MATH93

Première Spécialité Mathématiques
Cours complet – méthode – démonstrations exigibles

COURS

Chapitre 2

Suites numériques – Partie 1 : généralités

Version professeur

Sommaire du chapitre

I Des listes Python aux suites

I.1 Une liste ordonnée de nombres

Des listes aux suites

Les suites numériques apparaissent depuis longtemps dans des problèmes de comptage, d'approximation ou d'évolution. Aujourd'hui, les listes en Python offrent une manière très concrète de manipuler une succession ordonnée de nombres avant de passer à la notation mathématique.

Objectifs de la partie 1

- Passer d'une liste Python à la notation d'une suite et inversement.
- Calculer des termes d'une suite définie explicitement, par récurrence, par un algorithme ou par un motif.
- Représenter graphiquement une suite.
- Étudier le sens de variation d'une suite dans des cas simples.
- Reconnaître une suite majorée, minorée ou bornée.
- Utiliser Python pour générer des termes et rechercher un seuil.
- Conjecturer, sur des exemples simples, une limite éventuelle.

Une liste Python *(list)*

Une liste Python est une collection ordonnée d'objets. Les indices commencent à 0.

```
u = [3, 7, 11, 15, 19]
print(u[0])   # premier élément : 3
print(u[2])   # troisième élément : 11
```

`u = [3, 7, 11, 15, 19]` → indices → $u_0 = 3,\ u_1 = 7,\ u_2 = 11, \ldots$

Indice Python et indice mathématique

La correspondance est naturelle : l'élément d'indice n de la liste joue le rôle du terme u_n de la suite. Ainsi `u[3]` correspond à u_3.

I.2 Construire les premiers termes avec Python

Une première suite avec une liste en compréhension

On considère les nombres définis par $u(n) = 2n + 3$ pour $n \in \mathbb{N}$.

```
u = [2*n + 3 for n in range(6)]
print(u)
```

Le programme affiche

$$[3, 5, 7, 9, 11, 13].$$

Ainsi $u_0 = 3$, $u_1 = 5$, $u_2 = 7$, etc.

Trois manières de générer une liste

Le programme de Première demande de savoir produire des listes :

- en extension : `[3, 5, 7, 9]` ;
- par ajouts successifs avec `append` ;
- en compréhension : `[2*n+3 for n in range(6)]`.

II Définitions et modes de génération

II.1 Définition et notations

Définition 1 (Suite numérique)

Une suite numérique (sequence) est une fonction définie sur $\mathbb{N}$, ou sur une partie de $\mathbb{N}$, et à valeurs dans $\mathbb{R}$.
On peut d'abord la noter sous forme fonctionnelle

$$u : n \longmapsto u(n),$$

puis utiliser la notation indicielle

$$u_n = u(n).$$

La suite se note $(u_n)_{n\in\mathbb{N}}$, ou simplement (u_n) lorsqu'il n'y a pas d'ambiguïté.

Vocabulaire

- u_n est le terme d'indice $\boldsymbol{n}$ (term of index n) ;
- u_{n+1} est le terme suivant (next term) de u_n ;
- u_{n-1} est le terme précédent (previous term) de u_n si $n \geqslant 1$;
- une suite peut commencer à un rang n_0 différent de 0.

II.2 Suite définie explicitement

Définition 2 (Définition explicite)

Une suite est définie explicitement (explicit formula) lorsque son terme u_n est donné directement en fonction de n :

$$\boxed{u_n = f(n)}.$$

Calcul de termes

Soit (u_n) définie pour $n \geqslant 2$ par

$$u_n = \frac{1}{n-1}.$$

$$u_2 = 1, \qquad u_3 = \frac{1}{2}, \qquad u_4 = \frac{1}{3}, \qquad u_5 = \frac{1}{4}.$$

II.3 Suite définie par récurrence

Définition 3 (Relation de récurrence)

Une suite est définie par une relation de récurrence (recurrence relation) lorsque chaque terme est obtenu à partir d'un ou de plusieurs termes précédents. Il faut alors préciser un terme initial.
Par exemple :

$$\begin{cases} u_0 = 2, \\ u_{n+1} = 3u_n - 1. \end{cases}$$

Calculer les premiers termes

Pour la suite précédente :

$$u_1 = 3 \times 2 - 1 = 5,$$
$$u_2 = 3 \times 5 - 1 = 14,$$
$$u_3 = 3 \times 14 - 1 = 41.$$

Même suite, langage naturel et Python

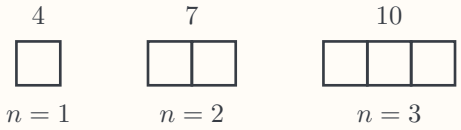
La relation $u_{n+1} = 3u_n - 1$ signifie : « multiplier le terme précédent par 3, puis soustraire 1 ».

```
u = 2
L = [u]
for n in range(5):
    u = 3*u - 1
    L.append(u)
print(L)
```

II.4 Une suite donnée par un motif

Un motif de carrés

On aligne des carrés construits avec des allumettes. On note u_n le nombre d'allumettes nécessaires pour construire n carrés alignés.



On lit $u_1 = 4$, $u_2 = 7$, $u_3 = 10$. À chaque nouveau carré, on ajoute 3 allumettes. Ainsi

$$u_{n+1} = u_n + 3 \quad \text{et} \quad u_n = 3n + 1.$$

III Représentation graphique d'une suite

Définition 4 (Représentation graphique)

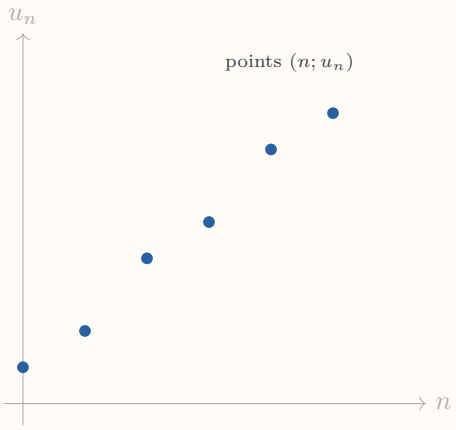
La représentation graphique d'une suite (u_n) est l'ensemble des points de coordonnées

$$\boxed{(n; u_n)}.$$

Contrairement à la courbe d'une fonction définie sur un intervalle réel, ces points sont isolés.

Lecture d'une représentation

La suite suivante est représentée par quelques-uns de ses termes.



On peut lire graphiquement, par exemple, $u_2 = 4$ et $u_5 = 8$.

Registres à savoir relier

Pour une même suite, il faut savoir passer du langage naturel à une formule, un algorithme, une liste de valeurs ou une représentation graphique.

IV Sens de variation d'une suite

IV.1 Définitions

Définition 5 (Suite croissante ou décroissante)

Soit (u_n) une suite définie à partir d'un certain rang.

- (u_n) est croissante (increasing) si, pour tout n, $u_{n+1} \geqslant u_n$;
- (u_n) est décroissante (decreasing) si, pour tout n, $u_{n+1} \leqslant u_n$;
- elle est constante (constant) si, pour tout n, $u_{n+1} = u_n$.

IV.2 Suites majorées, minorées et bornées

Définition 6 (Majoration et minoration)

Soit (u_n) une suite définie à partir d'un certain rang.

- (u_n) est majorée s'il existe un réel M tel que, pour tout indice n, $u_n \leqslant M$;
- (u_n) est minorée s'il existe un réel m tel que, pour tout indice n, $u_n \geqslant m$;
- (u_n) est bornée si elle est à la fois majorée et minorée.

Les nombres M et m sont respectivement appelés un majorant et un minorant de la suite. Ils ne sont pas nécessairement des termes de la suite.

Montrer qu'une suite est bornée

Pour tout $n \in \mathbb{N}$, on considère

$$u_n = \frac{n+1}{n+2}.$$

Comme $n+1 > 0$ et $n+2 > 0$, on a $u_n > 0$. De plus,

$$1 - u_n = 1 - \frac{n+1}{n+2} = \frac{1}{n+2} > 0.$$

Ainsi, pour tout $n \in \mathbb{N}$,

$$\boxed{0 < u_n < 1}.$$

La suite (u_n) est donc minorée par 0, majorée par 1, et par conséquent bornée.

IV.3 Étudier le signe de $u_{n+1} - u_n$

Méthode fondamentale

Pour étudier le sens de variation d'une suite, on peut calculer

$$u_{n+1} - u_n.$$

- si $u_{n+1} - u_n \geqslant 0$, la suite est croissante ;
- si $u_{n+1} - u_n \leqslant 0$, la suite est décroissante.

Une suite explicite

Soit $u_n = n^2 - 3n + 1$ pour $n \in \mathbb{N}$.

$$\begin{aligned} u_{n+1} - u_n &= (n+1)^2 - 3(n+1) + 1 - (n^2 - 3n + 1) \\ &= 2n - 2. \end{aligned}$$

Ainsi, pour $n \geqslant 1$, $u_{n+1} - u_n \geqslant 0$. La suite est croissante à partir du rang 1.

Suite explicite et fonction associée

Lorsque $u_n = f(n)$, l'étude des variations de la fonction f sur $[0; +\infty[$ peut aussi donner le sens de variation de la suite. Cette méthode sera particulièrement utile après le chapitre sur la dérivation.

V Python : calculer des termes et chercher un seuil

V.1 Calculer les termes d'une suite récurrente

Boucle `for` *(for loop)*

Lorsque le nombre d'étapes est connu, une boucle `for` permet de calculer les termes successifs.

```
u = 5
for n in range(8):
    u = 1.2*u + 3
print(u)
```

Après huit passages dans la boucle, la variable `u` contient le terme u_8 : le programme affiche donc u_8.

V.2 Rechercher un seuil

Définition 7 (Seuil)

Un seuil (threshold) est une valeur que l'on souhaite atteindre ou dépasser. Une boucle `while` est adaptée lorsque l'on ne connaît pas à l'avance le nombre d'itérations nécessaires.

Premier rang dépassant 1000

On considère $u_0 = 100$ et $u_{n+1} = 1,08u_n$. On cherche le premier rang n pour lequel $u_n > 1000$.

```
u = 100
n = 0
while u <= 1000:
    u = 1.08*u
    n = n + 1
print(n, u)
```

Le programme renvoie le premier rang n pour lequel $u_n > 1000$.

À savoir lire et écrire

Dans ce chapitre, on doit être capable de comprendre ou d'écrire des algorithmes simples permettant de calculer des termes d'une suite ou de rechercher un seuil.

VI Première intuition de la notion de limite

Observer le comportement pour de grands indices

En Première, on ne donne pas de définition formelle de la limite d'une suite. On cherche seulement à conjecturer son comportement lorsque n devient très grand, à partir de calculs, de graphiques ou d'algorithmes.

Vers une limite finie

$$u_n = 2 + \frac{1}{n} \quad (n \geqslant 1).$$

Lorsque n devient grand, $\frac{1}{n}$ devient très petit. On conjecture que u_n se rapproche de 2.

Vers $+\infty$

$$v_n = n^2.$$

Les termes deviennent arbitrairement grands. On dit intuitivement que la suite tend vers $+\infty$.

Pas de limite

$$w_n = (-1)^n.$$

Les termes alternent entre 1 et -1 : la suite ne se rapproche d'aucun réel et ne tend pas vers l'infini.

Le rôle de Python

Un programme de seuil permet d'explorer numériquement une conjecture de limite. Il ne constitue pas, à lui seul, une démonstration mathématique.

VII Bilan – Suites : généralités

1. Une suite numérique

$$u : n \longmapsto u(n), \qquad u_n = u(n).$$

- u_n : terme d'indice n (term of index n) ;
- bien repérer le rang initial.

2. Générer une suite

- forme explicite : $u_n = f(n)$;
- récurrence : $u_{n+1} = f(u_n)$ avec un terme initial ;
- algorithme / liste Python ;
- motif géométrique ou combinatoire.

3. Représenter une suite

$$\boxed{M_n(n; u_n)}$$

La représentation est un ensemble de points isolés.

4. Étudier les variations Une méthode classique consiste à étudier le signe de

$$\boxed{u_{n+1} - u_n}.$$

- $u_{n+1} - u_n \geqslant 0$: suite croissante ;
- $u_{n+1} - u_n \leqslant 0$: suite décroissante.

Une suite est bornée lorsqu'elle possède à la fois un minorant et un majorant.

5. Python

- liste : `[...]` ;
- `range` : produire des indices ;
- `append` : ajouter un terme ;
- boucle `for` : nombre d'étapes connu ;
- boucle `while` : recherche d'un seuil.

6. Limite : intuition seulement Pour de grands indices, on peut conjecturer :

- une limite finie ;
- une croissance vers $+\infty$ ou $-\infty$;
- une absence de limite.

Réflexes

- repérer le rang de départ ;
- distinguer explicite et récurrence ;
- ne pas confondre u_n et u_{n+1} ;
- pour les variations, comparer deux termes consécutifs.

Lexique essentiel

- suite : (sequence) ; terme : (term) ;
- indice : (index) ;
- relation de récurrence : (recurrence relation) ;
- croissante : (increasing) ; décroissante : (decreasing) ;
- seuil : (threshold).

↩ Fin du cours ↪