
Thème 1 : Suites numériques

I. Séance 1 : définitions

Exercice 1

Soit $(u_n)_{n \in \mathbb{N}}$ une suite de réels et soit $\ell \in \mathbb{R}$.

1. Qu'est-ce qu'une suite monotone ?
2. Traduire en mathématiques (avec les quantificateurs) les propositions mathématiques suivantes.
 - a) La suite $(u_n)_{n \in \mathbb{N}}$ est décroissante.
 - b) La suite $(u_n)_{n \in \mathbb{N}}$ est majorée.
 - c) La suite $(u_n)_{n \in \mathbb{N}}$ converge vers ℓ .
 - d) La suite $(u_n)_{n \in \mathbb{N}}$ diverge vers $+\infty$.
3. Reprendre les questions 2.a) et 2.b) dans le cas où les propriétés précédentes sont vérifiées seulement à partir d'un certain rang.
4. Écrire la négation des propositions de la question 2.

Exercice 2. (**) *Vrai ou Faux ?*

1. Si la suite (u_n) diverge vers $+\infty$, alors elle n'est pas majorée.
2. Une suite (u_n) croissante à partir d'un certain rang est minorée.
3. Si $(|u_n|)$ converge alors (u_n) converge.
4. Si $(|u_n|)$ tend vers 0 alors (u_n) tend vers 0.
5. Une suite convergente est monotone à partir d'un certain rang.
6. Une suite convergente et majorée est croissante.
7. Une suite divergeant vers $+\infty$ est croissante à partir d'un certain rang.
8. Une suite strictement croissante diverge vers $+\infty$.
9. Une suite strictement décroissante diverge vers $-\infty$.
10. Si (u_n) est croissante et pour tout $n \in \mathbb{N} : u_n \leq v_n$ alors (v_n) est croissante.
11. Si (u_n) tend vers 0 et (v_n) tend vers $+\infty$, alors on ne peut conclure sur la limite du quotient $\frac{u_n}{v_n}$.
12. Si (u_n) est divergente, alors la suite définie par : $\forall n \in \mathbb{N}, v_n = u_{n+1} - u_n$ est divergente.
13. Si (u_n) tend vers $\ell \neq 0$ alors : $\lim_{n \rightarrow +\infty} u_{n+1} - u_n = 0$ et $\lim_{n \rightarrow +\infty} \frac{u_{n+1}}{u_n} = 1$.

Séance 2 : manipulations de base (et Python)

Exercice 3

On considère la suite $(u_n)_{n \in \mathbb{N}}$ définie par :

$$\begin{cases} u_0 = 1 \\ \forall n \in \mathbb{N}, u_{n+1} = \frac{u_n}{1 + u_n^2} \end{cases}$$

1. Montrer que : $\forall n \in \mathbb{N}, u_n > 0$.
2. Montrer que la suite $(u_n)_{n \in \mathbb{N}}$ est monotone.
3. Étudier la convergence de la suite $(u_n)_{n \in \mathbb{N}}$.
4. a) Écrire en **Python** la fonction `calculSuite` qui prend en paramètre un entier `n` et renvoie le `n`^{ème} terme de la suite $(u_n)_{n \in \mathbb{N}}$.
b) Quel appel permet de calculer u_8 ?
5. a) Écrire en **Python** la fonction `calculPremiersTermes` qui prend en paramètre un entier `n` et renvoie la liste `L` contenant les `n` premiers termes de la suite $(u_n)_{n \in \mathbb{N}}$.
b) On considère alors le programme **Python** suivant :

```
1 import numpy as np
2 import matplotlib.pyplot as plt
3
4 n = 100
5 absc = np.linspace(0, n-1, n)
6 t = calculPremiersTermes(n)
7 plt.plot(absc, t)
8 plt.show()
```

Que réalise ce programme ?

6. a) Justifier qu'il existe un rang $n_0 \in \mathbb{N}$ tel que : $\forall n \geq n_0, |u_n| \leq 10^{-4}$.
b) Écrire en **Python** un programme qui permet de déterminer le premier entier n tel que $|u_n| \leq 10^{-4}$.

Séance 3 : suites récurrentes linéaires usuelles

Exercice 4

On considère une suite $(u_n)_{n \in \mathbb{N}}$ telle que $u_0 > 0$, $u_1 > 0$, et vérifiant la relation de récurrence :

$$\forall n \in \mathbb{N}, u_{n+2} = u_n^3 \times u_{n+1}^2$$

1. Montrer par une récurrence double que $u_n > 0$ pour tout entier naturel n .

2. On note $t_n = \ln(u_n)$.

a) Montrer que la suite (t_n) vérifie $\forall n \in \mathbb{N}, t_{n+2} = 3t_n + 2t_{n+1}$.

b) De quel type de suite s'agit-il ?

3. Déterminer, en fonction de u_0 et u_1 , le terme général de la suite (t_n) .

4. En déduire que :

$$\forall n \in \mathbb{N}, u_n = \exp \left(\frac{3^n}{4} \ln(u_0 u_1) + \frac{(-1)^n}{4} \ln \left(\frac{u_0^3}{u_1} \right) \right)$$

Exercice 5

On considère la suite (u_n) définie dans l'Exercice 4.

1. Écrire en **Python** la fonction `calculPremiersTermes` qui prend en paramètre un entier `n`, des valeurs `u0`, `u1` et renvoie la liste `L` contenant les `n` premiers termes de la suite $(u_n)_{n \in \mathbb{N}}$.

2. Écrire un programme permettant de tracer les 100 premiers termes de la suite (u_n) pour les valeurs $u_0 = 0,7$ et $u_1 = 1,2$.

Exercice 6

On appelle $(u_n)_{n \geq 0}$ la suite définie par :

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

1. Démontrer par récurrence : $\forall n \in \mathbb{N}, u_n \geq n$.

2. En déduire la limite de (u_n) quand n tend vers $+\infty$.

3. Déterminer la formule explicite de u_n .

Séance 4 : suites récurrentes et inégalités des accroissements finis

Exercice 7

Soit f la fonction définie sur $\mathbb{R}$ par $f : x \mapsto \frac{e^x}{e^{2x} + 1}$.

1. *a)* Démontrer que f est paire sur $\mathbb{R}$.

b) Justifier que f est $\mathcal{C}^1$ sur $\mathbb{R}$ et étudier ses variations.

c) Montrer que l'équation $f(x) = x$ admet une unique solution $\ell \in \mathbb{R}^+$.

d) Justifier que : $0 \leq \ell \leq \frac{1}{2}$.

Données numériques : $e^{1/2} \simeq 1,65$ et $e \simeq 2,72$ au centième près.

e) Montrer que : $\forall x \geq 0, |f'(x)| \leq f(x)$.

En déduire que : $\forall x \geq 0, |f'(x)| \leq \frac{1}{2}$.

f) Vérifier que $f([0, \frac{1}{2}]) \subset [0, \frac{1}{2}]$.

2. On définit la suite $(u_n)_{n \in \mathbb{N}}$ par :

$$u_0 = 0 \quad \text{et} \quad \forall n \in \mathbb{N}, u_{n+1} = f(u_n)$$

a) Montrer que, pour tout $n \in \mathbb{N}$, $u_n \in [0, \frac{1}{2}]$.

b) Montrer que, pour tout $n \in \mathbb{N}$:

$$|u_{n+1} - \ell| \leq \frac{1}{2} |u_n - \ell| \quad \text{puis que} \quad |u_n - \ell| \leq \frac{1}{2^{n+1}}$$

c) En déduire que la suite (u_n) converge vers ℓ .

3. Informatique

a) Écrire une fonction **Python** `f` qui prend en entrée un réel `x` et qui calcule $f(x)$.

b) En utilisant la fonction `f` précédente, écrire une fonction **SuiteU** qui prend en entrée un entier positif `n` et qui calcule u_n .

c) En utilisant la fonction **SuiteU** précédente, comment peut-on obtenir à l'aide de **Python** une valeur approchée de ℓ à 10^{-4} près ?

Exercice 8

Dans tout cet exercice, f désigne la fonction définie sur $]0, +\infty[$ par :

$$\forall x \in]0, +\infty[, f(x) = x - \ln(x)$$

Partie I : Étude de la fonction f

1. Dresser le tableau de variations de f en précisant ses limites en 0 et en $+\infty$.
2. Montrer que l'équation $f(x) = 2$, d'inconnue $x \in]0, +\infty[$, admet exactement deux solutions, que l'on note a et b , telles que $0 < a < 1 < b$.
3. Montrer : $b \in [2, 4]$. On donne : $\ln(2) \simeq 0,7$.

Partie II : Étude d'une suite

On pose : $u_0 = 4$ et $\forall n \in \mathbb{N}, u_{n+1} = \ln(u_n) + 2$.

4. Montrer que la suite $(u_n)_{n \in \mathbb{N}}$ est bien définie et que l'on a : $\forall n \in \mathbb{N}, u_n \in [b, +\infty[$.
5. Déterminer la monotonie de la suite $(u_n)_{n \in \mathbb{N}}$. En déduire qu'elle converge et préciser sa limite.
6. a) Montrer : $\forall n \in \mathbb{N}, u_{n+1} - b \leq \frac{1}{2}(u_n - b)$.
b) En déduire : $\forall n \in \mathbb{N}, 0 \leq u_n - b \leq \frac{1}{2^{n-1}}$.
7. a) Écrire une fonction **Python** d'en-tête `def suite(n)` : qui, prenant en argument un entier n de $\mathbb{N}$, renvoie la valeur de u_n .
b) Recopier et compléter la ligne 3 de la fonction **Python** suivante afin que, prenant en argument un réel `epsilon` strictement positif, elle renvoie une valeur approchée de b à `epsilon` près.

```
1 def valeur_approchee(epsilon) :  
2     n = 0  
3     while ..... :  
4         n = n + 1  
5     return suite(n)
```

Séance 5 : suites implicites

Exercice 9

Pour tout entier n positif, on définit sur $[0, +\infty[$ la fonction f_n par :

$$\forall x \in [0, +\infty[, f_n(x) = e^x + nx^2 - 3$$

1. a) Montrer que f_n est continue et dérivable sur son ensemble de définition.
Dresser son tableau de variations.
- b) Donner l'équation de la tangente de f_n en 1.
- c) Tracer dans un même repère les courbes de f_0 , f_1 et f_2 .
- d) Montrer que l'équation $f_n(x) = 0$ a exactement une solution positive, notée u_n .
- e) Préciser la valeur de u_0 . Dans la suite on supposera que $n \geq 1$.
- f) Vérifier que : $\forall n \in \mathbb{N}^*, u_n \in]0, 1[$.
2. Écrire une fonction **Python** qui prend un entier n et qui calcule une valeur approchée de u_n à 0,001 près par la méthode de dichotomie.
3. a) Montrer que : $\forall x \in]0, 1[, f_{n+1}(x) > f_n(x)$.
- b) En déduire le signe de $f_n(u_{n+1})$, puis le sens de variation de la suite (u_n) .
- c) Montrer que (u_n) est convergente. On note ℓ sa limite.
- d) On suppose dans cette question que $\ell > 0$.
Calculer la limite de $e^{u_n} + nu_n^2 - 3$ et en déduire une contradiction.
- e) Donner enfin la valeur de ℓ .
- f) Montrer que $\sqrt{\frac{n}{2}} u_n$ tend vers 1 quand n tend vers $+\infty$.

Exercice 10

Pour tout entier n non nul, on note h_n la fonction définie sur $\mathbb{R}_+^*$ par :

$$\forall x > 0, h_n(x) = f(x^n, 1) = x^n + 1 + \frac{1}{x^n}$$

1. Démontrer que pour tout entier naturel n non nul, la fonction h_n est strictement décroissante sur $]0, 1[$ et strictement croissante sur $[1, +\infty[$.
2. En déduire que pour tout entier n non nul, l'équation $h_n(x) = 4$ admet exactement deux solutions, notées u_n et v_n et vérifiant : $0 < u_n < 1 < v_n$.
3. a) Démontrer :

$$\forall x > 0, \forall n \in \mathbb{N}^*, h_{n+1}(x) - h_n(x) = \frac{(x-1)(x^{2n+1} - 1)}{x^{n+1}}$$
- b) En déduire : $\forall n \in \mathbb{N}^*, h_{n+1}(v_n) \geq 4$.
- c) Montrer alors que la suite (v_n) est décroissante.
4. a) Démontrer que la suite (v_n) converge vers un réel ℓ et montrer : $\ell \geq 1$.
- b) En supposant que $\ell > 1$, démontrer : $\lim_{n \rightarrow +\infty} v_n^n = +\infty$.
En déduire une contradiction.

c) Déterminer la limite de (v_n) .

5. a) Montrer : $\forall n \geq 1, v_n \leq 3$.

b) Écrire une fonction **Python** d'en-tête `def h(n,x)` qui renvoie la valeur de $h_n(x)$ lorsqu'on lui fournit un entier naturel n non nul et un réel $x \in \mathbb{R}_+$ en entrée.

c) Compléter la fonction suivante pour qu'elle renvoie une valeur approchée à 10^{-5} près de v_n par la méthode de dichotomie lorsqu'on lui fournit un entier $n \geq 1$ en entrée :

```
1  def v(n) :  
2      a = 1  
3      b = 3  
4      while (b-a) > 10**(-5) :  
5          c = (a + b)/2  
6          if h(n,c) < 4 :  
7              .....  
8          else :  
9              .....  
10     return .....
```