

Exercice 1.**Partie A : Étude d'une fonction et d'une somme.**

On pose, pour tout entier naturel n supérieur ou égal à 1 ,

$$S_n = \sum_{k=1}^n \frac{1}{k}$$

1. Étudier les variations de la fonction définie sur $[0; 1[$ par $f(x) = x + \ln(1 - x)$.

2. En déduire que

$$\forall x \in [0; 1[, \quad x \leq -\ln(1 - x)$$

puis que

$$\forall k \geq 2, \quad \frac{1}{k} \leq \ln(k) - \ln(k - 1)$$

3. En déduire la majoration

$$\forall n \in \mathbb{N}^*, \quad S_n \leq \ln(n) + 1$$

Partie B : Étude d'une suite.

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 1$ et la relation de récurrence $u_{n+1} = u_n + \frac{1}{u_n}$.

4. Montrer par récurrence que cette suite est bien définie et strictement positive.

5. En déduire le sens de variation de la suite $(u_n)_{n \in \mathbb{N}}$.

6. (a) Pour tout entier naturel k , exprimer $u_{k+1}^2 - u_k^2$ en fonction de u_k^2 .

(b) En déduire, à l'aide d'un raisonnement par récurrence, que

$$\forall n \in \mathbb{N}^*, \quad u_n^2 = 2n + 1 + \sum_{k=0}^{n-1} \frac{1}{u_k^2}$$

(c) Justifier que $\forall n \in \mathbb{N}^*, \quad u_n^2 \geq 2n + 1$. En déduire la limite de la suite $(u_n)_{n \in \mathbb{N}}$.

7. (a) À l'aide du résultat précédent, ainsi que la question 6. (b) montrer que pour tout entier $n \geq 2$,

$$u_n^2 \leq 2n + 2 + \frac{1}{2}S_{n-1}$$

où $S_{n-1} = \sum_{k=1}^{n-1} \frac{1}{k}$ est la somme étudiée dans la partie A.

(b) En utilisant un résultat de la partie A, montrer que pour tout entier $n \geq 2$,

$$u_n^2 \leq 2n + \frac{5}{2} + \frac{\ln(n-1)}{2}$$

8. En utilisant les résultats des questions précédentes, montrer l'équivalent $u_n \sim \sqrt{2n}$ lorsque n tend vers l'infini.

9. Étude informatique

(a) Proposer un script Python permettant de calculer la valeur u_n , pour une valeur de n donnée par l'utilisateur.

(b) Proposer un script Python permettant de calculer la plus petite valeur n pour laquelle $u_n \geq 100$.

Exercice 2.

Pour chaque entier naturel n , on définit la fonction f_n par :

$$\forall x \in [n; +\infty[, f_n(x) = \int_n^x e^{\sqrt{t}} dt$$

1. Montrer que la fonction f_n est de classe $\mathcal{C}^1$ sur $[n; +\infty[$ et justifier que $f'_n(x) = e^{\sqrt{x}}$.

2. (a) Encadrer $e^{\sqrt{t}}$, lorsque t appartient à l'intervalle $[n; x]$.

(b) En déduire que

$$e^{\sqrt{n}}(x - n) \leq f_n(x) \leq e^{\sqrt{x}}(x - n)$$

(c) Conclure : quelle est la limite de $f_n(x)$ lorsque $x \rightarrow +\infty$?

3. Montrer que pour chaque entier naturel n , il existe un unique réel x dans l'intervalle $[n; +\infty[$ tel que $f_n(x) = 1$.

On notera u_n ce réel, et on s'intéresse désormais à la suite $(u_n)_{n \in \mathbb{N}}$ ainsi définie.

$$\text{On notera que } u_n \in [n, +\infty[\text{ et que } f_n(u_n) = 1.$$

4. Montrer que $\lim_{n \rightarrow +\infty} u_n = +\infty$.

5. En utilisant l'encadrement de la question 2. (b), montrer que

$$\forall n \in \mathbb{N}, e^{-\sqrt{u_n}} \leq u_n - n \leq e^{-\sqrt{n}}$$

6. (a) Compléter le script Python suivant afin de calculer un entier n pour lequel $u_n - n \leq 10^{-4}$.

```

1  n=0
2
3  while ...
4      n=
5
6  print(n)
```

(b) Une fois exécuté, le script affiche l'une des trois valeurs suivantes : 55, 70 ou 85.

Préciser laquelle en admettant la valeur approchée $\ln(10) \simeq 2,3$.

7. On pose à présent $v_n = u_n - n$ pour tout entier naturel n .

(a) Montrer que $\lim_{n \rightarrow +\infty} v_n = 0$. On pourra faire appel au résultat de la question 5.

(b) Établir que pour tout réel $x \in [-1; +\infty[$, on a :

$$\sqrt{1+x} \leq 1 + \frac{x}{2}$$

(c) Vérifier ensuite que

$$\forall n \in \mathbb{N}^*, e^{-\sqrt{u_n}} = e^{-\sqrt{n} \sqrt{1 + \frac{v_n}{n}}}$$

puis utiliser le résultat de la question 7. (b) pour prouver que

$$e^{-\sqrt{u_n}} \geq e^{-\sqrt{n}} \times e^{-\frac{v_n}{2\sqrt{n}}}$$

(d) En déduire, grâce à l'encadrement obtenu à la question 5. que $u_n - n \sim e^{-\sqrt{n}}$.

Problème.**Partie A : Réduction d'un endomorphisme.**

On considère la matrice :

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

1. Déterminer le rang de la matrice A . *On attend des explications.*
2. En déduire que la matrice A n'est pas inversible, ainsi qu'une valeur propre de A .
3. Calculer la matrice $A^3 - 4A$.
4. En déduire les valeurs propres possibles de la matrice A .
5. Déterminer les valeurs propres de A et les sous-espaces propres associés.
6. Justifier que la matrice A est diagonalisable.

Donner une matrice inversible P et une matrice diagonale D telles que :

- Les valeurs diagonales de la matrice D soient dans l'ordre croissant.
- La deuxième ligne de la matrice P soit $(1 \ 1 \ 1)$
- $A = PDP^{-1}$

7. On considère la matrice $Q = \begin{pmatrix} -1 & 1 & 1 \\ 0 & 2 & -2 \\ 1 & 1 & 1 \end{pmatrix}$. Calculer la matrice $P \times Q$ et en déduire P^{-1} .

8. Montrer par récurrence que, pour tout entier naturel k , on a $A^k = PD^kP^{-1}$.

9. Écrire, pour tout entier naturel k non nul, la première colonne de la matrice A^k .

Vérifier que ce résultat reste valable si $k = 0$.

Partie B : Étude d'une suite de variables aléatoires.

On considère trois urnes contenant trois boules chacune :

- L'urne numéro 1 contient trois boules numérotées 1, 2 et 3.
- L'urne numéro 2 contient deux boules numérotées 1 et une boule numérotée 2.
- L'urne numéro 3 contient deux boules numérotées 1 et une boule numérotée 3.

On effectue une série illimitée de tirages avec remise dans ces urnes avec la règle suivante :

- Le premier tirage s'effectue dans l'urne 1.
- Si à un tirage donné on obtient le numéro k (avec $k \in \llbracket 1; 3 \rrbracket$), alors le tirage suivant s'effectue dans l'urne k .

Enfin, pour tout entier naturel non nul n , on notera X_n le numéro obtenu au n -ième tirage.

Par exemple, si l'on tire la boule numéro 2 au premier tirage (dans l'urne 1), alors $X_1 = 2$ et le second tirage s'effectue dans l'urne numéro 2. Si à ce deuxième tirage on obtient le numéro 1, alors $X_2 = 1$ et le troisième tirage s'effectue dans l'urne 1.

10. Reconnaître la loi de X_1 .

11. Donner les valeurs des neuf probabilités conditionnelles $P_{(X_k=j)}(X_{k+1}=i)$ avec i et j dans $\{1, 2, 3\}$.

12. Pour tout entier naturel n , on note

$$U_n = \begin{pmatrix} P(X_n = 1) \\ P(X_n = 2) \\ P(X_n = 3) \end{pmatrix}$$

- (a) On admet que $\{(X_k = 1), (X_k = 2), (X_k = 3)\}$ est un système complet d'événements.

Déterminer, grâce à la formule des probabilités totales, que

$$P(X_{n+1} = 1) = \frac{1}{3}P(X_n = 1) + \frac{2}{3}P(X_n = 2) + \frac{2}{3}P(X_n = 3)$$

- (b) De la même façon, exprimer $P(X_{n+1} = 2)$ et $P(X_{n+1} = 3)$ en fonction de $P(X_n = 1)$, $P(X_n = 2)$ et $P(X_n = 3)$.

- (c) Déterminer une matrice M carrée d'ordre 3, telle que pour tout entier naturel n non nul, on a $U_{n+1} = MU_n$.

- (d) Montrer qu'en posant $U_0 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}$, alors, pour tout n de $\mathbf{N}$, on a : $U_n = M^n U_0$.

- (e) Vérifier que $M = \frac{1}{3}A + \frac{1}{3}I$, puis établir que, pour tout n de $\mathbf{N}$, on a : $M^n = \left(\frac{1}{3}\right)^n \sum_{k=0}^n \binom{n}{k} A^k$.

(f) En déduire les 3 éléments de la première colonne de la matrice M^n , puis vérifier que la loi de X_n est donnée par :

$$\forall n \in \mathbf{N}^*, \quad P(X_n = 1) = \frac{1}{2} \left(1 + \left(-\frac{1}{3} \right)^n \right) \quad \text{et} \quad P(X_n = 2) = P(X_n = 3) = \frac{1}{4} \left(1 - \left(-\frac{1}{3} \right)^n \right).$$

13. Calculer l'espérance $E(X_n)$ de X_n .

14. Calculer $\lim_{n \rightarrow \infty} P(X_n = 1)$, $\lim_{n \rightarrow \infty} P(X_n = 2)$, $\lim_{n \rightarrow \infty} P(X_n = 3)$, et interpréter ces limites.

15. Compléter les parties manquantes du script Python suivant afin de simuler 100 tirages :

```

1  import random
2
3  X=random.randint(1,3)                ## Premier tirage
4
5  for k in ... :
6      if X==1:                          ## Tirage dans l'urne 1
7          r=random.random()
8          if r<1/3:
9              X=1
10         if 1/3<=r<=2/3:
11             X=2
12         if r > 2/3:
13             X=3
14     if X==2:                          ## Tirage dans l'urne 2
15         r=random.random()
16         if ... :
17             X= ...
18         if ... :
19             X= ...
20     if X==3:                          ## Tirage dans l'urne 3
21         r=random.random()
22         if ...
23             X= ...
24         if ... :
25             X= ...
26     print(X)

```

Indications : La commande `random.random()` permet de tirer un nombre aléatoire uniformément dans l'intervalle $[0;1]$, et la commande `random.randint(a, b)` tire une nombre entier uniformément entre a et b .

16. On exécute le script précédent, et on observe une succession aléatoire de "1", "2" et "3", ce qui est normal.

En notant soigneusement les occurrences de chacun de ces nombres, on constate que

- le chiffre "1" est affiché 50 fois
- le chiffre "2" est affiché 24 fois
- le chiffre "3" est affiché 26 fois

En quoi cette observation est-elle conforme à nos calculs ?