

DS n°7

le 30/01/2026

Durée : 4h

- L'usage de la calculatrice n'est pas autorisé.
- La clarté et la précision des raisonnements interviendront pour une grande part dans la notation.
- Le résultat d'une question peut être admis afin de traiter une question suivante.
- On encadrera le résultat de chaque question.

Exercice 1 :

Pour tout entier $n \geq 1$ on pose $I_n = \int_0^1 \frac{t^n}{1+t} dt$

1. (a) Vérifier que pour tout réel t appartenant à $[0; 1]$ on a : $\frac{t}{1+t} = 1 - \frac{1}{1+t}$.
- (b) En déduire que $I_1 = 1 - \ln(2)$.
- (c) Montrer que pour tout entier $n \geq 1$ on a : $I_{n+1} + I_n = \frac{1}{n+1}$.
- (d) En déduire la valeur de I_2 puis celle de I_3 .

Soit k un réel strictement positif. On considère la fonction f définie sur $\mathbb{R}$ par :

$$f(t) = k \frac{t}{1+t} \quad \text{si } t \in [0; 1] \text{ et } f(t) = 0 \text{ sinon}$$

2. En utilisant l'un des calculs de la question 1. déterminer la valeur qu'il faut donner à k pour que f puisse être une densité de probabilité. Vérifier que pour cette valeur de k la fonction f est bien une densité de probabilité.

Dans la suite de l'exercice, on suppose que k est la valeur trouvée à la question 2. et que X est une variable aléatoire ayant f pour densité. On note F sa fonction de répartition.

3. (a) Calculer $F(x)$ lorsque $x < 0$ et lorsque $x > 1$.
- (b) Montrer que pour tout réel $x \in [0; 1]$ on a : $F(x) = k(x - \ln(1+x))$.
4. (a) En utilisant l'un des calculs de la question 1 justifier que X admet une espérance et que

$$E(X) = \frac{\ln(2) - \frac{1}{2}}{1 - \ln(2)}$$

- (b) En utilisant l'un des calculs de la question 1 justifier que X admet une variance et calculer $V(X)$. On ne demande pas de simplifier l'expression de $V(X)$.

Exercice 2

On suppose que toutes les variables aléatoires présentées dans cet exercice sont définies sur le même espace probabilisé.

Partie A

Soit f la fonction définie sur $\mathbb{R}$ par : $\forall t \in \mathbb{R}, f(t) = \begin{cases} 1/t^3 & \text{si } t \geq 1 \\ 0 & \text{si } -1 < t < 1. \\ -1/t^3 & \text{si } t \leq -1 \end{cases}$

1. Démontrer que la fonction f est paire.
2. Justifier que l'intégrale $\int_1^{+\infty} f(t) dt$ converge et calculer sa valeur.

3. (a) À l'aide d'un changement de variable, montrer que pour tout réel A strictement supérieur à 1, on a :

$$\int_{-A}^{-1} f(t) dt = \int_1^A f(u) du.$$
 En déduire que l'intégrale $\int_{-\infty}^{-1} f(t) dt$ converge et donner sa valeur.
- (b) Montrer que la fonction f est une densité de probabilité.
4. On considère une variable aléatoire X admettant f pour densité. On note F_X la fonction de répartition de X .
- (a) Montrer que, pour tout réel x , on a :
$$F_X(x) = \begin{cases} 1/2x^2 & \text{si } x \leq -1 \\ 1/2 & \text{si } -1 < x < 1 \\ 1 - 1/2x^2 & \text{si } x \geq 1 \end{cases}$$
- (b) Démontrer que X admet une espérance, puis que cette espérance est nulle.
- (c) La variable aléatoire X admet-elle une variance ?
5. Soit Y la variable aléatoire définie par $Y = |X|$.
- (a) Donner la fonction de répartition de Y , et montrer que Y est une variable aléatoire à densité.
- (b) Montrer que Y admet pour densité la fonction f_Y définie par :
$$f_Y(x) = \begin{cases} 2/x^3 & \text{si } x \geq 1 \\ 0 & \text{sinon} \end{cases}$$
- (c) Montrer que Y admet une espérance et la calculer.

Partie B

1. Soit D une variable aléatoire prenant les valeurs -1 et 1 avec équiprobabilité, indépendante de la variable aléatoire Y .
 Soit T la variable aléatoire définie par $T = DY$.
- (a) Déterminer la loi de la variable $Z = \frac{D+1}{2}$. En déduire l'espérance et la variance de D .
- (b) Justifier que T admet une espérance et préciser sa valeur.
- (c) Montrer que pour tout réel x , on a :

$$P(T \leq x) = \frac{1}{2} P(Y \leq x) + \frac{1}{2} P(Y \geq -x)$$

- (d) En déduire la fonction de répartition de T .
2. Soit U une variable aléatoire suivant la loi uniforme sur $]0, 1[$ et V la variable aléatoire définie par :

$$V = \frac{1}{\sqrt{1-U}}.$$

- (a) Rappeler la fonction de répartition de U .
- (b) Déterminer la fonction de répartition de V et vérifier que les variables V et Y suivent la même loi.
3. (a) On suppose que les bibliothèques `Numpy` et `Numpy.random` ont été importées par les commandes :

```
import numpy as np
import numpy.random as rd
```

Écrire une fonction en langage `Python`, d'en-tête `def D(n):`, qui prend un entier $n \geq 1$ en entrée, et renvoie une liste contenant n réalisations de la variable aléatoire D .

- (b) On considère le script suivant :

```
n = int(input("entrer n "))
a = D(n)
b = [ rd.random() for i in range(n)]
c = [ a[i]/np.sqrt(1-b[i]) for i in range(n)]
print(np.mean(c))
```

De quelle variable aléatoire les coefficients du vecteur `c` sont-ils une simulation ? Pour `n` assez grand, quelle sera la valeur affichée ? Justifier votre réponse.

Exercice 3 :

Les deux parties de cet exercice sont indépendantes l'une de l'autre.

Partie A : Réduction simultanée et spectre

Soit $\mathcal{M}_3(\mathbb{R})$ l'espace vectoriel des matrices carrées d'ordre trois à coefficients réels. On pose :

$$I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}, \quad J = \begin{pmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{pmatrix} \quad \text{et} \quad K = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{pmatrix},$$

et on considère $\mathcal{E} = \text{Vect}(I, J, K)$ le sous-espace vectoriel de $\mathcal{M}_3(\mathbb{R})$ engendré par les matrices I, J et K .

1. Montrer que (I, J, K) est une base de $\mathcal{E}$, en déduire la dimension de $\mathcal{E}$.
2. Justifier sans calcul que les matrices J et K sont diagonalisables.
3. (a) Exprimer la matrice J^3 comme un multiple de J .
(b) En déduire que les valeurs propres de J appartiennent à l'ensemble $\{-\sqrt{2}, 0, \sqrt{2}\}$.

On pose $U_1 = \begin{pmatrix} \sqrt{2} \\ 1 \\ 1 \end{pmatrix}$ et $U_2 = \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$.

1. (a) Vérifier que U_1 et U_2 sont des vecteurs propres de J .
(b) Déterminer un vecteur propre U_3 de J associé à la valeur propre $-\sqrt{2}$.
2. (a) Justifier que (U_1, U_2, U_3) est une base de $\mathcal{M}_{3,1}(\mathbb{R})$.
(b) Donner une matrice inversible P de $\mathcal{M}_3(\mathbb{R})$ telle que :

$$P^{-1}JP = \begin{pmatrix} \sqrt{2} & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\sqrt{2} \end{pmatrix}.$$

3. (a) Montrer que (U_1, U_2, U_3) est aussi une base de vecteurs propres de K .
(b) Déterminer la matrice $P^{-1}KP$.
4. Soit M une matrice de $\mathcal{E}$ de coordonnées $(a, b, c) \in \mathbb{R}^3$ dans la base (I, J, K) .
(a) Exprimer la matrice $P^{-1}MP$ sous la forme d'un tableau de nombres dépendant de a, b et c .
(b) En déduire les valeurs propres de M .
5. On considère l'application linéaire $s : \mathcal{E} \rightarrow \mathbb{R}^3$ définie par :

$$s(M) = s(aI + bJ + cK) = (a + b\sqrt{2} + c, a - c, a - b\sqrt{2} + c)$$

pour toute matrice $M = aI + bJ + cK$ avec $(a, b, c) \in \mathbb{R}^3$.

- (a) Donner la matrice S de s relativement à la base (I, J, K) de $\mathcal{E}$ et à la base canonique de $\mathbb{R}^3$.
- (b) Montrer que l'application linéaire s est bijective.

Partie B : Un algorithme de coloration des graphes

Soit $n \geq 1$ un entier, on considère un graphe non orienté G donné par sa matrice d'adjacence $A \in \mathcal{M}_n(\mathbb{R})$. On note $\mathcal{S} = \{s_0, \dots, s_{n-1}\}$ l'ensemble des sommets de G , dans les programmes informatiques on confondra un sommet s_i avec son numéro i . On dit que deux sommets sont *voisins* s'ils sont distincts et reliés par une arête.

Une *coloration* de G est une application $c : \mathcal{S} \rightarrow \mathbb{N}$ telle que $c(s_i) \neq c(s_j)$ si les sommets s_i et s_j sont voisins. Dans cette définition, $\mathbb{N}$ représente l'ensemble des « couleurs » disponibles, la coloration c attribue à chaque sommet une « couleur » de sorte que deux sommets voisins soient de « couleurs » différentes.

Le graphe G admet la *coloration triviale* donnée par $c(s_i) = i$ pour tout $i \in \llbracket 0, n-1 \rrbracket$; il peut cependant admettre une coloration nécessitant moins de n « couleurs ». Ainsi, le graphe à cinq sommets ci-dessous admet la coloration à trois « couleurs » définie par : $c(s_0) = 0$, $c(s_1) = 1$, $c(s_2) = 0$, $c(s_3) = 1$, $c(s_4) = 2$.

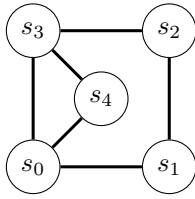


Figure 1 : Un graphe d'ordre cinq

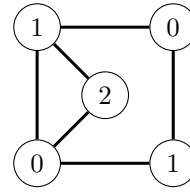


Figure 2 : Le graphe colorié avec trois « couleurs » (0, 1 et 2)

Les questions suivantes ont pour but de réaliser un programme Python qui renvoie une coloration d'un graphe G quelconque, en essayant de minimiser le nombre de couleurs utilisées. On commence par rédiger deux fonctions auxiliaires, « voisins » et « min_ext », qui serviront pour la fonction finale « coloration ». On suppose que la matrice d'adjacence A de G est définie à l'aide de la commande « `np.array` ».

1. Recopier et compléter le programme Python ci-dessous de manière à ce qu'il définisse une fonction « voisins », prenant en arguments la matrice d'adjacence A et un entier $i \in \llbracket 0, n-1 \rrbracket$, et renvoyant la liste des sommets voisins de s_i .

```
def voisins(A,i):
    n = len(A[i])
    V = []
    for j in range(n):
        if j!= i and ... :
            V.append(...)
    return(V)
```

2. Rédiger en Python une fonction « min_ext » qui prend en argument une liste d'entiers naturels L , et qui renvoie le plus petit entier naturel n'appartenant pas à L (par exemple, si $L = [1, 0, 3,]$ alors la commande « `min_ext(L)` » renvoie 2). On pourra transcrire en langage Python l'algorithme suivant :

- On affecte à une variable m la valeur 0.
- Tant que m appartient à la liste L :
On augmente de 1 la valeur de m .
- On renvoie m .

3. À l'aide des fonctions introduites précédemment on rédige maintenant une fonction « coloration » prenant en argument la matrice d'adjacence $A \in \mathcal{M}_n(\mathbb{R})$ d'un graphe G , et renvoyant une coloration de G sous la forme d'une liste d'entiers $C = [C_0, \dots, C_{n-1}]$, où C_i désigne la « couleur » du sommet s_i pour tout $i \in \llbracket 0, n-1 \rrbracket$.

On construit cette fonction selon l'algorithme « glouton » ci-dessous :

- On affecte à la variable n le nombre de sommets de G .
- On affecte à la variable C la liste $[0, 1, \dots, n-1]$.
- Pour i allant de 1 à $n-1$:
 . On affecte à la variable « `C_voisins` » la liste des « couleurs » des sommets voisins de s_i .
 . On affecte à C_i le plus petit entier naturel qui n'est pas élément de la liste « `C_voisins` ».
- On renvoie la liste C .

Recopier et compléter la fonction « coloration » ci-dessous.

```
def coloration(A):
    n = len(A[0])
    C = ...
    for i in range(1,n):
        C_voisins = [ ... for j in ... ]
        C[i] = min_ext(...)
    return(C)
```