

DS1

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. Aucun document n'est autorisé. L'utilisation de toute calculatrice et de tout matériel électronique est interdite. 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.

Exercice 1 Résoudre le système suivant :

$$\begin{cases} x + y + 2z = 5 \\ x - y - z = 1 \\ x + z = 3. \end{cases}$$

Exercice 2 On considère la matrice $A = \begin{pmatrix} 2 & 0 & 1 \\ 0 & 3 & 1 \\ 0 & 0 & 3 \end{pmatrix}$.

1. Montrer par récurrence que pour tout entier naturel n , on a $A^n = \begin{pmatrix} 2^n & 0 & 3^n - 2^n \\ 0 & 3^n & n3^{n-1} \\ 0 & 0 & 3^n \end{pmatrix}$.

2. Application à l'étude de deux suites.

On considère les deux suites $(a_n)_{n \geq 0}$ et $(b_n)_{n \geq 0}$ définies par $a_0 = 2$, $b_0 = 0$ et pour tout entier naturel n ,

$$a_{n+1} = 2a_n + 3^n \quad \text{et} \quad b_{n+1} = 3b_n + 3^n.$$

a. Dans le script Python ci-dessous, quelle instruction faut-il ajouter en ligne 4 pour que la fonction suite retourne la valeur de a_n pour un entier n donné ?

```
1 def suite(n):
2     a = 2
3     for i in range(n):
4         ...
5     return a
```

b. Pour tout entier naturel n , on pose $X_n = \begin{pmatrix} a_n \\ b_n \\ 3^n \end{pmatrix}$. Montrer que pour tout entier naturel n , on a

$$X_{n+1} = AX_n.$$

c. Recopier et compléter les lignes 4, 5 et 7 du script ci-dessous afin que la fonction suite_bis retourne la valeur de a_n pour un entier n donné.

```
1 import numpy as np
2
3 def suite_bis(n):
4     A = np.array(...)
5     X = np.array(...)
6     for i in range(n):
7         X = np.dot(...)
8     return X[0]
```

d. Montrer par récurrence que pour tout entier naturel n , on a $X_n = A^n X_0$.

e. En déduire, à l'aide de la question 1, que pour tout entier naturel n , on a $a_n = 2^n + 3^n$ et $b_n = n3^{n-1}$.

3. Application au calcul des puissances d'une autre matrice.

On considère les matrices $M = \begin{pmatrix} 4 & 0 & -2 \\ -1 & 3 & 1 \\ 1 & 0 & 1 \end{pmatrix}$, $P = \begin{pmatrix} 0 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 1 \end{pmatrix}$ et $Q = \begin{pmatrix} -1 & 0 & -1 \\ 0 & 1 & 0 \\ -1 & 0 & 0 \end{pmatrix}$.

- Calculer PQ et QP .
- Vérifier que $PMQ = A$.
- Montrer par récurrence que pour tout entier naturel n , on a $M^n = QA^nP$.

En déduire que pour tout entier naturel n , on a $M^n = \begin{pmatrix} 2 \times 3^n - 2^n & 0 & 2(2^n - 3^n) \\ -n3^{n-1} & 3^n & n3^{n-1} \\ 3^n - 2^n & 0 & 2^{n+1} - 3^n \end{pmatrix}$.

4. Application au calcul d'une somme.

- Montrer que pour tout entier naturel k , on a $2b_k = b_{k+1} - b_k - 3^k$.
- Pour tout entier naturel n , calculer $\sum_{k=0}^n 3^k$.
- Montrer que pour tout entier naturel n , on a $\sum_{k=0}^n (b_{k+1} - b_k) = b_{n+1}$.
- Déduire des questions précédentes et de la question 2.e que pour tout entier naturel n , on a

$$\sum_{k=0}^n k3^{k-1} = \frac{(n+1)3^n}{2} + \frac{1}{4} - \frac{3^{n+1}}{4}.$$

Exercice 3

A. Soient f et g les fonctions définies sur $\mathbb{R}$ par

$$f(x) = e^x + e^{-x} \quad \text{et} \quad g(x) = e^x - e^{-x}.$$

On admet que f et g sont dérivables sur $\mathbb{R}$.

- Montrer que pour tout réel x , on a $f'(x) = g(x)$ et $g'(x) = f(x)$.
- Déterminer les limites de g en $-\infty$ et en $+\infty$.
 - Étudier les variations de g et dresser son tableau de variations.
- Résoudre l'équation $g(x) = 0$.
 - Dresser le tableau de signes de $g(x)$.
 - En déduire le tableau de variations de f . On y fera figurer les limites en $-\infty$ et en $+\infty$.
 - Étudier la convexité de f .
- Étudier la convexité de g .
- On note $\mathcal{C}_f$ et $\mathcal{C}_g$ les courbes respectives de f et g .
 - Déterminer une équation de la tangente $\mathcal{T}$ à $\mathcal{C}_g$ au point d'abscisse 0.
 - Préciser la position relative de $\mathcal{C}_g$ et $\mathcal{T}$.
 - En déduire que pour tout $x \leq 0$, on a $g(x) \leq 2x$, et que pour tout $x \geq 0$, on a $g(x) \geq 2x$.
- Justifier que pour tout réel x , on a $f(x) \geq g(x)$.
 - Que peut-on en déduire concernant la position relative de $\mathcal{C}_f$ et $\mathcal{C}_g$?
- Dans un repère orthogonal, tracer l'allure des courbes $\mathcal{C}_f$ et $\mathcal{C}_g$, ainsi que la droite $\mathcal{T}$.
- Notons $\mathcal{A}$ l'aire (exprimée en unités d'aire) de la surface délimitée par $\mathcal{T}$ et $\mathcal{C}_f$ entre les bornes 0 et 1.
 - Hachurer cette surface sur le graphique de la question 7.
 - Déterminer l'aire de la surface comprise entre $\mathcal{C}_f$, l'axe des abscisses et les droites d'équations $x = 0$ et $x = 1$.
 - Calculer $\mathcal{A}$.
- On considère les fonctions h et k définies sur $\mathbb{R}$ par

$$h(x) = f(x) - 2 - x^2 \quad \text{et} \quad k(x) = g(x) - 2x - \frac{1}{3}x^3.$$

On admet que h et k sont dérivables sur $\mathbb{R}$.

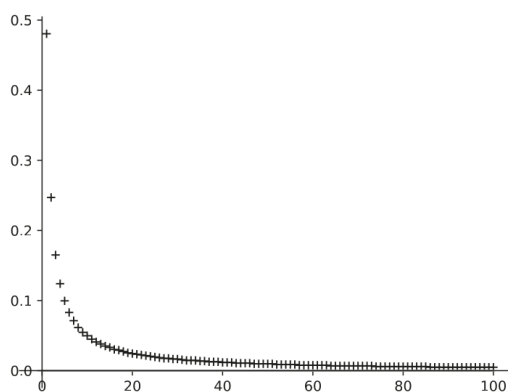
- a. Pour tout réel x , calculer $h'(x)$.
 - b. En déduire que pour tout réel x , on a $f(x) \geq 2 + x^2$.
 - c. Pour tout réel x , calculer $k'(x)$.
 - d. En déduire que pour tout $x \leq 0$, on a $g(x) \leq 2x + \frac{1}{3}x^3$, et que pour tout $x \geq 0$, on a $g(x) \geq 2x + \frac{1}{3}x^3$.
- B. 1.** Soit n un entier naturel non nul. Montrer que l'équation $g(x) = \frac{1}{n}$ admet une unique solution.
- Dans la suite, on note u_n cette solution. Ainsi, pour tout entier naturel n non nul, on a $g(u_n) = \frac{1}{n}$.
2. Montrer que $0 < u_1 < 1$.
 3. Montrer que pour tout entier naturel n non nul, on a $u_n > 0$.
 4.
 - a. Montrer que la suite $(u_n)_{n \geq 1}$ est décroissante.
 - b. En déduire que $(u_n)_{n \geq 1}$ est convergente.
 5. La bibliothèque numpy est importée de la manière suivante :

```
import numpy as np
```

- a. Écrire, en langage Python, une fonction nommée d qui prend en entrée un réel x et un entier naturel n non nul et qui renvoie le réel $g(x) - \frac{1}{n}$.
- b. Recopier et compléter les lignes 5, 8, 10 et 11 de la fonction Python ci-dessous, nommée SuiteU, qui prend en entrée un entier naturel non nul n et qui renvoie sous la forme d'une liste des valeurs approchées à 10^{-3} près des termes $u_1, \dots, u_n$ de la suite $(u_n)_{n \geq 1}$.

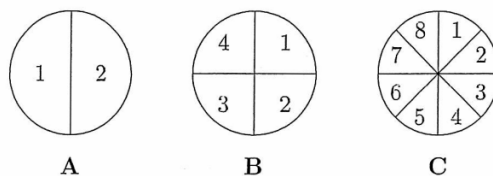
```
1 def SuiteU(n):
2     U = np.zeros(n)
3     for k in range(n):
4         a = 0 ; b = 1
5         while b - a > ...:
6             c = (a + b) / 2
7             if d(a, k+1) * d(c, k+1) < 0:
8                 b = ...
9             else:
10                ...
11            U[k] = ...
12    return(U)
```

- c. Les fonctions précédentes permettent d'obtenir le tracé suivant des 100 premiers termes de la suite $(u_n)_{n \geq 1}$.



Quelle conjecture peut-on émettre quant à la limite de $(u_n)_{n \geq 1}$?

Exercice 4 Un forain organise un jeu de fléchettes dans une fête foraine. Le jeu se présente sous la forme de trois cibles A, B et C. La cible A est séparée en deux secteurs, la cible B est séparée en quatre secteurs et la cible C est séparée en 8 secteurs. Sur chaque cible, les secteurs ont la même aire, ce qui signifie qu'un joueur qui lance une fléchette au hasard sur une cible donnée a les mêmes chances d'atteindre chaque secteur de cette cible.



Le jeu consiste à lancer des fléchettes sur les cibles successives selon le protocole suivant :

- On commence par la cible A. Il faut atteindre le secteur 1 de cette cible pour avoir le droit de passer à la cible B ; dans le cas contraire, on continue à lancer la fléchette en direction de la cible A.
- De même, lorsqu'on lance une fléchette en direction de la cible B, il faut atteindre le secteur 1 pour avoir le droit de passer à la cible C ; dans le cas contraire, on continue les lancers vers la cible B.
- Enfin, le joueur ne lance qu'une fois la fléchette en direction de la cible C. Le secteur qu'il atteint décide du lot qu'il gagne.

On suppose que le joueur atteint toujours la cible visée et que pour une cible donnée, chaque secteur est atteint avec la même probabilité. On suppose que le joueur continue de lancer des fléchettes autant de fois qu'il est nécessaire pour avoir le droit de passer à la cible C.

Soit $n \in \mathbb{N}^*$. On note A_n l'événement « lors du lancer de la n -ième fléchette, le joueur tire vers la cible A ». On définit de même l'événement B_n . On note enfin a_n et b_n les probabilités respectives de A_n et B_n . Le joueur commençant par la cible A, on a donc $a_1 = 1$ et $b_1 = 0$.

1. a. Calculer les probabilités a_2 et b_2 .

b. Vérifier que $b_3 = \frac{5}{8}$.

2. En utilisant la formule des probabilités totales, justifier que pour tout entier $n \geq 1$, on a

$$a_{n+1} = \frac{1}{2} a_n \quad \text{et} \quad b_{n+1} = \frac{3}{4} b_n + \frac{1}{2} a_n.$$

3. a. Recopier et compléter les lignes 3 et 4 du programme Python ci-dessous afin qu'il affiche les valeurs de a_n et de b_n , l'entier n supérieur ou égal à 1 étant donné par l'utilisateur.

```
1 a = 1 ; b = 0
2 for i in range(1, n):
3     b = ...
4     a = ...
5 print(a, b)
```

- b. Si on échange les lignes 3 et 4, le résultat affiché est-il le même ? Pourquoi ?

4. Montrer que pour tout entier n supérieur ou égal à 1, on a $a_n = \left(\frac{1}{2}\right)^{n-1}$.

5. Pour tout entier n supérieur ou égal à 1, on pose $v_n = 2^{n-1} b_n + 2$.

a. Montrer que $v_1 = 2$ et que pour tout entier n supérieur ou égal à 1, on a $v_{n+1} = \frac{3}{2} v_n$.

b. En déduire, pour tout entier n supérieur ou égal à 1, une expression de v_n en fonction de n .

c. Établir que pour tout entier n supérieur ou égal à 1, on a $b_n = 2 \times \frac{3^{n-1}}{4^{n-1}} - \frac{2}{2^{n-1}}$.

6. Soit X la variable aléatoire égale au nombre de fléchettes lancées par le joueur. Recopier et compléter les lignes 12, 13 et 14 du programme ci-dessous afin qu'il simule l'expérience et affiche la valeur de X .

```
1 import numpy.random as rd
2
```

```
3 cible = "a"
4 n = 0
5 while cible != "c":
6     n = n+1
7     if cible == "a":
8         secteur = rd.randint(1,3)
9         if secteur == 1:
10             cible = "b"
11     elif cible == "b":
12         secteur = ...
13         if ...:
14             ...
15 print(n)
```

7. Il faut payer 1 euro pour chaque fléchette lancée. Les joueurs se découragent vite. Le forain constate qu'en fait, chaque joueur effectue au moins deux lancers et s'il n'obtient pas le droit de tirer sur la cible C après les deux premiers lancers, alors il abandonne le jeu. Les deux euros qu'il a donnés sont alors perdus.
- Justifier que la probabilité pour un joueur donné de gagner un lot dans ces conditions est $\frac{1}{8}$.
 - Vingt joueurs se présentent successivement et jouent selon ce principe. On suppose que les résultats de chaque joueur sont indépendants les uns des autres. Soit Y la variable aléatoire égale au nombre de joueurs qui gagnent un lot. Reconnaitre la loi de Y . Décrire l'ensemble $Y(\Omega)$ des valeurs prises par Y et donner l'expression de $P([Y = k])$ pour tout entier k appartenant à cet ensemble.
 - Calculer l'espérance et la variance de Y .
 - Chaque lot offert par le forain est d'une valeur de 5 euros. Une fois que les 20 joueurs ont tenté leur chance, on note G la variable aléatoire égale au gain en euros du forain, G étant négatif si le forain perd de l'argent. Justifier que $G = 2(20 - Y) - 2Y$. Calculer le gain moyen du forain.