

Récapitulatif sur les calculs de suites, sommes et produits avec Python

1 Calcul des termes d'une suite et représentation graphique

Exercice 1.1 1. Définir en Python les suites suivantes :

- $a_n = \frac{n^2 + 3 + \ln(n)}{e^{-n} + \sqrt{n^4 + n^2}}$
- $b_n = \sqrt{n^2 + 2n} - \sqrt{2n^2 + 1}$
- $c_n = \ln(n^2 + \sqrt{n} + 1) - \ln(n - \sqrt{n} + 1)$.

2. Pour chacune de ces suites, calculer plusieurs de leurs termes et représenter graphiquement les 50 premiers termes.
3. Conjecturer leurs limites.
4. **A faire à la fin du TP** Prouver ces conjectures.

Exercice 1.2 On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 0$ et pour tout entier naturel n , $u_{n+1} = \cos(e^{u_n}) + \frac{u_n}{2}$.

1. Construire une fonction en Python qui, étant donné un entier $n \in \mathbb{N}$, calcule u_n .
2. Représenter graphiquement les 50 premiers termes de la suite.
3. Conjecturer le comportement asymptotique $(u_n)_{n \in \mathbb{N}}$.
4. *On admet que la suite $(u_n)_{n \in \mathbb{N}}$ converge vers une limite ℓ .* Construire une fonction en Python qui permet de déterminer le plus petit entier $n \in \mathbb{N}$ tel que $|u_n - \ell| \leq 10^{-3}$.

Exercice 1.3 On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par $u_0 = 0$ et pour tout entier naturel n , $u_{n+1} = u_n^2 + 1$.

1. Construire une fonction en Python qui, étant donné un entier $n \in \mathbb{N}$, calcule u_n .
2. **A faire à la fin du TP** Montrer que, pour tout entier naturel n , $u_n \geq n$.
3. En déduire le comportement asymptotique de $(u_n)_{n \in \mathbb{N}}$.
4. Construire une fonction en Python qui permet de déterminer le plus petit entier $n \in \mathbb{N}$ tel que $u_n \geq 10^5$.

Exercice 1.4 Soit $(u_n)_n$ la suite définie par :

$$\begin{cases} u_0 = 1, \\ \forall n \in \mathbb{N} \quad u_{n+1} = -2u_n + 3n + 2. \end{cases}$$

On pose : $\forall n \in \mathbb{N} \quad v_n = u_n - n - \frac{1}{3}$.

1. Ecrire une fonction Python qui prend en entrée un entier n et qui calcule les n premiers termes de la suite $(u_n)_n$.
2. Compléter la fonction précédente pour qu'elle calcule également les n premiers termes de la suite $(v_n)_n$.
3. De quelle nature semble la suite $(v_n)_n$?
4. **A faire à la fin du TP** Le démontrer.

2 Calcul de sommes et de produits

Exercice 2.1 1. Ecrire une fonction ayant comme argument d'entrée un entier $n \in \mathbb{N}^*$ et qui renvoie la valeur de

$$S_n = \sum_{k=1}^n \frac{2k+1}{k^2(k+1)^2}.$$

2. Ecrire une fonction ayant comme argument d'entrée un entier $n \in \mathbb{N}$ et qui renvoie la valeur de

$$T_n = \frac{n(n+2)}{(n+1)^2}.$$

3. Evaluer les deux fonctions précédentes pour différentes valeurs de n et émettre une conjecture sur T_n et S_n .

4. **A faire à la fin du TP** Montrer que pour tout $k \in \mathbb{N}^*$,

$$\frac{1}{k^2} - \frac{1}{(k+1)^2} = \frac{2k+1}{k^2(k+1)^2}.$$

5. **A faire à la fin du TP** Démontrer la conjecture.

Exercice 2.2 On suppose que les élèves d'ECG1 sont tous nés une année à 365 jours et on suppose que les jours de naissance sont répartis de façon équiprobable dans l'année.

1. Soit $n \in \llbracket 1; 365 \rrbracket$, montrer que dans une classe de n élèves d'ECG1, la probabilité qu'au moins deux élèves aient la même date de naissance est

$$p_n = 1 - \prod_{k=0}^{n-1} \left(1 - \frac{k}{365}\right)$$

2. Écrire une fonction Python qui prend en argument $n \in \llbracket 1; 365 \rrbracket$ et qui renvoie p_n .

3. (a) Vérifier que, pour tout $n \in \llbracket 1; 364 \rrbracket$,

$$p_{n+1} = 1 - (1 - p_n) \left(1 - \frac{n}{365}\right).$$

(b) Écrire alors une fonction Python qui prend en argument $x \in]0; 1[$ et qui calcule le plus petit $n \in \llbracket 1; 365 \rrbracket$ tel que $p_n \geq x$.

(c) La tester avec $x = 0,99$.