ECG2B
Lycée Marcelin Berthelot

Année scolaire 2025-2026
Mathématiques

Devoir surveillé n°3
10/01/2026
Durée : 4h

Coquilles corrigées

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 ***mettre en évidence*** *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.

S'il existe au moins un exercice dans lequel aucune question d'informatique n'est traitée sérieusement, un malus de 1 pt sera appliqué à la note finale.

Tout au long de ce problème on supposera les imports Python usuels réalisés :

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

Exercice 1 : SQL

Un amateur d'art décide de créer une base de données selon le schéma suivant :

- Une table `Artistes` répertoriant des artistes , qui contient 4 colonnes : un identifiant (), Nom - Pays - Siècle)

id_Artiste	Nom_Artiste	Pays	Siècle
1	Picasso	Espagne	20
2	Léonard de Vinci	Italie	15
3	Monet	France	14
⋮	⋮	⋮	⋮

- Une table `Musées` répertoriant des musées sous la forme

id_Musée	Nom_Musée	Pays	Ville	Prix	Ouverture	Fermeture
1	Louvre	France	Paris	15.5	10	20
2	Met	USA	New York	10	8	17
3	Uffizi	Italie	Florence	20	9	19
⋮	⋮	⋮	⋮	⋮	⋮	⋮

(dans cette table, l'entrée au Louvre coûte 15.5€ , et le Louvre est ouvert de 10h à 20h. On considérera que les horaires des musées peuvent s'exprimer avec des nombres entiers)

- Une table `Oeuvres` répertoriant des œuvres sous la forme :

Titre	Auteur	Genre	Musée	Estimation
L'Adoration des Mages	2	Peinture	3	1000000
La Cathédrale de Rouen	3	Peinture	2	500000
Guitare	1	Sculpture	1	80000
⋮	⋮	⋮	⋮	⋮

(dans cette table on peut lire que *L'Adoration des Mages* est une peinture de Léonard de Vinci, exposée aux Uffizi, et estimée à 1 000 000 €)
Les estimations[1] se feront par des nombres entiers.

1. Donner un code SQL permettant de créer la table `Artistes`, et d'y ajouter sa première ligne (on mettra une clé primaire si cela semble pertinent).

2. On suppose la table `Musées` créée, et dans laquelle on a déclaré le champ `id_Musée` comme clé primaire. Quelles clés étrangères doit-on mettre sur cette base de données ?

3. Il ne vous aura pas échappé que Monet est un peintre du 19ème siècle. Corriger cette erreur sur la table `Artistes` à l'aide d'une commande SQL.

4. Donner des requêtes SQL permettant de :

   (a) Lister les noms des artistes français.

   (b) Renvoyer les noms et pays des musées, classés par prix décroissant.

   (c) Afficher l'estimation moyenne de toutes les peintures présentes dans cette table.

   (d) Afficher les titres des peintures de Léonard de Vinci.

   (e) Lister les titres et estimations des sculptures exposées en France.

Exercice 2 (Réduction, et une chaîne de Markov)

Soit $N = \begin{pmatrix} 1 & 2 & 2 \\ 1 & 1 & 0 \\ 1 & 0 & 1 \end{pmatrix}$.

1. (a) Calculer $N - I_3$. En déduire une valeur propre de N ; et la dimension du sous-espace propre associé (sans déterminer celui-ci).

   (b) On rentre les commandes suivantes dans Python :

   ```
   N=np.array([[1,2,2],[1,1,0],[1,0,1]])

   print(np.dot(al.matrix_power(N,2)-np.eye(3),N-3*np.eye(3)))
   ```

   et on obtient l'affichage

   ```
   [[0. 0. 0.]
   [0. 0. 0.]
   [0. 0. 0.]]
   ```

   Déduire de cela un polynôme annulateur de N. Qu'en déduire sur ses valeurs propres ?

   (c) Déterminer une base du sous-espace propre $E_3(N)$.

   (d) Calculer les produits $N \times \begin{pmatrix} -2 \\ 1 \\ 1 \end{pmatrix}$ et $N \times \begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}$. En déduire le spectre de N, des bases de tous ses sous-espaces propres.

   (e) Donner P et D telles que $N = PDP^{-1}$.
   On rangera les coefficients de D par ordre décroissant ; et on choisira P de troisième ligne $\begin{pmatrix} 1 & 1 & 1 \end{pmatrix}$.

2. (a) Soit $Q = \begin{pmatrix} 1 & 1 & 1 \\ 0 & -2 & 2 \\ -1 & 1 & 1 \end{pmatrix}$. Calculer PQ ; et en déduire P^{-1}.

   (b) Démontrer : $\forall\, n \in \mathbb{N},\ N^n = PD^nP^{-1}$.

   (c) Donner la formule explicite de la première colonne de N^n pour tout entier naturel n.

Un facteur se déplace entre 3 villes dénotées 1,2,3 selon les règles suivantes :

- S'il est dans la ville 1 à une étape de temps, il se trouve à l'étape suivante dans l'une des 3 villes, choisie de manière équiprobable ;

[1] Désolé pour cette colonne bassement matérielle.

- S'il est dans la ville 2 ou 3 à une étape de temps, il y reste avec probabilité $\frac{1}{3}$, ou va dans la ville 1 avec proba $\frac{2}{3}$.

On suppose qu'à l'origine, le facteur est dans la ville 1.

3. (a) Programmer une fonction Python `ville_suivante(ville)` qui prend en argument un numéro de ville dans $[\![1,3]\!]$, dans laquelle le facteur se trouve, et renvoie la ville (aléatoire) dans laquelle il se trouvera à l'étape suivante (le nombre de lignes n'est pas imposé).

```
def ville_suivante(ville):
    if ville == 1:
        ville = ...
    else :  # on est en ville 2 ou 3
        ...
        ...
    return ville
```

   (b) En déduire une fonction Python `facteur(n)` qui prend en argument un entier n et renvoie une liste de taille $n+1$ donnant les villes visitées par le facteur aux temps $0,1,2,\dots,n$.

   (c) Compléter la fonction `freq_ville` ci-dessous, qui prend en argument un entier n et renvoie une liste à 3 éléments contenant la fraction de temps passée, après n étapes de temps, par le facteur dans chacune des trois villes 1,2,3.

```
def freq_ville(n):
    freq = np.zeros(3)
    L = ... # liste des villes parcourues
    for v in L :
        freq[...] = freq[...] + 1
    return ...
```

Pour tout $n\in\mathbb{N}$, on note X_n la variable aléatoire à valeurs dans $\{1,2,3\}$ donnant la position du facteur à la n-ième étape de temps. On admet que $(X_n)_{n\in\mathbb{N}}$ est une chaîne de Markov. On note $V_n = \big(\mathbb{P}(X_n=1) \quad \mathbb{P}(X_n=2) \quad \mathbb{P}(X_n=3)\big)$ le n-ème état probabiliste de cette chaîne.

4. (a) Expliquer comment se construit la matrice de transition de cette chaîne. On note cette matrice M ; vérifier que $M = \frac{1}{3}\,{}^t N$.

   (b) Déterminer le(s) état(s) stable(s) de cette chaîne.

   (c) Démontrer que $V_{n+1} = V_n M$.

   (d) En déduire une expression de V_n en fonction de V_0, M et n.

   (e) Montrer

$$\forall\, n\in\mathbb{N},\ V_n = \left(\frac{1}{2}\left(1+\left(-\frac{1}{3}\right)^n\right) \quad \frac{1}{4}\left(1-\left(-\frac{1}{3}\right)^n\right) \quad \frac{1}{4}\left(1-\left(-\frac{1}{3}\right)^n\right)\right)$$

5. (a) Calculer $\mathbb{E}(X_n)$.

   (b) Pour $i\in[\![1,3]\!]$, calculer $\lim\limits_{n\to+\infty}\mathbb{P}(X_n=i)$. Que retrouve-t-on ?

On admet qu'avec probabilité 1, il existe une étape de temps où le facteur se trouve dans la ville 3. On s'intéresse alors au temps de premier passage dans la ville 3. Ce temps est encore une variable aléatoire, notée B ; on a donc notamment :

$$\forall\, n\in\mathbb{N}^*,\ (B>n) = \left(\bigcap_{i=0}^{n}(X_i\neq 3)\right)$$

On introduit pour tout entier $n\in\mathbb{N}$ les événements :

$$U_{1,n} = \left(\bigcap_{i=0}^{n-1}(X_i\neq 3)\right)\cap(X_n=1) \quad \text{et} \quad U_{2,n} = \left(\bigcap_{i=0}^{n-1}(X_i\neq 3)\right)\cap(X_n=2)$$

(on a notamment $U_{1,0}=(X_0=1)$, $U_{2,0}=(X_0=2)$)

6. Coder une fonction Python `temps3()` utilisant notamment la fonction `ville_suivante` et renvoyant la valeur prise par B lors d'une modélisation du parcours du facteur.

7. (a) Comment interpréter les événements $U_{1,n}$ et $U_{2,n}$?
   (b) Montrer que $(B > n) = U_{1,n} \cup U_{2,n}$.

8. (a) Montrer les propriétés suivantes pour tout $n \in \mathbb{N}$:

$$U_{1,n+1} \cap (X_n = 1) = U_{1,n} \cap (X_{n+1} = 1) \quad \text{et} \quad \mathbb{P}\left(U_{1,n+1} \cap (X_n = 1)\right) = \frac{1}{3}\mathbb{P}(U_{1,n})$$

$$U_{1,n+1} \cap (X_n = 2) = U_{2,n} \cap (X_{n+1} = 1) \quad \text{et} \quad \mathbb{P}\left(U_{1,n+1} \cap (X_n = 2)\right) = \frac{2}{3}\mathbb{P}(U_{2,n})$$

   (b) En déduire :

$$\forall\, n \in \mathbb{N},\ \mathbb{P}(U_{1,n+1}) = \frac{1}{3}\mathbb{P}(U_{1,n}) + \frac{2}{3}\mathbb{P}(U_{2,n})$$

   On admet qu'un raisonnement similaire donne :

$$\forall\, n \in \mathbb{N},\ \mathbb{P}(U_{2,n+1}) = \frac{1}{3}\mathbb{P}(U_{1,n}) + \frac{1}{3}\mathbb{P}(U_{2,n})$$

9. On pose alors, pour tout entier n :

$$a_n = \mathbb{P}(U_{1,n})\ ;\ b_n = \mathbb{P}(U_{2,n})\ ;\ U_n = \begin{pmatrix} a_n \\ b_n \end{pmatrix}$$

   (a) Préciser la valeur de U_0.
   (b) Expliquez comment vous obtiendriez les expressions de a_n et b_n en fonction de n.
   (on ne demande pas de mettre en œuvre cette méthode)
   (c) On trouve :

$$\forall\, n \in \mathbb{N},\ \begin{cases} a_n = \dfrac{1}{2\sqrt{2} \times 3^n}\left(\sqrt{2}\left(1+\sqrt{2}\right)^n + \sqrt{2}\left(1-\sqrt{2}\right)^n\right) \\ b_n = \dfrac{1}{2\sqrt{2} \times 3^n}\left(\left(1+\sqrt{2}\right)^n - \left(1-\sqrt{2}\right)^n\right) \end{cases}$$

   En déduire l'expression de $\mathbb{P}(B > n)$ pour tout entier n.

10. On admet qu'en cas de convergence de la série de terme général $\mathbb{P}(B > n)$, B admet une espérance et $\mathbb{E}(B) = \sum_{n=0}^{+\infty} \mathbb{P}(B > n)$.
    Montrer que $\mathbb{E}(B) = \dfrac{9}{2}$.

Exercice 3 : Calcul différentiel

Les deux parties de cet exercice sont indépendantes.

Partie 1

On pose l'équation différentielle suivante (à coefficients non constants) :

$$(E)\ :\ y' = (-2t + 1)y$$

1. (a) Soit y une fonction de $\mathbb{R}$ dans $\mathbb{R}$, et $z : t \mapsto y(t)e^{t^2}$.
   Montrer que y est solution de (E) si et seulement si z est solution d'une équation différentielle linéaire à coefficients constants qu'on explicitera.
   (b) Montrer que les solutions de (E) sont les fonctions de la forme $t \mapsto Ke^{t-t^2}$ où K est une constante réelle.

Partie 2

On considère la matrice $A = \begin{pmatrix} -1 & -2 & 1 \\ -2 & -4 & 2 \\ 1 & 2 & -1 \end{pmatrix}$.

1. Le but de cette question est de diagonaliser la matrice A.

   (a) Justifier que la matrice A est de rang 1.

(b) En déduire une valeur propre de A ainsi qu'une base du sous-espace propre associé.

(c) Justifier que -6 est valeur propre de A et qu'un vecteur propre associé est $X_3 = \begin{pmatrix} 1 \\ 2 \\ -1 \end{pmatrix}$.

(d) Déterminer une matrice inversible P et une matrice diagonale D de $\mathcal{M}_3(\mathbb{R})$ telles que $A = PDP^{-1}$.

On considère le système différentiel suivant :

$$(S_1) \ : \ \begin{cases} x' &= -x - 2y + z \\ y' &= -2x - 4y + 2z \\ z' &= x + 2y - z \end{cases}.$$

2. Montrer, sans donner la forme générale des solutions, que $X : t \mapsto \begin{pmatrix} 1 + e^{-6t} \\ 2e^{-6t} \\ 1 - e^{-6t} \end{pmatrix}$ est une solution de (S_1).

3. (a) Soient $X_1 : t \mapsto X_1(t) = \begin{pmatrix} x_1(t) \\ y_1(t) \\ z_1(t) \end{pmatrix}$ et $X_2 : t \mapsto X_2(t) = \begin{pmatrix} x_2(t) \\ y_2(t) \\ z_2(t) \end{pmatrix}$ deux solutions du système (S_1).
On suppose qu'il existe $t_0 \in \mathbb{R}$ vérifiant $X_1(t_0) = X_2(t_0)$.
Que pouvez-vous dire de X_1 et X_2 ?

(b) Sans calcul mais en justifiant rigoureusement, donner la ou les solution(s) de (S_1) vérifiant $X(0) = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

4. Donner l'ensemble des solutions de (S_1). Quels sont les points d'équilibre de (S_1) ?

5. Déterminer la solution $X : t \mapsto X(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}$ du système (S_1) vérifiant $X(0) = \begin{pmatrix} 1 \\ 1 \\ 0 \end{pmatrix}$.

6. On utilise le solveur de systèmes différentiels de Python pour obtenir la solution du système différentiel (S_1) avec la condition initiale $X(0) = \begin{pmatrix} 3 \\ 3 \\ 1 \end{pmatrix}$ sur l'intervalle $[0, 1]$.
On trace les courbes de x, y, z et on obtient un des trois dessins suivants : lequel ? Justifier votre réponse.

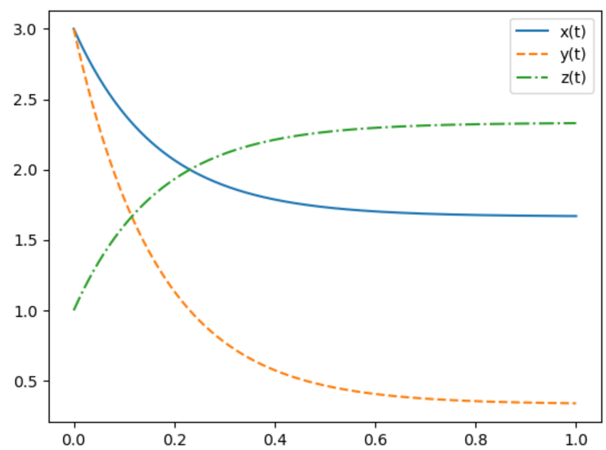

Figure 1

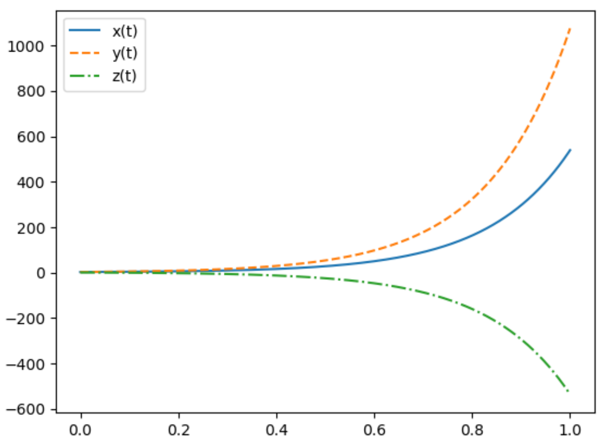

Figure 2

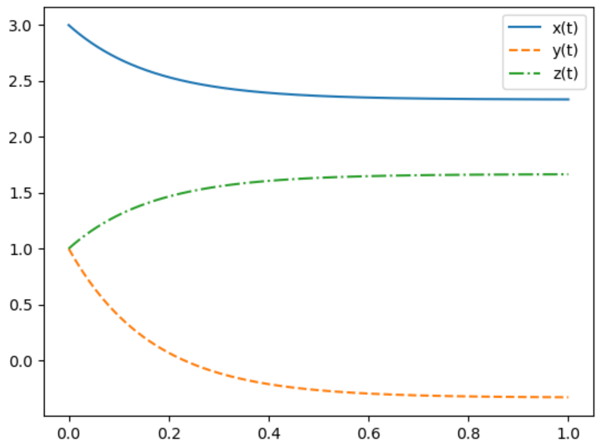

Figure 3