

Durée : quatre heures

La présentation, la lisibilité, l'orthographe, la qualité de la rédaction, la clarté et la précision des raisonnements entreront pour une part importante dans l'appréciation des copies.

Les candidats sont invités à encadrer dans la mesure du possible les résultats de leurs calculs.

Ils ne doivent faire usage d'aucun document : l'utilisation de toute calculatrice et de tout matériel électronique est interdite.

Seule l'utilisation d'une règle graduée est autorisée.

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

Problème I : Une belle expo !

Partie A : Analyse des tableaux

On définit une fonction notée g sur $\mathbb{R}$ par $g(x) = (1-x)e^{-3x}$ et on notera $\mathcal{C}_1$ sa courbe représentative dans un repère orthogonal.

1. Calculer $g(1)$, $g(0)$ puis $\lim_{x \rightarrow +\infty} g(x)$ ainsi que $\lim_{x \rightarrow -\infty} g(x)$.
2. Justifier soigneusement que g est deux fois dérivable sur $\mathbb{R}$
3. Calculer $g'(x)$ pour $x \in \mathbb{R}$
4. Déterminer une équation de la tangente $\mathcal{T}_1$ à la courbe $\mathcal{C}_1$ en le point d'abscisse 1.
5. Etudier les variations de g sur $\mathbb{R}$ puis dresser le tableau des variations de g (en y insérant les limites trouvées en 1°)
6. Dresser le tableau des signes de g sur $\mathbb{R}$ (en justifiant)
7. Calculer $g''(x)$ en fonction de $x \in \mathbb{R}$ puis étudier la convexité de g sur $\mathbb{R}$
8. La courbe $\mathcal{C}_1$ admet-elle un point d'inflexion ? Si oui, quelles en sont les coordonnées ?
9. *Programmation Python*

- (a) Compléter le script Python suivant afin qu'il puisse calculer des images de la fonction g lorsque x est donné en entrées :

```
from math import *
def g(x):
    .....
    return.....
x=input("donner une valeur de x")
.....
```

On construit une représentation graphique de g à l'aide du script suivant :

```
import numpy as np
import matplotlib.pyplot as plt
X = np.linspace(-5, 5, 101)
Y = g(X)
plt.plot(X, Y)
```

- (b) Sur quel intervalle I cette représentation graphique est-elle réalisée ?
(c) Quel est le niveau de précision numérique des abscisses générées pour réaliser ce graphique ?
(d) Quelle commande taper à la console pour voir apparaître la représentation graphique générée ?

Partie B : Belle suite à intégrer

On pose à présent, pour $n \in \mathbb{N}^*$:

$$\begin{aligned} f_n : [0; 1] &\longrightarrow \mathbb{R} \\ x &\longmapsto (1-x)^n e^{-3x} \end{aligned}$$

et on pourra considérer que f_0 est la fonction $x \mapsto e^{-3x}$ définie sur $[0; 1]$.

1. Calculer $\int_0^1 e^{-3x} dx$. On notera u_0 cette valeur.
2. Etablir que, pour tout $n \in \mathbb{N}$, la fonction f_n est dérivable et continue sur $[0; 1]$.
3. Vérifier que, pour tout $n \in \mathbb{N}$, on a : $\forall x \in [0; 1] \quad f'_{n+1}(x) = -(n+1)f_n(x) - 3f_{n+1}(x)$
4. On note à présent $u_n = \int_0^1 f_n(t) dt$ pour $n \in \mathbb{N}^*$.
 - (a) Vérifier que $(u_n)_{n \in \mathbb{N}}$ ainsi définie est décroissante.
 - (b) Justifier que $(u_n)_{n \in \mathbb{N}}$ est positive.
 - (c) En déduire que $(u_n)_{n \in \mathbb{N}}$ converge vers un réel l positif.
5. Etablir que, pour tout $n \in \mathbb{N}$ on a $u_n \leq \frac{1}{n+1}$
6. En déduire la limite l de $(u_n)_{n \in \mathbb{N}}$.
7. Dans cette question, on cherche à déterminer la valeur de $\lim_{n \rightarrow +\infty} nu_n$.
 - (a) Vérifier que, pour tout $n \in \mathbb{N}$ on a :

$$f_{n+1}(0) - f_{n+1}(1) = (n+1) \int_0^1 f_n(t) dt + 3 \int_0^1 f_n(t) dt$$

Indication : On pourra utiliser la question 3° de cette partie.

- (b) Etablir la relation de récurrence satisfaite par $(u_n)_{n \in \mathbb{N}}$: $\forall n \in \mathbb{N} \quad u_{n+1} = \frac{1}{3} - \frac{nu_n}{3} - \frac{1}{3}u_n$
- (c) En déduire la valeur de limite cherchée.

Partie C : Couloirs en Séries

On pose, dans cette partie, pour $n \in \mathbb{N}$: $v_n = \sum_{k=0}^n \frac{(-1)^k 3^k}{k!}$

1. Ecrire un script Python qui prend n en entrée (un entier naturel) et qui renvoie la valeur de v_n .
2. Justifier que la suite $(v_n)_{n \in \mathbb{N}}$ converge et donner la valeur de sa limite.
3. En déduire la valeur de : $S = \sum_{n=0}^{+\infty} \frac{(-1)^n 3^n}{n!} - f_0(1)$
4. Etablir enfin que, pour tout entier naturel n on a : $u_n = \frac{(-1)^n n!}{3^{n+1}} (v_n - f_0(1))$

Problème II : Déboire avec la Matrice

Partie A (l'attaque d'une matrice en puissance)

On désignera par I_3 la matrice identité de $\mathcal{M}_3(\mathbb{R})$ et par O_3 la matrice nulle de $\mathcal{M}_3(\mathbb{R})$. La notation 0_3 désignera la matrice nulle de $\mathcal{M}_{3,1}(\mathbb{R})$. Soit enfin $T \in \mathcal{M}_3(\mathbb{R})$ matrice carrée définie par :

$$T = \begin{pmatrix} 0 & \frac{1}{2} & \frac{1}{2} \\ 0 & 0 & 1 \\ \frac{1}{2} & \frac{1}{2} & 0 \end{pmatrix}$$

- Calculer $(2T + I_3)^2$ puis $(2T + I_3)^2 (T - I_3)$.
- Résoudre l'équation matricielle d'inconnue $Z \in \mathcal{M}_{3,1}(\mathbb{R})$ définie par $(S) : TZ = 0_3$.
La matrice T est-elle inversible ? Justifier.
- Pour $\lambda \in \mathbb{R}$ donné, on définit $M_\lambda = T - \lambda I_3$.
 - Décrire les matrices M_1 , M_{-1} et $M_{-\frac{1}{2}}$ explicitement.
 - On pose E_1 l'équation matricielle d'inconnue $Z \in \mathcal{M}_{3,1}(\mathbb{R})$ définie comme $M_1 Z = 0_3$.
Vérifier que l'équation E_1 admet une infinité de solutions et en fournir une non nulle.
 - On pose $E_{-\frac{1}{2}}$ l'équation matricielle d'inconnue $Z \in \mathcal{M}_{3,1}(\mathbb{R})$ définie comme $M_{-\frac{1}{2}} Z = 0_3$.
Vérifier que l'équation $E_{-\frac{1}{2}}$ admet une infinité de solutions et en fournir une non nulle.

4. On considère $P = \begin{pmatrix} 1 & 1 & 7 \\ 1 & -2 & -6 \\ 1 & 1 & 1 \end{pmatrix}$.

- Vérifier que P est inversible et vérifie $P^{-1} = \frac{1}{18} \begin{pmatrix} 4 & 6 & 8 \\ -7 & -6 & 13 \\ 3 & 0 & -3 \end{pmatrix}$
- Soit $A = \begin{pmatrix} -\frac{1}{2} & 1 \\ 0 & -\frac{1}{2} \end{pmatrix}$. En vous aidant de la formule du binôme de Newton, expliciter A^n en fonction de $n \in \mathbb{N}$.
- La matrice A est-elle inversible, et si oui, quelle est son inverse ?
- On pose $D = P^{-1}TP$. Vérifier que pour tout entier naturel n on a :

$$D^n = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \left(\frac{-1}{2}\right)^n & n \left(\frac{-1}{2}\right)^{n-1} \\ 0 & 0 & \left(\frac{-1}{2}\right)^n \end{pmatrix}$$

Indication : On pourra raisonner par récurrence après avoir calculé D explicitement.

- Démontrer que $T^n = PD^nP^{-1}$ quel que soit $n \in \mathbb{N}$.
- En déduire T^n explicitement en fonction de $n \in \mathbb{N}$.

Partie B (ar ... de rire)

Un homme sort d'un bar, tard le soir, après avoir un peu abusé de la patience du patron.

Comme (mystérieusement) il ne voit plus très bien où il met les pieds, il se déplace aléatoirement entre trois lieux : Le Bar (B), sa Maison (M) et le commissariat (C). Chaque fois qu'il est en un lieu autre que le commissariat, il se déplace vers un autre lieu avec équiprobabilité (il ne reste pas sur place). En effet, à chaque passage au commissariat, l'homme est immédiatement reconduit à sa maison (ce qui lui vaut un déplacement).

On notera ainsi, abusivement, lorsque $n \in \mathbb{N}$ désigne le nombre de déplacements effectués par notre individu :

- C_n l'événement "l'individu est au commissariat après n déplacements"
- B_n l'événement "l'individu est au bar après n déplacements"
- M_n l'événement "l'individu se situe à son domicile (sa maison) après n déplacements"

On posera $b_n = \mathbb{P}[B_n]$, $c_n = \mathbb{P}[C_n]$ et $m_n = \mathbb{P}[M_n]$ pour $n \in \mathbb{N}$ et on pose $X_n = (b_n ; c_n ; m_n) \in \mathcal{M}_{1,3}(\mathbb{R})$.

- Déterminer une matrice M de $\mathcal{M}_3(\mathbb{R})$ telle que la relation $X_{n+1} = X_n M$ soit vérifiée pour tout $n \in \mathbb{N}$.
- Démontrer que $\forall n \in \mathbb{N} \quad X_n = X_0 \cdot M^n$
- Décrire, d'après l'énoncé, X_0 .
- Justifier que, pour tout $n \in \mathbb{N}$, on a $b_n + c_n + m_n = 1$ d'après l'énoncé.
- Vérifier par le calcul matriciel explicite de X_n que l'on retrouve bien $b_n + c_n + m_n = 1$
- Calculer $\lim_{n \rightarrow +\infty} b_n$ et $\lim_{n \rightarrow +\infty} m_n$.
- Après un nombre arbitrairement grand de déplacements, en quel lieu est-il le plus probable de trouver cet homme ?

Problème III : Un couple uni : quel beau tableau !

On lance trois pièces équilibrées et note X le nombre de *face* obtenu. Ensuite, on génère un nombre aléatoire noté Y suivant une loi **uniforme** sur $\llbracket 1; X + 1 \rrbracket$.

On note $(\Omega; \mathcal{A}; \mathbb{P})$ un espace de probabilités associé à cette expérience. $(X; Y)$ est ainsi un couple de VAR défini sur cet espace.

1. Rappeler l'espérance, la variance de X après avoir reconnu sa loi en justifiant soigneusement.
2. *Etudes marginales*
 - (a) Quelle est la loi conditionnelle de Y sachant $[X = 0]$ réalisé ?
 - (b) Pour chaque entier naturel non nul $n \leq 3$, donner la loi conditionnelle de Y sachant $[X = n]$ réalisé.
3. Déterminer la loi marginale de Y puis en calculer la valeur $\mathbb{E}[Y]$ de son espérance.
4. *Etude conjointe*
 - (a) Calculer, en détaillant votre démarche, la valeur de $\mathbb{P}[(X; Y) = (2; 1)]$
 - (b) Résumer, dans un tableau adéquat, la loi conjointe du couple $(X; Y)$.
 - (c) Ecrire une (ou plusieurs) instruction(s) Python permettant d'encoder, sous forme de matrice, le tableau qui précède.
5. Déterminer la loi conditionnelle de X sachant $[Y = 1]$ réalisé. Justifier soigneusement.
6. Déterminer, par le calcul, la valeur de $\mathbb{E}[XY]$. Les variables aléatoires X et Y sont-elles indépendantes ?
7. Calculer la covariance de X et Y notée $\text{cov}(X; Y)$.
8. *Programmation* On construit le script suivant afin de réaliser des simulations de l'expérience décrite :

```
from math import*
import numpy as np
import numpy.random as rd
import matplotlib.pyplot as plt
def simul():
    X = rd.binomial(3, 0.5)
    Y = rd.randint(1, X+2)
    return(np.array([X, Y]))
T = np.zeros([4, 4])
for k in range(100):
    C = simul()
    T[C[0], C[1]-1] = T[C[0], C[1]-1] + 1
```

On admettra que l'appel de `simul()` permet de générer une réalisation du couple $(X; Y)$ au format `nd.array`.

- (a) Combien de simulations ont effectivement été réalisées par ce script ? Justifier votre réponse en indiquant la (ou les) lignes vous permettant d'obtenir cette information.
- (b) La matrice T générée par l'exécution du script est la suivante :

```
>>> T
array([[13.,  0.,  0.,  0.],
       [20., 23.,  0.,  0.],
       [10., 11., 10.,  0.],
       [ 4.,  4.,  3.,  2.]])
```

Réorganiser ces données sous la forme d'un **tableau de tri croisé**.

- (c) Proposer une ou des instructions en langage python permettant d'obtenir la moyenne μ_Y des valeurs de Y ainsi générées.
A quelle valeur théorique devrait-on comparer μ_Y pour vérifier la cohérence du protocole proposé avec ce script ?