

Approfondissement

Suites numériques : généralités

Pour aller plus loin : Héron, Fibonacci, factorielle, sommes d'inverses

5 exercices · niveau ●●● · durée conseillée : 2 h 30 · calculatrice autorisée

Exercice 1 La méthode de Héron pour approcher $\sqrt{2}$



Héron d'Alexandrie (i^{er} siècle) approchait $\sqrt{2}$ ainsi : si u est une valeur approchée de $\sqrt{2}$, alors $\frac{2}{u}$ en est une autre, située « de l'autre côté » de $\sqrt{2}$, et leur moyenne est une meilleure approximation. On définit donc la suite (u_n) par $u_0 = 2$ et, pour tout entier naturel n , $u_{n+1} = \frac{1}{2} \left(u_n + \frac{2}{u_n} \right)$. On admet que tous ses termes sont strictement positifs.

1.1 Calculer u_1 , u_2 et u_3 sous forme de fractions irréductibles.

1.2 Montrer que, pour tout entier naturel n , $u_{n+1} - \sqrt{2} = \frac{(u_n - \sqrt{2})^2}{2u_n}$. En déduire que $u_n \geq \sqrt{2}$ pour tout n .

1.3 Montrer que $u_{n+1} - u_n = \frac{2 - u_n^2}{2u_n}$, puis que la suite (u_n) est décroissante.

1.4 Calculer $u_3^2 - 2$. En déduire que $0 \leq u_3 - \sqrt{2} \leq \frac{1}{408^2 \times 2}$ (on pourra factoriser $u_3^2 - 2$). Combien de décimales exactes de $\sqrt{2}$ la fraction u_3 fournit-elle ?

1.5 Écrire une fonction Python heron(a, n) qui renvoie le terme u_n de la suite définie par $u_0 = a$ et $u_{n+1} = \frac{1}{2} \left(u_n + \frac{a}{u_n} \right)$, pour un réel $a > 1$. Vers quel nombre la suite semble-t-elle tendre ?

Exercice 2 La suite de Fibonacci et le nombre d'or



La suite de Fibonacci est définie par $F_0 = 0$, $F_1 = 1$ et, pour tout $n \in \mathbb{N}$, $F_{n+2} = F_{n+1} + F_n$: chaque terme est la somme des deux précédents.

```

1 def fibo(n):
2     L = [0, 1]
3     for k in range(n - 1):
4         L.append(L[-1] + L[-2])
5     return L

```

2.1 Calculer F_2 à F_{12} . Que renvoie fibo(6) ? (L[-1] désigne le dernier élément de la liste L, et L[-2] l'avant-dernier.)

2.2 Montrer que la suite (F_n) est croissante à partir du rang 1.

2.3 Pour $n \geq 1$, on pose $r_n = \frac{F_{n+1}}{F_n}$. Calculer r_1 à r_{10} (valeurs arrondies à 10^{-4} pour les derniers). La suite (r_n) est-elle monotone ? Conjecturer sa limite.

2.4 Montrer que $r_{n+1} = 1 + \frac{1}{r_n}$. En admettant que (r_n) admet une limite φ et que l'on peut « passer à la limite » dans cette égalité, montrer que $\varphi^2 = \varphi + 1$, puis que $\varphi = \frac{1 + \sqrt{5}}{2}$ (le nombre d'or).

2.5 Vérifier que $F_{n+1}F_{n-1} - F_n^2 = (-1)^n$ pour $n = 1, \dots, 6$ (identité de Cassini ; sa démonstration pour tout n utilise un raisonnement par récurrence — hors programme ; on l'admet pour tout $n \geq 1$). En déduire que $|r_n - r_{n-1}| = \frac{1}{F_n F_{n-1}}$ pour $n \geq 2$ et interpréter.

Exercice 3 Factorielle contre puissance



Pour tout entier $n \geq 1$, on note $n! = 1 \times 2 \times \cdots \times n$ (lire « factorielle n ») et on pose $0! = 1$.

- 3.1 Calculer $n!$ pour n de 1 à 8. Justifier que $(n+1)! = (n+1) \times n!$: la suite $(n!)$ vérifie une relation de récurrence.
- 3.2 Écrire une fonction Python `fact(n)` qui renvoie $n!$ à l'aide d'une boucle `for`, puis une fonction qui renvoie le plus petit entier n tel que $n! > 10^6$, à l'aide d'une boucle `while`. Quel entier obtient-on ?
- 3.3 On pose $u_n = \frac{2^n}{n!}$ pour $n \in \mathbb{N}$. Montrer que $\frac{u_{n+1}}{u_n} = \frac{2}{n+1}$, puis que la suite (u_n) est décroissante à partir du rang 1.
- 3.4 Montrer que, pour tout $n \geq 2$, $\frac{u_{n+1}}{u_n} \leq \frac{2}{3}$. En déduire que $u_n \leq 2 \times \left(\frac{2}{3}\right)^{n-2}$ pour $n \geq 2$, puis conjecturer la limite de (u_n) . Qui « gagne » entre 2^n et $n!$?
- 3.5 Même étude pour $v_n = \frac{10^n}{n!}$: à partir de quel rang la suite (v_n) est-elle décroissante ? Que vaut son plus grand terme ?

Exercice 4 La somme des inverses des carrés



Pour tout entier $n \geq 1$, on pose $S_n = \frac{1}{1^2} + \frac{1}{2^2} + \cdots + \frac{1}{n^2}$.

- 4.1 Calculer S_1 , S_2 et S_3 . Exprimer S_{n+1} en fonction de S_n et en déduire le sens de variation de (S_n) .
- 4.2 Écrire une fonction Python `S(n)` qui renvoie S_n . Donner des valeurs approchées de S_{10} , S_{100} et S_{1000} .
- 4.3 Montrer que, pour tout entier $k \geq 2$, $\frac{1}{k^2} \leq \frac{1}{k-1} - \frac{1}{k}$.
- 4.4 En additionnant ces inégalités pour k allant de 2 à n , démontrer que $S_n \leq 2 - \frac{1}{n}$, donc que $S_n < 2$ pour tout $n \geq 1$.
- 4.5 Que peut-on conjecturer quant à la limite de (S_n) ? *Pour aller plus loin* : Euler a démontré en 1735 que cette limite vaut $\frac{\pi^2}{6}$. Comparer avec les valeurs de la question 4.2.

Exercice 5 La série harmonique : croissante, lente... mais pas bornée



Pour tout entier $n \geq 1$, on pose $H_n = 1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{n}$.

- 5.1 Justifier que la suite (H_n) est strictement croissante. Calculer H_1 , H_2 , H_3 et H_4 .
- 5.2 Montrer que, pour tout $n \geq 1$, $H_{2n} - H_n = \frac{1}{n+1} + \cdots + \frac{1}{2n} \geq n \times \frac{1}{2n} = \frac{1}{2}$.
- 5.3 En déduire que $H_2 \geq 1 + \frac{1}{2}$, $H_4 \geq 1 + \frac{2}{2}$, $H_8 \geq 1 + \frac{3}{2}$, puis que $H_{2^p} \geq 1 + \frac{p}{2}$ pour $p = 1, 2, \dots, 10$. Conjecturer la limite de (H_n) .
- 5.4 À l'aide d'un programme Python utilisant une boucle `while`, déterminer le plus petit entier n tel que $H_n > 5$, puis tel que $H_n > 10$. Commenter, en comparant avec l'exercice 4.