CONCOURS BLANC 1 MATHS I

ECG2 MATHS APPLIQUÉES

Dans tout le sujet, on suppose déjà importées sous leur alias habituels les bibliothèques Python usuelles.

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

Exercice 1.- EMLyon 2009 Exercice 2.
On note $f : \mathbb{R} \to \mathbb{R}$ l'application définie par

$$f(x) = \begin{cases} \frac{x}{e^x - 1}, & \text{si } x \neq 0 \\ 1, & \text{si } x = 0 \end{cases}.$$

Partie I : Étude d'une fonction

1. **a.** Montrer que f est continue sur $\mathbb{R}$.
   b. Justifier que f est de classe $\mathcal{C}^1$ sur $]-\infty, 0[$ et sur $]0, +\infty[$, puis calculer $f'(x)$ pour tout $x \in \mathbb{R}^*$.
   c. Montrer que $\lim\limits_{x \to 0} f'(x) = \frac{-1}{2}$.
   d. Établir que f est de classe $\mathcal{C}^1$ sur $\mathbb{R}$ et préciser $f'(0)$.
2. **a.** Étudier les variations de l'application $u : \mathbb{R} \to \mathbb{R}$ définie par
   $$u(x) = (1 - x)e^x - 1.$$
   b. Montrer que pour tout $x \in \mathbb{R}$, $f'(x) < 0$.
   c. Déterminer les limites de f en $-\infty$ et en $+\infty$. Dresser le tableau de variations de f.
   d. Tracer l'allure de la courbe représentative de f.

Partie II : Étude d'une suite récurrente associée à la fonction f

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

3. Écrire une fonction Python d'en-tête `def suite_u(n)` : qui calcule et renvoie la valeur de u_n.
4. Montrer que f admet un point fixe et un seul, noté α que l'on calculera.
5. **a.** Établir que pour tout $x \in [0, +\infty[$, $e^{2x} - 2xe^x - 1 \geqslant 0$.
   b. Montrer que pour tout $x \in]0, +\infty[$, $f'(x) + \frac{1}{2} = \frac{e^{2x} - 2xe^x - 1}{2(e^x - 1)^2}$.
   c. Montrer que pour tout $x \in [0, +\infty[$, $-\frac{1}{2} \leqslant f'(x) \leqslant 0$.
   d. Établir que pour tout $n \in \mathbb{N}$, $|u_{n+1} - \alpha| \leqslant \frac{1}{2} |u_n - \alpha|$.
6. En déduire que pour tout $n \in \mathbb{N}$, $|u_{n+1} - \alpha| \leqslant \frac{1}{2^n}(1 - \alpha)$.
7. Conclure que la suite $(u_n)_{n \in \mathbb{N}}$ converge vers α.
8. Écrire un programme en Python qui calcule et affiche le plus petit entier $n \in \mathbb{N}$ tel que $|u_n - \alpha| < 10^{-9}$.
9. Si on ne connaissait pas la valeur de α, comment aurait-on pu déterminer un entier n tel que $|u_n - \alpha| < 10^{-9}$? Écrire l'appel correspondant en Python.

Date: 4 Décembre 2023 8h30-12h30.
http://louismerlin.fr.

Partie III : Étude d'une fonction définie par une intégrale

On note $G : \mathbb{R} \to \mathbb{R}$ l'application définie par
$$G(x) = \int_x^{2x} f(t)dt.$$

10. Montrer que G est de classe $\mathcal{C}^1$ sur $\mathbb{R}$ et, pour tout $x \in \mathbb{R}$,
$$G'(x) = \begin{cases} \frac{x(3-e^x)}{e^{2x}-1}, & \text{si } x \neq 0 \\ 1, & \text{si } x = 0 \end{cases}.$$

11. **a.** Montrer que pour tout $x \in [0, +\infty[$, $0 \leqslant G(x) \leqslant xf(x)$.
En déduire la limite de G en $+\infty$.

b. Montrer que pour tout $x \in]-\infty, 0]$, $G(x) \leqslant xf(x)$.
En déduire la limite de G en $-\infty$.

12. Dresser le tableau de variations de G. On n'essaiera pas de calculer $G(\ln(3))$.

Exercice 2.- EML 2020 Exercice 2 (légèrement modifié).
On définit, pour tous réels a et b la matrice $M(a,b)$ par

$$M(a,b) = \begin{pmatrix} a & 0 & 0 & a \\ a & 0 & 0 & a \\ a & 0 & 0 & a \\ b & b & b & b \end{pmatrix}$$

et on note $E = \left\{ M(a,b) \mid (a,b) \in \mathbb{R}^2 \right\}$. L'objectif de cet exercice est de déterminer les matrices de E qui sont diagonalisables.

1. **a.** Montrer que E est un sous-espace vectoriel de $\mathcal{M}_4(\mathbb{R})$.
Déterminer une base de E et sa dimension.

b. Le produit de deux matrices quelconque de E appartient-il encore à E ?

2. Étude du cas $a = 0$ et $b = 0$.
Justifier que la matrice $M(0,0)$ est diagonalisable.

3. Étude du cas $a \neq 0$ et $b = 0$.
Soit a un réel non nul. On note A la matrice $M(a,0)$.

a. Calculer A^2 et déterminer un polynôme annulateur de A.

b. En déduire les valeurs propres de la matrice A et préciser une base de chacun des sous-espaces propres associés.

c. En déduire que la matrice A est diagonalisable. Déterminer une matrice P de $\mathcal{M}_4(\mathbb{R})$ inversible et une matrice D de $\mathcal{M}_4(\mathbb{R})$ diagonale telles que $A = PDP^{-1}$.

4. Étude du cas $a = 0$ et $b \neq 0$.
Soit b un réel non nul. On note B la matrice $M(0,b)$.

a. On considère le programme Python suivant :

```
1  import numpy as np
2  import numpy.linalg as alg
3
4  b=3
5
6  B = np.array([[0,0,0,0],[0,0,0,0],[0,0,0,0],[b,b,b,b]])
7
8  r0 = alg.matrix_rank(B)
9  rb = alg.matrix_rank(B-b*np.eye(4))
10
11 print("r0 = ",r0)
12 print("rb = ",rb)
```

L'exécution de ce programme retourne

```
r0 = 1
rb = 3
```

et on admet que ce retour ne change pas lorsqu'on modifie la valeur de b à la ligne 4 du programme. Que peut-on conjecturer sur les valeurs propres de B et la dimension des espaces propres associés.

b. Vérifier les conjectures faites à la question précédente.

c. La matrice B est-elle diagonalisable ?

5. Étude du cas $a \neq 0$ et $b \neq 0$.
Soient a et b deux réels non nuls. On note f l'endomorphisme de $\mathbb{R}^4$ dont la matrice dans la base canonique est $M(a,b)$. On pose

$$v_1 = (1,1,1,0), \qquad v_2 = (0,0,0,1) \qquad \text{et} \qquad T = \begin{pmatrix} a & a \\ 3b & b \end{pmatrix}.$$

a. Montrer que $\mathrm{Ker}(f)$ est de dimension 2 et préciser une base (v_3, v_4) de $\mathrm{Ker}(f)$.

b. Montrer que la famille $\mathcal{B}' = (v_1, v_2, v_3, v_4)$ est une base de $\mathbb{R}^4$.

c. Déterminer la matrice notée N de l'endomorphisme f dans la base $\mathcal{B}'$.

d. Soit λ un réel non nul et $X = \begin{pmatrix} x \\ y \\ z \\ t \end{pmatrix}$ une matrice colonne de $\mathcal{M}_{4,1}(\mathbb{R})$. Montrer que X est un vecteur propre de N associé à la valeur propre λ *si et seulement si* ($\begin{pmatrix} x \\ y \end{pmatrix}$ est un vecteur propre de T associé à la valeur propre λ et $z = t = 0$).

e. On suppose dans cette question **uniquement** que $(a, b) = (1, 1)$
Déterminer les valeurs propres de T. En déduire que la matrice $M(1, 1)$ est diagonalisable.

f. On suppose dans cette question **uniquement** que $(a, b) = (1, -1)$.
Justifier que T n'admet aucune valeur propre. La matrice $M(1, -1)$ est-elle diagonalisable ?

g. Montrer l'équivalence

$$M(a, b) \text{ est diagonalisable } \Leftrightarrow a^2 + 10ab + b^2 > 0.$$

Exercice 3.- EDHEC 2021 Problème.
On dispose de deux pièces identiques donnant pile avec le probabilité p, élément de $]0,1[$, et face avec la probabilité $q = 1 - p$.

PARTIE I : UN JEU NAÏF

Deux joueurs A et B s'affrontent lors de lancers de ces pièces de la façon suivante., les lancers de chaque pièce étant supposés indépendants :

Pour la première manche, A et B lancent chacun leur pièce simultanément jusqu'à ce qu'ils obtiennent pile, le gagnant du jeu étant celui qui obtient pile le premier. En cas d'égalité, et en cas d'égalité seulement, les joueurs participent à une deuxième manche dans les même conditions et avec la même règle, et ainsi de suite jusqu'à la victoire de l'un d'entre eux.

Pour tout k de $\mathbb{N}^*$, on note X_k (resp. Y_k) la variable aléatoire égale au rang d'obtention du premier pile par A (resp. B) lors de la k-ième manche.

On note, toujours pour k dans $\mathbb{N}^*$, E_k l'événement : "il y a égalité à la fin de la k-ième manche" et on note E l'événement : "il y a perpétuellement égalité".

On note G (resp. H) l'événement : "A (resp. B) gagne à ce jeu" et pour tout entier naturel n non nul, on note G_n (resp. H_n) l'événement : "A (resp. B gagne le jeu à la n-ième manche".

1. Étude de la première manche.

- **a.** Donner la loi commune à X_1 et Y_1. En déduire qu'il est quasi-impossible que la première manche dure éternellement. On admet alors qu'il en est de même pour chaque manche jouée.
- **b.** Écrire l'événement E_1 à l'aide des variables X_1 et Y_1.
- **c.** Montrer que $P(E_1) = \sum\limits_{i=1}^{+\infty} P(X_1 = i)P(Y_1 = i)$ et en déduire l'expression explicite de $P(E_1)$ en fonction de p et q.
- **d.** Justifier sans aucun calcul que les événements G_1 et H_1 sont équiprobables. En déduire la probabilité de G_1 en fonction de p et q.

2. Calcul de la probabilité de l'événement G.

- **a.** Écrire, pour tout entier naturel n supérieur ou égal à 2, l'événement G_n à l'aide des événements E_k et de l'événement $(X_n < Y_n)$.
- **b.** Pour tout entier k supérieur ou égal à 2, calculer $P_{E_1\cap\cdots\cap E_{k-1}}(E_k)$ puis en déduire que pour tout entier $n \geqslant 2$,
$$P(G_n) = \left(\frac{p}{1+q}\right)^{n-1} \frac{q}{1+q}.$$
- **c.** Vérifier que le résultat précédent reste valable pour $n = 1$.
- **d.** Exprimer G en fonction des G_n puis conclure, après calcul que $P(G) = \frac{1}{2}$.
- **e.** Expliquer comment obtenir la probabilité de l'événement H et en déduire que ce jeu a presque sûrement une fin, c'est-à-dire que $P(E) = 0$.

PARTIE II : UN AUTRE JEU

En parallèle du jeu précédent, A parie sur le fait que la manche gagnée par le vainqueur le sera par un lancer d'écart et B parie sur le contraire.

3. **a.** À l'aide du système complet d'événements $(X_1 = i)_{i\in\mathbb{N}^*}$, montrer que $P(Y_1 = X_1 + 1) = \frac{pq}{1+q}$.

- **b.** En déduire la probabilité u que l'un des deux joueurs gagne à la première manche par un lancer d'écart.

4. **a.** Utiliser les événements E_k pour décrire l'événement K_n : "l'un des deux joueurs gagne la n-ième manche par un lancer d'écart".

- **b.** En déduire, pour tout entier naturel non nul n, la valeur de $P(K_n)$.

5. Donner finalement la probabilité de l'événement K :"A gagne ce pari".

Partie III : informatique

On rappelle que la commande rd.rand() renvoie un nombre uniformément au hasard dans l'intervalle $[0, 1]$.

6. Écrire un programme python d'en-tête def geom(p) permettant de simuler une variable aléatoire qui suit la loi géométrique de paramètre p.

7. Compléter le script suivant pour qu'il simule l'expérience décrite dans la partie I et affiche le nom du vainqueur du premier jeu ainsi que le numéro de la manche à laquelle il a gagné.

```python
p = input('entrez une valeur pour p')
c = 1
X = geom(p)
Y = geom(p)
while X == Y:
  X = ......
  Y = ......
  c = ......
if X < Y :
  ......
else :
  ......
print(c)
```

8. Compléter la commande suivante afin qu'une fois ajoutée au script précédent elle permette de simuler le deuxième jeu et d'en donner le nom du vainqueur.

```python
if ...... :
  print('A gagne le deuxieme jeu')
else :
  ......
```