

DEVOIR DE RENTRÉE

ECG2 MATHS APPLIQUÉES

EXERCICE 1 *Échauffement.*

1. Déterminer toutes les fonctions φ définies et dérivables sur \mathbb{R} et solutions de l'équation différentielle

$$\varphi'(t) = 2\varphi(t).$$

2. Montrer que la fonction $t \mapsto te^{2t}$ est solution particulière de l'équation différentielle

$$\varphi'(t) = 2\varphi(t) + e^{2t}.$$

3. Déterminer toutes les fonctions φ définies et dérivables sur \mathbb{R} et solutions de l'équation différentielle

$$\varphi'(t) = 2\varphi(t) + e^{2t}.$$

EXERCICE 2 Algèbre/Calcul.

On considère les matrices

$$A = \begin{pmatrix} 0 & 1 & 1 \\ -2 & 3 & 2 \\ 1 & -1 & 0 \end{pmatrix} \quad \text{et} \quad I = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

1. a. On considère le programme Python suivant

```
1 import numpy as np
2 import numpy.linalg as alg
3
4 A = np.array([[0,1,1],[-2,3,2],[1,-1,0]])
5 B = A - np.eye(3)
6
7 print(alg.matrix_power(B,2))
```

L'exécution de ce programme retourne

```
1 [[0 0 0]
2  [0 0 0]
3  [0 0 0]]
```

Quelle égalité entre matrices est suggérée par ce programme ? Vérifier cette égalité par le calcul.

- b. En déduire que A est inversible et écrire A^{-1} comme combinaison linéaire de A et I .
2. On pose $N = A - I$.
- Trouver une base de l'image et du noyau de la matrice N .
 - Exprimer, pour tout entier naturel n , la matrice A^n comme combinaison linéaire de I et N .
 - Vérifier que l'expression précédente est aussi valable pour $n = -1$.
3. a. Résoudre le système $AX = X$ d'inconnue $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$.
- b. Justifier que pour un nombre réel $\lambda \neq 1$, le système $AX = \lambda X$ d'inconnue $X = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ n'a pas de solutions autre que $\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$.

EXERCICE 3 Analyse/Informatique d'après EMLyon 2023 Exercice 1.

Pour tout $x \in]0, +\infty[$, on pose $f(x) = \frac{e^{-x}}{x}$.

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 1$ et par la relation de récurrence $u_{n+1} = f(u_n)$, valable pour tout entier naturel n .

1. a. Étudier les variations de la fonction $f : x \mapsto f(x)$ (on dressera son tableau de variations, en précisant les limites).
b. Vérifier que chaque terme de la suite $(u_n)_{n \in \mathbb{N}}$ est correctement défini et strictement positif.
2. Informatique.

- a. Recopier et compléter la fonction Python suivante afin que l'appel `fonc_1(a)` renvoie le plus petit entier n tel que $u_n > a$.

```

1 def fonc_1(a) :
2     from numpy import exp
3     u = 1
4     n = 0
5     while ..... :
6         u = exp(-u)/u
7         n = .....
8     return n

```

- b. On considère maintenant la fonction Python

```

1 def fonc_2(a) :
2     from numpy import exp
3     u = 1
4     n = 0
5     while u > a :
6         u = exp(-u)/u
7         n = n+1
8     return n

```

Les appels `fonc_1(10**6)` et `fonc_2(10**(-6))` donnent respectivement 6 et 5.

Qu'en déduire pour u_5 et u_6 ?

Commenter ce résultat en une ligne.

- c. Écrire une fonction Python qui a pour argument un entier n et qui renvoie la valeur de u_n .
3. Pour $x \in [0, +\infty[$, on pose $g(x) = e^{-x} - x^2$.
 - a. Démontrer que la fonction $g : x \mapsto g(x)$ réalise une bijection de $[0, +\infty[$ sur $] -\infty, 1]$.
 - b. En déduire que l'équation $f(x) = x$ d'inconnue x possède une unique solution dans l'intervalle $]0, +\infty[$ que l'on notera α .
 - c. Justifier que $\frac{1}{e} < \alpha < 1$. On rappelle que $e \approx 2,7$.
4. a. Démontrer que l'on a : $u_2 > u_0$.
b. En déduire que la suite $(u_{2n})_{n \in \mathbb{N}}$ est croissante.
c. Justifier que la suite $(u_{2n+1})_{n \in \mathbb{N}}$ converge.
5. Pour $x \in]0, +\infty[$, on pose $h(x) = f \circ f(x)$. On pose également $h(0) = 0$.
 - a. Soit x un réel strictement positif. Déterminer $h(x)$.
 - b. Démontrer que la fonction $h : x \mapsto h(x)$ est continue sur $[0, +\infty[$.
 - c. Démontrer que l'équation $h(x) = x$ d'inconnue x admet exactement deux solutions sur $[0, +\infty[$ qui sont 0 et α , α étant le réel introduit à la question 3.b.
 - d. En déduire la limite de la suite $(u_{2n+1})_{n \in \mathbb{N}}$.
6. La suite $(u_{2n})_{n \in \mathbb{N}}$ est-elle majorée ? Admet-elle une limite ?

EXERCICE 4 Probabilités discrètes d'après ECRICOME 2022 Exercice 3 Partie I.

On dispose de trois urnes U_1, U_2 et U_3 et d'une infinité de jetons numérotés $1, 2, 3, \dots$

On répartit un par un les jetons dans les urnes : pour chaque jeton, on choisit au hasard et avec équiprobabilité une des trois urnes dans laquelle on place le jeton. Le placement de chaque jeton est indépendant de tous les autres jetons, et la capacité des urnes en jetons n'est pas limitée.

Pour tout entier naturel n , non nul, on note X_n (respectivement Y_n, Z_n) le nombre de jetons présents dans l'urne 1 (respectivement l'urne 2, l'urne 3) après avoir réparti les n premiers jetons.

Pour tout entier naturel non nul, on note V_n l'événement : "Après la répartition des n premiers jetons, au moins une urne reste vide".

1. Soit $n \in \mathbb{N}^*$.
 - a. Justifier que X_n, Y_n et Z_n suivent la même loi binomiale dont on précisera les paramètres.
 - b. Expliciter $P(X_n = 0)$ et $P(X_n = n)$.
 - c. Justifier que $(Y_n = 0) \cap (Z_n = 0) = (X_n = n)$.
 - d. Exprimer l'événement V_n à l'aide des événements $(X_n = 0)$, $(Y_n = 0)$ et $(Z_n = 0)$.
 - e. En déduire que

$$P(V_n) = 3 \left(\frac{2}{3}\right)^n - 3 \left(\frac{1}{3}\right)^n.$$

2. On note V l'événement "Au moins l'une des trois urnes reste toujours vide".
Exprimer l'événement V à l'aide des événements V_n , puis démontrer que $P(V) = 0$.
3. Soit T la variable aléatoire égale au nombre de jetons nécessaires pour que, pour la première fois, chaque urne contienne au moins un jeton.
 - a. Compléter le programme Python suivant pour qu'il simule le placement des jetons jusqu'au moment où chaque urne contient au moins un jeton, et pour qu'il renvoie la valeur prise par la variable aléatoire T .

```

1 import numpy.random as rd
2
3 def T():
4     X=0
5     Y=0
6     Z=0
7     n=0
8     L=[X,Y,Z]
9     while ..... :
10        i = rd.randint(0,3) # choix d'un nombre entier entre 0 et 2
11        L[i] = .....
12        n = n+1
13    return .....
```

- b. Écrire un programme Python qui simule 10 000 fois la variable aléatoire T et qui renvoie une valeur approchée de son espérance (en supposant pour le moment que son espérance existe).
4. Déterminer $T(\Omega)$.
5. Démontrer que pour tout $n \in T(\Omega)$,

$$P(T = n) = P(V_{n-1}) - P(V_n).$$

6. Démontrer que la variable aléatoire admet une espérance et calculer cette espérance.