DS 8 (B)

Devoir surveillé du Mardi 20 Février

La calculatrice est interdite. Durée : 4h

Exercice 1

On note E l'espace vectoriel des fonctions polynomiales de degré inférieur ou égal à 2 et on rappelle que la famille (e_0, e_1, e_2) est une base de E, les fonctions e_0, e_1, e_2 étant définies par :

$$\forall t \in \mathbb{R}, \quad e_0(t) = 1, \quad e_1(t) = t, \quad e_2(t) = t^2.$$

On considère l'application φ qui, à toute fonction P de E, associe la fonction, notée $\varphi(P)$, définie par :

$$\forall x \in \mathbb{R}, \quad (\varphi(P))(x) = \int_0^1 P(x+t)dt.$$

1. (a) Montrer que φ est linéaire.
   (b) Déterminer $(\varphi(e_0))(x)$, $(\varphi(e_1))(x)$ et $(\varphi(e_2))(x)$ en fonction de x, puis écrire $\varphi(e_0)$, $\varphi(e_1)$ et $\varphi(e_2)$ comme combinaison linéaire de e_0, e_1 et e_2.
   (c) Déduire des questions précédentes que φ est un endomorphisme de E.
2. (a) Écrire la matrice A de φ dans la base (e_0, e_1, e_2). On vérifiera que la première ligne de A est :
   $$\begin{pmatrix} 1 & \frac{1}{2} & \frac{1}{3} \end{pmatrix}.$$
   (b) Justifier que φ est un automorphisme de E.
   (c) La matrice A est-elle diagonalisable ?
3. Compléter les commandes `Python` suivantes pour que soit affichée la matrice A^n pour une valeur de n entrée par l'utilisateur :

```python
n = input('entrez une valeur pour n : ')
A = .....
print( ..... )
```

4. (a) Montrer par récurrence que, pour tout entier naturel n, il existe un réel u_n tel que l'on ait :
   $$A^n = \begin{pmatrix} 1 & \frac{n}{2} & u_n \\ 0 & 1 & n \\ 0 & 0 & 1 \end{pmatrix}$$
   Donner u_0 et établir que : $\forall n \in \mathbb{N}, \quad u_{n+1} = u_n + \frac{1}{6}(3n+2).$
   (b) En déduire, par sommation, l'expression de u_n pour tout entier n.
   (c) Écrire A^n sous forme de tableau matriciel.

Exercice 2

Dans tout l'exercice, la lettre n désigne un entier naturel.

On pose $u_n = \displaystyle\int_0^1 \frac{x^n}{4-x^2}\mathrm{d}x$ et on a en particulier $u_0 = \displaystyle\int_0^1 \frac{1}{4-x^2}\mathrm{d}x$.

1. (a) Déterminer les réels a et b tels que :

$$\forall x \in [0,1], \quad \frac{1}{4-x^2} = \frac{a}{2-x} + \frac{b}{2+x}$$

   (b) En déduire que $u_0 = \dfrac{1}{4}\ln(3)$.

2. Calculer u_1.

3. (a) Pour tout entier naturel n, exprimer $4u_n - u_{n+2}$ explicitement en fonction de n.

   (b) Compléter la fonction Python ci-dessous afin qu'elle renvoie la valeur de u_n à l'appel de `suite(n)`.

```python
def suite(n):
    if (-1)**n == 1 :
        u = np.log(3) / 4
        for k in range(2,n + 1,2) :
            u = 4 * u - ...
    else :
        u = np.log(2 / np.sqrt(3))
        for k in range(3,n + 1,2) :
            u = 4 * u - ...
    return u
```

4. (a) Utiliser la définition de la suite $(u_n)_{n\in\mathbb{N}}$ pour établir l'encadrement suivant :

$$\forall n \in \mathbb{N}, \quad \frac{1}{4(n+1)} \leq u_n \leq \frac{1}{3(n+1)}$$

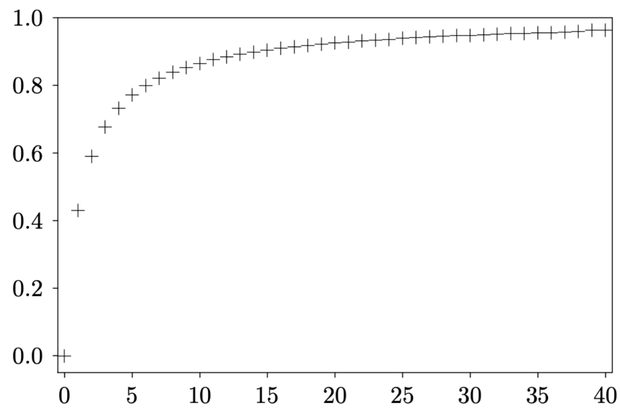
   (b) En déduire la convergence de la suite $(u_n)_{n\in\mathbb{N}}$ ainsi que la valeur de $\lim\limits_{n\to+\infty} u_n$.

   (c) La série de terme général u_n est-elle convergente ou divergente ? Pour quelle raison ?

5. (a) On considère le script suivant qui utilise la fonction déclarée plus haut :

```python
x = np.arange(0,41)
u = [] # liste vide
for n in range(41):
    u.append(3 * n * suite(n))
plt.plot(x,u,'+')
plt.show()
```

   Ce script renvoie le graphique suivant :



Laquelle des quatre conjectures suivantes peut-on émettre quant au comportement de la suite $(u_n)_{n\in\mathbb{N}}$ au voisinage de $+\infty$?

(1) $u_n \underset{+\infty}{\sim} 3n$ (2) $\lim\limits_{n\to+\infty} u_n = 1$. (3) $u_n \underset{+\infty}{\sim} \dfrac{1}{3n}$. (4) $u_n \underset{+\infty}{\sim} \dfrac{1}{n}$.

(b) Établir, grâce à une intégration par parties, l'égalité suivante :

$$\forall n \in \mathbb{N}, \quad u_n = \frac{1}{3(n+1)} - \frac{2}{n+1}\int_0^1 \frac{x^{n+2}}{(4-x^2)^2}\,\mathrm{d}x$$

(c) Montrer par encadrement que :

$$\lim_{n\to+\infty} \int_0^1 \frac{x^{n+2}}{(4-x^2)^2}dx = 0.$$

(d) Vérifier la conjecture établie à la question 5.(a).

Exercice 3

On considère la fonction f définie sur $\mathbb{R}$ par :

$$f(x) = \begin{cases} \dfrac{1}{2x^2} & \text{si } x \leq -1 \text{ ou } x \geq 1, \\ 0 & \text{sinon.} \end{cases}$$

1. (a) Vérifier que f est une fonction paire.
   (b) Montrer que f peut être considérée comme une fonction densité de probabilité.

Dans la suite, on considère une variable aléatoire X définie sur un espace probabilisé $(\Omega, \mathcal{A}, P)$, admettant f comme densité. On note F_X la fonction de répartition de X.

2. La variable aléatoire X admet-elle une espérance ?

3. On pose $Y = \ln(|X|)$ et on admet que Y est une variable aléatoire, elle aussi définie sur l'espace probabilisé $(\Omega, \mathscr{A}, P)$. On note F_Y sa fonction de répartition.

   (a) Montrer que, pour tout réel x, on a : $F_Y(x) = F_X(e^x) - F_X(-e^x)$.
   (b) Sans expliciter la fonction F_Y, justifier que Y est une variable aléatoire à densité.
   (c) Soit f_Y est une densité de Y. Montrer que pour tout réel x sauf peut-être en un nombre fini de points, on a : $f_Y(x) = 2e^x f(e^x)$.
   (d) Expliciter f_Y puis reconnaitre la loi de Y. En déduire sans calcul son espérance et sa variance.

4. (a) Montrer que, si x est positif, alors $1 - \mathrm{e}^{-x}$ appartient à $[0, 1[$ et montrer que, si x est strictement négatif, alors $1 - e^{-x}$ est strictement négatif.
   (b) On considère une variable aléatoire U suivant la loi uniforme sur $[0, 1[$. Déterminer la fonction de répartition de la variable aléatoire $Z = -\ln(1 - U)$ et reconnaître la loi de Z.
   (c) On rappelle qu'en Python, la fonction `rd.random()` permet de simuler la loi uniforme sur $]0, 1[$. Compléter la déclaration de fonction suivante pour qu'elle simule la variable aléatoire Y :

```python
def simulY():
    y = .....
    return(y)
```

Exercice 4

Partie I : étude d'une variable aléatoire

Les sommets d'un carré sont numérotés 1, 2, 3 et 4 de telle façon que les côtés du carré relient le sommet 1 au sommet 2, le sommet 2 au sommet 3, le sommet 3 au sommet 4 et le sommet 4 au sommet 1.
Un mobile se déplace aléatoirement sur les sommets de ce carré selon le protocole suivant :

- Au départ, c'est à dire à l'instant 0, le mobile est sur le sommet 1.
- Lorsque le mobile est à un instant donné sur un sommet, il se déplace l'instant suivant sur l'un quelconque des trois autres sommets, et ceci de façon équiprobable.

Pour tout n de $\mathbb{N}$, on note X_n la variable aléatoire égale au numéro du sommet sur lequel se situe le mobile à l'instant n. D'après le premier des deux points précédents, on a donc $X_0 = 1$.

1. Donner la loi de X_1, ainsi que l'espérance $E(X_1)$ de la variable aléatoire X_1.

   On admet pour la suite que la loi de X_2 est donnée par :
   $$P(X_2 = 1) = \frac{1}{3}, P(X_2 = 2) = P(X_2 = 3) = P(X_2 = 4) = \frac{2}{9}.$$

2. Pour tout entier naturel n supérieur ou égal à 2, donner, en justifiant, l'ensemble des valeurs prises par X_n.

3. (a) Utiliser la formule des probabilités totales pour établir que, pour tout entier naturel n supérieur ou égal à 2, on a :
   $$P(X_{n+1} = 1) = \frac{1}{3}\left(P(X_n = 2) + P(X_n = 3) + P(X_n = 4)\right).$$
   (b) Vérifier que cette relation reste valable pour $n = 0$ et $n = 1$.
   (c) Justifier que, pour tout n de $\mathbb{N}$: $P(X_n = 1) + P(X_n = 2) + P(X_n = 3) + P(X_n = 4) = 1$.
   En déduire l'égalité :
   $$\forall n \in \mathbb{N}, \quad P(X_{n+1} = 1) = -\frac{1}{3}P(X_n = 1) + \frac{1}{3}$$
   (d) Établir alors que : $\forall n \in \mathbb{N},\ P(X_n = 1) = \dfrac{1}{4} + \dfrac{3}{4}\left(-\dfrac{1}{3}\right)^n$.

4. (a) En procédant de la même façon qu'à la question précédente, montrer que l'on a :
   $$\forall n \in \mathbb{N},\ P(X_{n+1} = 2) = \frac{1}{3}\left(P(X_n = 1) + P(X_n = 3) + P(X_n = 4)\right).$$
   (b) En déduire une relation entre $P(X_{n+1} = 2)$ et $P(X_n = 2)$.
   (c) Montrer enfin que : $\forall n \in \mathbb{N}, \quad P(X_n = 2) = \dfrac{1}{4} - \dfrac{1}{4}\left(-\dfrac{1}{3}\right)^n$.

5. On admet que, pour tout entier naturel n, on a :
   $$P(X_{n+1} = 3) = -\frac{1}{3}P(X_n = 3) + \frac{1}{3} \quad \text{et} \quad P(X_{n+1} = 4) = -\frac{1}{3}P(X_n = 4) + \frac{1}{3}.$$
   En déduire sans calcul que :
   $$\forall n \in \mathbb{N},\ P(X_n = 3) = P(X_n = 4) = \frac{1}{4} - \frac{1}{4}\left(-\frac{1}{3}\right)^n.$$

6. Déterminer, pour tout entier naturel n, l'espérance $E(X_n)$ de la variable aléatoire X_n.

Partie 2 : calcul des puissances d'une matrice A

Pour tout n de $\mathbb{N}$, on considère la matrice ligne de $\mathscr{M}_{1,4}(\mathbb{R})$:

$$U_n = \begin{pmatrix} P(X_n = 1) & P(X_n = 2) & P(X_n = 3) & P(X_n = 4) \end{pmatrix}.$$

7. (a) Montrer (grâce à certains résultats de la partie 1) que, si l'on pose $A = \dfrac{1}{3}\begin{pmatrix} 0 & 1 & 1 & 1 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 \end{pmatrix}$, on a :
$$\forall n \in \mathbb{N},\ U_{n+1} = U_n A$$
   (b) Établir par récurrence que :
$$\forall n \in \mathbb{N}, \quad U_n = U_0 A^n.$$
   (c) En déduire la première ligne de A^n.

8. Expliquer comment choisir la position du mobile au départ pour trouver les trois autres lignes de la matrice A^n, puis écrire ces trois lignes.

Partie 3 : une deuxième méthode de calcul des puissances de A

On considère les matrices I et J suivantes : $I = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ et $J = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$.

9. Déterminer les réels a et b tels que $A = aI + bJ$.

10. (a) Calculer J^2 puis établir que, pour tout entier naturel k non nul, on a : $J^k = 4^{k-1}J$.
   (b) À l'aide de la formule du binôme de Newton, en déduire, pour tout entier naturel non nul, l'expression de A^n comme combinaison linéaire de I et de J.
   (c) Vérifier que l'expression trouvée reste vraie pour $n = 0$.

Partie 4 : informatique

11. (a) Expliquer soigneusement ce que fait la fonction `Python` suivante lorsqu'on lui donne en entrée une valeur $i \in [\![1, 4]\!]$:

```python
def deplacement(i):
    S = [k for k in range(1, 5) if k != i]
    r = rd.randint(3)
    return(S[r])
```

   (b) Compléter le script `Python` suivant pour qu'il affiche les 100 premières positions du mobile dont le voyage est étudié dans ce problème, ainsi que le nombre de fois n où il est revenu sur le sommet numéroté 1 au cours des 100 premiers instants.

```python
x = np.zeros(100)
x[0] = 1
for k in range(1,100):
    x[k] = deplacement( .... )
n = np.sum( ..... )
print(x, n)
```

   (c) Après avoir exécuté cinq fois ce script, les réponses concernant le nombre de fois où le mobile est revenu sur le sommet 1 sont $n = 23$, $n = 28$, $n = 25$, et $n = 26$. En quoi est-ce normal ?