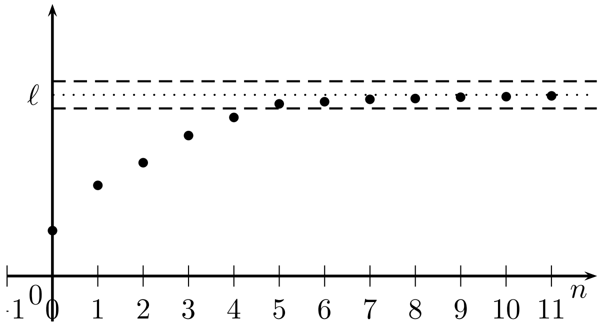
Chapitre 8 : Suites numériques – Notion de limite

I Notion de limite

I.1 Suite convergeant vers un nombre réel ℓ

Définition
Soient (u_n) une suite numérique et ℓ un nombre réel.
On dit que (u_n) admet pour limite ℓ (ou converge vers ℓ) lorsque tout intervalle ouvert contenant ℓ (aussi petit soit-il) contient tous les termes de la suite à partir d'un certain rang.
On note alors $\lim u_n = \ell$.

Illustration :
Le graphique ci-dessous représente une suite (u_n) qui converge vers $\ell = 4$.
On a illustré en ordonnée un intervalle ouvert contenant 4 : $]3,7;4,3[$.



Remarque
$\lim u_n = \ell$ signifie que tous les termes de la suite u_n deviennent "infiniment proches" de ℓ lorque n devient très grand.
Il est inutile de préciser $n \to +\infty$ car c'est toujours le cas dans ce chapitre.
On note simplement $\lim u_n = \ell$ pour désigner $\lim\limits_{n\to+\infty} u_n = \ell$.

Théorème (admis)
Si une suite est convergente, sa limite est unique.

Exemple

Expression de u_n	$\dfrac{1}{n}$	$\dfrac{1}{\sqrt{n}}$	$\dfrac{1}{n^k}$, $k \geqslant 1$	$-3 + \dfrac{2}{n}$	
Limite de la suite ($n \to +\infty$)					4

Remarque
Il existe des suites qui ne sont pas convergentes (on dit alors divergentes).
La suite définie par $u_n = (-1)^n$ n'a pas de limite, elle est divergente.

I.2 Suites ayant une limite infinie

Définition
On dit que la suite (u_n) a pour limite $+\infty$ (ou diverge vers $+\infty$) si tout intervalle du type $[A;+\infty[$ contient tous les termes de la suite à partir d'un certain rang.
On note alors $\lim u_n = +\infty$.
On dit que la suite (u_n) a pour limite $-\infty$ (ou diverge vers $-\infty$) si tout intervalle du type $]-\infty;A]$ contient tous les termes de la suite à partir d'un certain rang.
On note alors $\lim u_n = -\infty$.

Exemple :
$\lim n^2 = \ldots$; $\lim n^3 = \ldots$; $\lim \sqrt{n} = \ldots$, et $\lim(\ldots\ldots) = -\infty$

II Recherche de seuil

II.1 Exemple avec une suite divergeant vers $+\infty$

Exercice 1
Considérons la suite (u_n) définie par $u_n = n^2$.
On sait que (u_n) est croissante (car la fonction $x \mapsto x^2$ est croissante sur $[0;+\infty[$).
De plus, en calculant quelques termes ($u_{100} = 10000$, $u_{1000} = 1000000 = 10^6$), on peut conjecturer qu'elle diverge vers $+\infty$.
On cherche le plus petit entier n tel que $n^2 \geqslant 10^7$.
On utilise un algorithme de seuil. Compléter.

Langage naturel	Fonction Python associée
n `prend la valeur 0`	`def seuil():`
U `prend la valeur` n^2	.
`Tant que ...,`	.
n `prend la valeur` $n+1$	.
U `prend la valeur` n^2	.
`Fin Tant que`	.
`Afficher` n	.

On obtient $N = \ldots$.
Cela signifie que le plus petit entier n tel que $n^2 \geqslant 10^7$ est $n = \ldots$
En outre comme la suite (n^2) est croissante, on peut affirmer que ....
....
Les entiers n pour lesquels $n^2 \geqslant 10^7$ sont donc ....

Remarque
Sur cet exemple (simple), on pouvait aussi résoudre l'inéquation $n^2 \geqslant 10^7$.
Comme n est un entier positif, cela implique $n \geqslant \sqrt{10^7} \approx 3162,3$.
Le plus petit entier qui convient est donc 3163.

Commentaire
De façon générale, dans un **algorithme de seuil**, on utilise une **boucle non bornée** (Tant que), et la condition du test est la **négation** de la condition demandée dans la question.
Sur l'exemple précédent, on cherche le plus petit entier n tel que $n^2 \geqslant 10^7$, et dans l'algorithme, Tant que $U < 10^7$.

Exercice 2

1. La négation de $n < 10$ est ....
2. La négation de $n \geqslant 13$ est ....

Exercice 3
Considérons la suite (u_n) définie pour tout entier $n \in \mathbb{N}$ par $u_n = n \times \sqrt{n}$.

1. Montrer que (u_n) est croissante.
2. On admet que $\lim u_n = +\infty$.
   Écrire un algorithme qui renvoie le plus petit entier n_0 tel que pour tout $n \geqslant n_0$, $u_n \geqslant 10^4$.
3. Programmer l'algorithme à la calculatrice ou Python et donner la valeur de n_0.

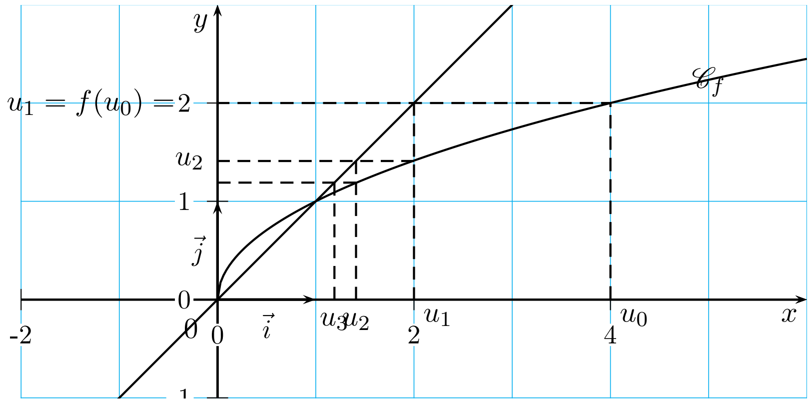
III Représentation des suites définies par récurrence $u_{n+1} = f(u_n)$

Cadre : on s'intéresse aux suites (u_n) définies par récurrence : $\begin{cases} u_0 \text{ donné} \\ u_{n+1} = f(u_n) \end{cases}$

III.0.a Premier exemple :

Soit (u_n) la suite définie par $\begin{cases} u_0 = 4 \\ u_{n+1} = \sqrt{u_n} \end{cases}$

Donc, $f(x) = \sqrt{x}$. On représente la fonction f et on trace la droite d'équation $y = x$. On place les termes de la suite (u_n) sur l'axe des abscisses.
La droite d'équation $y = x$ permet de reporter une même valeur de l'axe des ordonnées vers l'axe des abscisses.

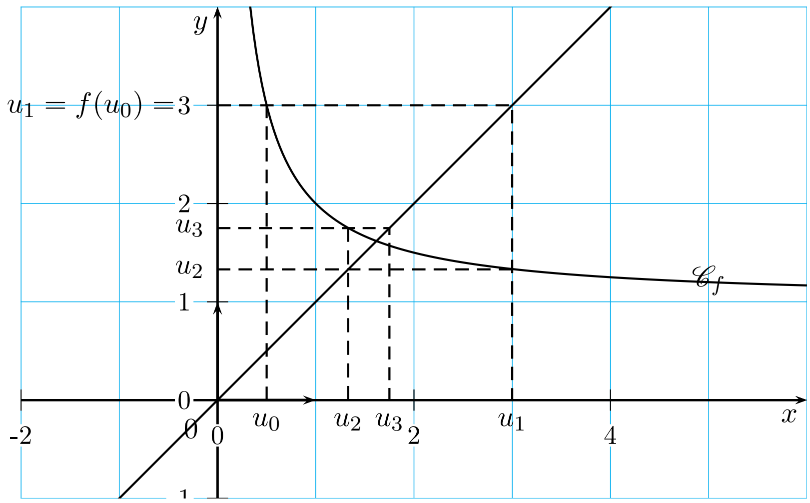


On peut conjecturer que la suite (u_n) est décroissante, bornée ($1 \leqslant u_n \leqslant 4$), et qu'elle converge vers 1.

III.0.b Deuxième exemple :

Soit (u_n) la suite définie par $\begin{cases} u_0 = \dfrac{1}{2} \\ u_{n+1} = 1 + \dfrac{1}{u_n} \end{cases}$.

Ici, $f(x) = 1 + \dfrac{1}{x}$.



On peut conjecturer que la suite (u_n) n'est ni croissante ni décroissante.

On retiendra deux allures classiques :
- — la forme en escalier lorsque f est croissante sur $[0; +\infty[$,
- — la forme en spirale lorsque f est décroissante sur $[0; +\infty[$.

IV Exercices portant sur des algorithmes : terme, somme, seuil

Exercice 4 (calcul de terme)
Soit (u_n) la suite définie sur $\mathbb{N}$ par $\begin{cases} u_0 = 5 \\ u_{n+1} = 2u_n + 6 \end{cases}$

1. Écrire un algorithme qui renvoie le nombre u_n lorsque l'on entre l'entier n.

Langage courant	Programme CASIO	Programme TEXAS
`Entrer N`	`"N=" ?→ N`	`Prompt N`
`U ← 5`	`5 → U`	`5 → U`
`Pour K allant de 1 à N`	`For 1 → K to N`	`For (K,1,N)`
`    U ← 2U+6`	`2U+6 → U`	`2U+6 → U`
`FinPour`	`Next`	`End`
`Afficher U`	`U ◢`	`Disp U`

2. Compléter la fonction Python d'argument n qui renvoie u_n pour un entier n donné.

```
def Terme(n) :
    u=...
    for k in range(...,...):
        ...
    return(u)
```

3. Programmer la fonction et donner u_{10}.
   On obtient $u_{10} = 11\,258$.

Exercice 5 (somme de termes)
On reprend la suite : $\begin{cases} u_0 = 5 \\ u_{n+1} = 2u_n + 6 \end{cases}$. Pour tout entier n donné en entrée ($n \geqslant 1$), on cherche à obtenir $S_n = u_0 + u_1 + \cdots + u_n = \displaystyle\sum_{k=0}^{n} u_k$.

1. On donne l'algorithme en langage courant. Compléter la fonction Python associée.

Langage courant

```
Entrer N
U ← 5
S ← U
Pour K allant de 1 à N
    U ← 2U+6
    S ← S+U
FinPour
Afficher S
```

Fonction Python

```
def Somme(n) :
    ............
    ............
    ............
    ............
    ............
    ............
```

2. Programmer et donner $S_{10} = u_0 + u_1 + \cdots + u_{10}$.
   On obtient $S_{10} = 22\,451$.

Exercice 6 (seuil avec une suite convergente)
On considère la suite (A_n) définie par $A_0 = 1$ et pour tout entier $n \geqslant 0$, $A_{n+1} = \dfrac{1}{3}A_n + 4$.

1. Calculer A_1 et A_2.
2. On admet que la suite (A_n) converge vers 6.
   (a) Écrire un algorithme qui renvoie le plus petit entier n_0 tel que $|A_{n_0} - 6| < 10^{-7}$.
   (b) Programmer l'algorithme à la calculatrice ou Python et indiquer la valeur de n_0.

Remarque
En seconde, on a vu que la distance entre deux nombres a et b est $|a - b| = |b - a|$.
L'inégalité $|A_n - 6| < 10^{-7}$ signifie que l'écart entre A_n et 6 est strictement inférieur à 10^{-7}.

Exercice 7 (variantes pour les sommes de termes)
Écrire une fonction Python qui détermine la somme S_n des premiers termes jusqu'à u_n inclus.

1. Pour tout entier naturel n, $u_n = 2 + 5n$.

   Pour tout $n \geqslant 0$, on pose $S_n = u_0 + u_1 + \cdots + u_n = \displaystyle\sum_{k=0}^{n} u_k$.

2. $\begin{cases} u_3 = -1 \\ \text{Pour tout } n \geqslant 3, u_{n+1} = -2u_n + 6 \end{cases}$

   On cherche à obtenir $S_n = u_3 + u_4 + \cdots + u_n = \displaystyle\sum_{k=3}^{n} u_k$ pour un entier $n \geqslant 4$ donné.

Exercice 8 (terme)
Soit (u_n) la suite définie par $u_0 = 2$, et pout tout $n \in \mathbb{N}$, $u_{n+1} = n - 2u_n$.

1. Calculer u_1, u_2.
2. Écrire un algorithme qui renvoie u_n pour un entier n donné en entrée ($n \geqslant 1$).
   Donner u_{20} et vérifier à l'aide du mode suite de la calculatrice.

Exercice 9 (somme)
Écrire un algorithme qui renvoie S_n pour $n \geqslant 1$ donné en entrée, puis donner la valeur de S_{10}.

1. Pour tout n, $S_n = u_0 + u_1 + \cdots + u_n$, où $u_0 = 3$ et $u_{n+1} = 2u_n - 11$.
2. Pour tout n, $S_n = u_0 + u_1 + \cdots + u_n$, où $u_n = 4n + 7$.
3. Pour tout n, $S_n = \displaystyle\sum_{k=0}^{n} \frac{1}{1+k^2}$.

Exercice 10 (seuil)
Considérons la suite (u_n) définie pour tout entier n par $u_n = n + 2\sqrt{n}$.

1. Montrer que (u_n) est croissante.
2. On admet que $\lim u_n = +\infty$.
   Écrire un algorithme qui renvoie le plus petit entier n_0 tel que pour tout $n \geqslant n_0$, $u_n \geqslant 10^3$ (voir livre page 115).
3. Programmer l'algorithme à la calculatrice et donner la valeur de n_0.

Exercice 11 (seuil)
Soit (a_n) la suite définie par son premier terme $a_0 = 2$ et la relation de récurrence :

$$\text{pour tout } n \geqslant 0,\ a_{n+1} = \frac{2}{3}a_n + 3.$$

1. Construire la droite d'équation $y = \frac{2}{3}x + 3$ et la droite d'équation $y = x$ (aller au moins jusqu'à 10 en abscisses et ordonnées).
2. Construire les premiers termes de la suite sur l'axe des abscisses.
3. Calculer à la main a_1, a_2 et a_3. Rédiger les calculs.
4. Utiliser la calculatrice pour donner une valeur approchée de a_{10}, a_{20}, et a_{30}. On arrondira à 0,0001 près.
5. Que peut-on conjecturer sur la convergence de la suite (a_n) ?
6. Écrire un algorithme qui renvoie le plus petit entier n_0 tel que $|a_{n_0} - 9| < 0,0001$.
7. Programmer cet algorithme à la calculatrice et indiquer la valeur de n_0.