ou égal à 1.
    10.a. Pour tout réel x, déterminer $f_n''(x)$ en fonction de x et n.
    10.b. En déduire que (C_n) possède un point d'inflexion si n est impair et trois si n est pair.
    10.c. Tracer, selon la parité de n, l'allure de (C_n).

Exercice 3

Question de cours. Convergence en loi d'une suite de variables aléatoires.

On considère une urne contenant initialement une boule bleue et deux boules rouges.
On effectue, dans cette urne, des tirages successifs de la façon suivante : on pioche une boule au hasard et on note sa couleur, puis on la replace dans l'urne en ajoutant une boule de la même couleur que celle qui vient d'être obtenue.
Pour tout k de $\mathbb{N}^*$, on note :

- B_k l'événement : "on obtient une boule bleue au k-ième tirage",
- R_k l'événement : "on obtient une boule rouge au k-ième tirage".

Partie I. Rangs d'apparition de la première boule bleue et de la première boule rouge

On définit la variable aléatoire Y égale au rang d'apparition de la première boule bleue et la variable aléatoire Z égale au rang d'apparition de la première boule rouge.

1. Recopier et compléter les lignes manquantes de la fonction suivante afin qu'elle simule l'expérience étudiée et renvoie une réalisation de Y.

```
def simule_Y():
    b=...
    r=...
    Y=1
    while ...
        ...
        Y+=1
    return Y
```

2. On suppose créée et complétée la fonction précédente et on exécute le script suivant :

```
c=0
for k in range(10000):
    if simule_Y()==1:
        c+=1
print(c/10000)
```

On obtient : `0.3343`. Interpréter ce résultat.

3. 3.a. Montrer que : $\forall n \in \mathbb{N}^*,\ \mathbb{P}([Y=n]) = \dfrac{2}{(n+1)(n+2)}$. Justifier ensuite que $Y(\Omega) = \mathbb{N}^*$.

   3.b. La variable aléatoire Y admet-elle une espérance ? une variance ?

4. Déterminer la loi de Z. La variable aléatoire Z admet-elle une espérance ? une variance ?

Partie II. Nombre de boules rouges obtenues au cours de n tirages

On définit, pour tout k de $\mathbb{N}^*$, la variable aléatoire X_k égale à 1 si l'on obtient une boule rouge au k-ième tirage et égale à 0 sinon.
On définit, pour tout n de $\mathbb{N}^*$, la variable aléatoire S_n égale au nombre de boules rouges obtenues au cours des n premiers tirages.

5. Donner, pour tout n de $\mathbb{N}^*$, une relation entre S_n et certaines variables aléatoires X_k pour $k \in \mathbb{N}^*$.
6. Écrire une fonction `Python` de sorte que l'exécution de `simule_S(n)` simule l'expérience étudiée et renvoie une réalisation de S_n.
7. Déterminer la loi de X_1, son espérance et sa variance.
8. 8.a. Déterminer la loi du couple (X_1, X_2).

   8.b. En déduire la loi de X_2.

   8.c. Les variables aléatoires X_1 et X_2 sont-elles indépendantes ?
9. Soient $n \in \mathbb{N}^*$ et $k \in [\![0, n]\!]$.

   9.a. Calculer $\mathbb{P}(R_1 \cap \ldots \cap R_k \cap B_{k+1} \cap \ldots \cap B_n)$.

   9.b. Justifier : $\mathbb{P}([S_n = k]) = \binom{n}{k} \mathbb{P}(R_1 \cap \ldots \cap R_k \cap B_{k+1} \cap \ldots \cap B_n)$.

   En déduire : $\mathbb{P}([S_n = k]) = \dfrac{2(k+1)}{(n+1)(n+2)}$.
10. Montrer que, pour tout n de $\mathbb{N}^*$, S_n admet une espérance et : $\mathbb{E}(S_n) = \dfrac{2n}{3}$.
11. Soit $n \in \mathbb{N}^*$.

    11.a. Montrer : $\forall k \in [\![0, n]\!],\ \mathbb{P}_{[S_n=k]}([X_{n+1} = 1]) = \dfrac{k+2}{n+3}$.

    11.b. En déduire : $\mathbb{P}([X_{n+1} = 1]) = \dfrac{\mathbb{E}(S_n)+2}{n+3}$.

    11.c. Déterminer alors la loi de la variable aléatoire X_{n+1}. Que remarque-t-on ?

Partie III. Étude d'une convergence en loi

On s'intéresse dans cette partie à la proportion de boules rouges obtenues lors des n premiers tirages.

On pose, pour tout n de $\mathbb{N}^*$, $T_n = \dfrac{S_n}{n}$.

12. Écrire une fonction `Python` prenant en argument un entier naturel non nul n et renvoyant un histogramme de fréquences, en $n+1$ classes, sur 100000 réalisations de la variable aléatoire T_n.
13. Justifier, pour tout n de $\mathbb{N}^*$: $\forall x < 0,\ \mathbb{P}\big([T_n \leqslant x]\big) = 0$, et : $\forall x > 1,\ \mathbb{P}\big([T_n \leqslant x]\big) = 1$.
14. Soit $x \in [0,1]$. Montrer, pour tout n de $\mathbb{N}^*$:
$$\mathbb{P}\big([T_n \leqslant x]\big) = \frac{(\lfloor nx \rfloor + 1)(\lfloor nx \rfloor + 2)}{(n+1)(n+2)}$$
15. En déduire que la suite de variables aléatoires $(T_n)_{n\in\mathbb{N}^*}$ converge en loi vers une variable aléatoire à densité, dont on précisera la fonction de répartition et une densité.

★★★★★★★ Fin ★★★★★★★

Aide-mémoire Python

Listes

- `[]` : Créer une liste vide
- `L[i:j]` : Désigne la liste composée des éléments de `L` dont les indices vont de `i` à `j-1`
- `L[i:j:pas]` : Désigne la liste composée des éléments de `L` d'indices `i, i+pas, i+2*pas,...` compris entre `i` et `j-1`
- `[a]*n` ou `n*[a]` : Créer une liste avec `n` fois l'élément `a`
- `L.append(a)` : Ajoute l'élément `a` à la fin de la liste `L`
- `L1 + L2` : Concatène les deux listes `L1` et `L2`
- `len(L)` : Renvoie le nombre d'éléments de la liste `L`
- `del(L[k])` : Supprime l'élément d'indice `k` de la liste `L`
- `L.count(a)` : Renvoie le nombre d'occurrences de `a` dans la liste `L`
- `L.remove(a)` : Enlève la première occurrence de la valeur `a` de la liste `L`
- `max(L)` : Renvoie le plus grand élément de la liste `L`
- `min(L)` : Renvoie le plus petit élément de la liste `L`
- `sum(L)` : Renvoie la somme de tous les éléments de la liste `L`
- `a in L` : Vaut `True` si `a` se trouve au moins une fois dans `L` et `False` sinon

Modules mathématiques

numpy

`import numpy as np`

- `np.array(L)` : Transforme la liste `L` en vecteur ou matrice numpy
- `np.linspace(a,b,n)` : Crée un vecteur de n valeurs uniformément réparties entre a et b (inclus)
- `np.zeros([n,m])` : Crée la matrice nulle de taille $n \times m$
- `np.zeros(n)` : Crée le vecteur nul de taille n
- `np.ones([n,m])` : Crée la matrice de taille $n \times m$ dont tous les coefficients valent 1
- `np.ones(n)` : Crée le vecteur de taille n dont tous les coefficients valent 1
- `np.eye(n)` : Crée la matrice identité de taille n
- `np.diag(L)` : Crée la matrice diagonale dont les termes diagonaux sont les éléments de la liste `L`
- `np.transpose(M)` : Renvoie la transposée de M
- `np.dot(M,P)` : Renvoie le produit matriciel MP
- `np.sum(M)` : Renvoie la somme de tous les éléments de M
- `np.prod(M)` : Renvoie le produit de tous les éléments de M
- `np.max(M)` : Renvoie le plus grand élément de M
- `np.min(M)` : Renvoie le plus petit élément de M
- `np.shape(M)` : Renvoie dans un couple le format de la matrice M
- `np.mean(M)` : Renvoie la moyenne des éléments de M
- `np.var(M)` : Renvoie la variance des éléments de M
- `np.std(M)` : Renvoie l'écart-type des éléments de M
- `np.median(M)` : Renvoie la médiane des éléments de M
- `np.cumsum(M)` : Renvoie la matrice des sommes cumulées (gauche à droite, haut à bas) des éléments de M

- `np.arange(a,b,eps)` : Renvoie la liste des flottants de a à b de pas constant eps
- `np.abs(x)` : Renvoie $|x|$
- `np.floor(x)` : Renvoie $\lfloor x \rfloor$
- `np.sqrt(x)` : Renvoie $\sqrt{x}$ si $x \geqslant 0$
- `np.log(x)` : Renvoie $\ln(x)$ si $x > 0$
- `np.exp(x)` : Renvoie e^x
- `np.e` : Renvoie e
- `np.pi` : Renvoie π

Sous module d'algèbre linéaire `linalg` de numpy

`import numpy.linalg as al`

- `al.inv(M)` : Renvoie l'inverse de la matrice carrée M si elle est inversible
- `al.eig(M)` : Si la matrice M est diagonalisable, renvoie un couple (D, P) où D est le vecteur des coefficients diagonaux d'une matrice diagonale semblable à M et P une matrice de passage associée
- `al.matrix_power(M,n)` : Renvoie la puissance `n`-ième de la matrice carrée M
- `al.matrix_rank(M)` : Renvoie le rang de la matrice M
- `al.solve(M,Y)` : Renvoie une solution de l'équation matricielle $MX = Y$ lorsque M est inversible

Sous module `random` de numpy pour la simulation probabiliste

`import numpy.random as rd`

- `rd.random([r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi uniforme $\mathscr{U}\left([0;1]\right)$.
- `rd.randint(a,b,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi uniforme discrète $\mathscr{U}\left(\llbracket a; b-1 \rrbracket\right)$.
- `rd.binomial(n,p,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi binomiale $\mathscr{B}(n;p)$
- `rd.geometric(p,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi géométrique $\mathscr{G}(p)$
- `rd.poisson(a,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi de Poisson $\mathscr{P}(a)$
- `rd.exponential(a,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi exponentielle $\mathscr{E}\left(\dfrac{1}{a}\right)$
- `rd.normal(m,d,[r,s])` : Simule une réalisation d'une matrice (`r`, `s`) dont les coefficients sont des variables aléatoires indépendantes qui suivent la loi normale $\mathscr{N}(m, d^2)$.

Si le paramètre `[r,s]` est remplacé par `r`, ces fonctions renvoient la réalisation d'un vecteur de longueur `r` correspondant à la loi en question, et si ce paramètre est omis, elles renvoient un seul coefficient suivant les mêmes contraintes.

Sous module graphique `pyplot` de `matplotlib`

`import matplotlib.pyplot as plt`

- `plt.plot(X,Y,options)` : Crée la courbe des points définis par les listes `X`, abscisses, et `Y`, ordonnées suivant les options graphiques définies par la chaîne de caractères `options`
- `plt.bar(X,Y)` : Crée le diagramme en bâtons défini par les listes ou vecteurs `X`, abscisses et `Y`, ordonnées
- `plt.hist(X,Y,density=True)` : Crée l'histogramme de fréquences des valeurs définies par la liste `X`, `Y` étant soit le nombre de classes, soit les bornes des classes
- `plt.boxplot(X)` : Génère le diagramme en boîte basé sur le vecteur de données `X`
- `plt.axis('equal')` : Rend le repère orthonormé
- `plt.xlim(xmin,xmax)` : Fixe les bornes de l'axe des abscisses
- `plt.ylim(ymin,ymax)` : Fixe les bornes de l'axe des ordonnées
- `plt.show()` : Affiche le graphique