Chapitre1

SUITES RÉCURRENTES ET IMPLICITES

1.1 Suites récurrentes $u_{n+1} = f(u_n)$

DÉFINITION 1.1

Une suite récurrence est une suite (u_n) définie par la donnée de son premier terme u_0 (ou u_1) et une relation de récurrence de la forme

$$u_{n+1} = f(u_n)$$

(où f est une certaine fonction), qui permet, de proche en proche, de calculer tous les termes de la suite.

1.1.1 Définition d'une suite récurrente

La plupart du temps, on montre par récurrence que (u_n) est bien définie notamment en montrant simultanément que ses termes se trouvent tous dans un intervalle inclus dans le domaine de définition de f.
Tout se passe bien lorsque l'on travaille sur un intervalle I **stable** sous l'action de f, et qu'on y prend le premier terme de la suite.

REMARQUE **1.1.**
Un intervalle I est dit stable par f si : $\forall x \in I, \quad f(x) \in I$.

A retenir

On fera donc bien attention, si le cas se présente, à ne pas oublier de montrer que u_n existe dans la récurrence.

EXERCICE 1.1. On considère la suite (u_n) définie par

$$u_0 = 1, \quad \text{et} \quad u_{n+1} = u_n + \frac{1}{u_n}$$

Montrer que, pour tout $n \in \mathbb{N}$, u_n est bien défini et que $u_n > 0$.

1.1.2 Obtenir un terme de rang quelconque avec Python

Écrire un code Python permettant de renvoyer le terme u_n d'une suite récurrente, où n est entré par l'utilisateur ou en argument de la fonction est une question classique, dont la réponse est facile.
Naturellement, tous les programmes de ce chapitre utilisent les bibliothèques et libraires usuelles

```
import numpy as np
import matplotlib.pyplot as plt
def suite_u(n):
    u=...... # initialisation : premier terme de la suite
    for k in range(1, n+1):
        u=f(u) # ou bien f est deja definie
               # ou bien remplacee directement par son expression
    return u
```

Une alternative serait de demander un programme qui renvoie la liste des termes successifs de la suite sous forme de liste

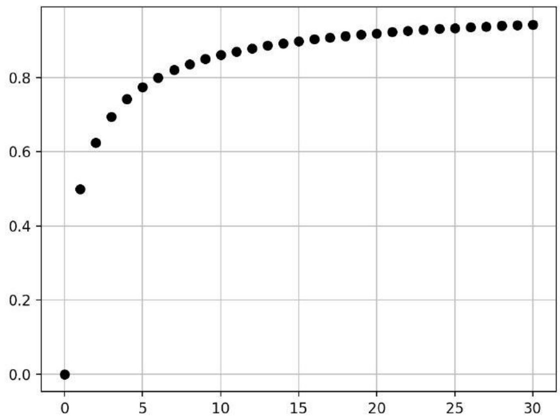
```
def termes_u(n):
    L=[]
    u=.... # initialisation : premier terme de la suite
    for k in range(n):
      L.append(u)
        u=f(u)
    return u
```

1.1.3 Représentation graphique

Avec Python

Pour représenter la même suite avec Python, on utilise la commande `plt.plot( )`. Plus précisément, on peut utiliser le code

```
def suite_u(n):
    u= 0
    for k in range(1, n+1):
        u=(u**2+1)/2
    return u
stop=30 # rang du dernier terme represente
U=[suite_u(k) for k in range(stop+1)]
N = [ k  for k in range(stop+1)]
plt.grid() # plus pratique pour la lecture
plt.plot(N, U, 'ko')
plt.show()
```



REMARQUE **1.2.**
On peut ainsi aisément formuler des conjectures sur cette suite : elle est visiblement strictement croissante, positive et elle semble converger vers 1. Mais tout ceci reste à prouver !

1.1.4 Variations d'une suite récurrente

A retenir

Il est faut de dire que la suite (u_n) suit les mêmes variations que la fonction f !
Une fonction f croissante peut générer une suite (u_n) décroissante !!

$\rightsquigarrow$ Une fonction f croissante (sur l'intervalle où vivent les termes) va permettre de générer une suite **monotone** mais qui peut être décroissante.

REMARQUE **1.3.**
L'étude de la monotonie peut se faire par deux méthodes.
Dans certains exercices, on aura le choix mais le plus souvent c'est l'énoncé du sujet qui guide via les questions posées.

Méthode 1: Par récurrence avec la croissance de f

Cette méthode nécessite d'être en mesure de calculer les deux premiers termes de la suite, ce qui n'est pas toujours le cas si par exemple u_0 est donné arbitraire dans un certain intervalle.
Soient (u_n) une suite récurrente, $u_{n+1} = f(u_n)$, avec f croissante sur I et, pour tout $n \in \mathbb{N}, u_n \in I$. Alors,
• Si $u_0 \leqslant u_1, (u_n)$ est croissante;
• Si $u_0 \geqslant u_1, (u_n)$ est décroissante.
Les conditions (i) ou (ii) permettent d'initialiser la récurrence, dont l'hérédité est triviale du fait de la croissance de f qui préserve les inégalités de l'hypothèse de récurrence.

EXERCICE 1.2. On continue avec (u_n) définie par $u_0 = 0$ et, pour $n \in \mathbb{N}$, $u_{n+1} = \dfrac{u_n^2+1}{2}$.

1. Montrer que, pour tout $n \in \mathbb{N}, u_n \in [0;1]$.

2. Montrer que (u_n) est croissante.

REMARQUE **1.4.**
Si la fonction f est décroissante (sur l'intervalle où vivent les termes de la suite) la suite (u_n) n'est plus monotone.
On peut en revanche montrer par la même méthode (une récurrence) que les suites (u_{2n}) et (u_{2n+1}) le sont. En comparant u_0 et u_2 pour la première, et u_1 et u_3 pour la seconde, on a le sens de chacune. Ceci repose sur l'observation que $g = f \circ f$ est dans ce cas croissante et que $u_{2(n+1)} = g(u_{2n})$ et $u_{2(n+1)+1} = g(u_{2n+1})$

$$\begin{tikzcd} u_n \arrow[r, bend left, "f"] \arrow[rr, bend right, "f \circ f = g"'] & u_{n+1} \arrow[r, bend left, "f"] & u_{n+2} \end{tikzcd}$$

EXERCICE 1.3. Soit (u_n) définie par $u_0 = 1$ et, pour $n \in \mathbb{N}, u_{n+1} = f(u_n)$ avec $f(x) = 1 + \frac{2}{x}$.
(1) Montrer que, pour tout $n \in \mathbb{N}, u_n$ est bien défini et que $u_n \in [1;3]$.
(2) Calculer u_1, u_2, u_3. La suite (u_n) est-elle monotone?
(3) Montrer que (u_{2n}) est croissante et que (u_{2n+1}) est décroissante.
On pourra montrer que $u_{2(n+1)} = g(u_{2n})$, où $g = f \circ f$ est croissante.

Méthode 2 : Avec le signe de $f(x)-x$

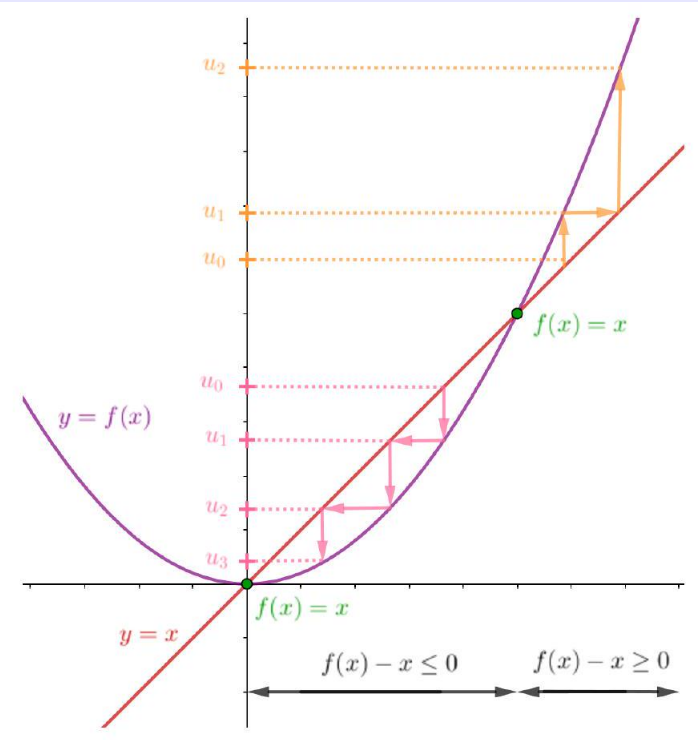
Observant que $u_{n+1}-u_n = f(u_n)-u_n$, connaître le signe de $f(x)-x$ permet tout de suite, (en évaluant donc en $x = u_n$) de connaître les variations de la suite, si bien sûr on sait que **tous les termes de la suite de situent dans un intervalle où le signe de $f(x)-x$ est constant.**
Plus précisément, on a le résultat suivant, dont la preuve est immédiate.

Soit (u_n) une suite définie par la relation de récurrence $u_{n+1} = f(u_n)$. Alors,
• Si, pour tout $n \in \mathbb{N}$, $u_n \in I$ et si $f(x)-x \geqslant 0$ pour $x \in I$, alors (u_n) est croissante.
• Si, pour tout $n \in \mathbb{N}$, $u_n \in I$ et si $f(x)-x \leqslant 0$ pour $x \in I$, alors (u_n) est décroissante.

$\rightsquigarrow$ On insiste sur la nécessité de la validité des inégalités là où se trouvent tous les termes de la suite. Si le signe de $f(x)-x$ varie, c'est la position de u_0 par rapport au(x) point(s) fixe(s) qui détermine le sens de variation de (u_n).

Illustration

Une même fonction f croissante (ici sur $\mathbb{R}_+$) peut générer, selon le premier terme u_0, une suite (u_n) croissante ou décroissante.



1.1.5 Point fixe: candidat éventuel pour une limite finie

En déterminant le signe de $f(x) - x$ dans la méthode précédente, on résout notamment l'**équation de point fixe** $f(x) = x$, dont les solutions donnent les candidats pour une limite finie éventuelle.

L'existence des points fixes de f ne garantit en aucun cas la convergence de la suite (u_n), il faut un argument supplémentaire (bien souvent le théorème de convergence monotone) pour montrer la convergence et ensuite choisir le bon point fixe comme valeur pour cette limite.

Méthode 3 : dite du point fixe

Soit (u_n) une suite récurrente définie par $u_{n+1} = f(u_n)$. Si, pour tout $n \in \mathbb{N}$, $u_n \in I$, si f est **continue sur** I**, et si** (u_n) est convergente, alors par passage à la limite dans la relation $u_{n+1} = f(u_n)$, la limite $\ell \in I$ est un point fixe f dans I.

$$\begin{array}{ccc} u_{n+1} & = & f(u_n) \\ \downarrow & n \to +\infty & \downarrow \\ \ell & = & f(\ell) \end{array}$$

REMARQUE **1.5.**
Si les seuls points fixes de f sont à l'extérieur de l'intervalle où vivent les termes de la suite (voire s'il n'y a pas de point fixe), on peut alors conclure (par un raisonnement par l'absurde) que la suite diverge!

1.1.6 Critère de convergence: convergence monotone

THÉORÈME 1.2 (**Théorème de convergence monotone**)

Soit (u_n) une suite. Alors,

- Si (u_n) est croissante et majorée, elle est convergente.
- Si (u_n) est croissante et non majorée, elle diverge vers $+\infty$.
- Si (u_n) est décroissante et minorée, elle est convergente.
- Si (u_n) est décroissante et non minorée, elle diverge vers $-\infty$.

Attention la convergence monotone ne donne pas la valeur de limite qui n'est souvent pas le majorant (ou le minorant) qu'on a utilisé. On évitera les conclusions hâtives.

REMARQUE **1.6.**
On utilise aussi ce théorème pour des démonstrations par l'absurde.
En effet, il est difficile de montrer par exemple qu'une suite n'est pas majorée. En supposant par l'absurde qu'elle l'est (lorsqu'on sait que la suite est croissante), on en déduit alors la convergence vers un point fixe puis on abouti à une contradiction (par exemple s'il n'y a pas de point fixe dans l'intervalle où vivent les termes de la suite).

EXERCICE 1.4. Montrer que la suite de l'Exercice 2 converge vers une limite ℓ à préciser.

Dans le cas où f est décroissante, on peut être amené à appliquer le théorème de convergence monotone aux suites (u_{2n}) et (u_{2n+1}). Si celles-ci convergent vers une même limite, alors il en est de même pour (u_n).

EXERCICE 1.5. Montrer que les suites (u_{2n}) et (u_{2n+1}) de l'Exercice 3 convergent et préciser leurs limites respectives. Conclure.

1.1.7 Utilisation de l'IAF

THÉORÈME 1.3 (**Inégalité des accroissements finis**)

Soit f une fonction de classe $\mathscr{C}^1$ sur un intervalle $[a,b]$. Alors, pour tous $x, y \in [a,b]$, on a :

$$|f(x) - f(y)| \leqslant \max_{t\in[a,b]} \left|f'(t)\right| |x-y|$$

REMARQUE **1.7.**
Ainsi, lorsque la suite n'est pas monotone (mais pas seulement dans ce cas), l'utilisation de l'inégalité des accroissements finis (IAF) peut permettre d'obtenir des estimations des **écarts successifs** entre les termes de la suite et le candidat limite (aussi point fixe de f).
Une récurrence permet ensuite d'obtenir un encadrement donnant la conclusion souhaitée (la convergence) par application du théorème des gendarmes.
Cette méthode donne même une indication sur la vitesse de convergence, qui est alors plus rapide qu'une convergence géométrique (ce qui est rapide).

Méthode 4 : Utilisation de l'IAF

Soient (u_n) une suite définie par la relation de récurrence $u_{n+1} = f(u_n)$ et I un intervalle tels que :
- Pour tout n, $u_n \in I$;
- Il existe $\ell \in I$, tel que $f(\ell) = \ell$;
- f est de classe $\mathscr{C}^1$ sur I et, pour tout $x \in I$, $\left|f'(x)\right| \leqslant k$ où $0 < k < 1$.

On dit que f est contractante mais ce terme est HP.

On va appliquer l'IAF à f sur I avec $x = u_n$ et à $y = \ell$ un point fixe de f (et il est donc capital de savoir que f est bien $\mathscr{C}^1$ sur l'intervalle I où vivent les termes de la suite ainsi que le point fixe candidat) permet d'obtenir que, pour tout $n \in \mathbb{N}$,

$$|u_{n+1} - \ell| = \left|f(u_n) - f(\ell)\right| \leqslant k|u_n - \ell|$$

Ensuite, une **récurrence** (immédiate) donne l'estimation :

$$(\star) \qquad |u_n - \ell| \leqslant k^n |u_0 - \ell|$$

et le théorème d'encadrement permet de conclure que (u_n) converge vers ℓ.

REMARQUE **1.8.**
Il est nécessaire que $0 < k < 1$ pour avoir $k^n \to 0, n \to +\infty$.

REMARQUE **1.9.**
La relation $(\star)$ nous dit en fait que la suite $(|u_n - \ell|)_n$ est sous-géométrique de raison k.
C' est aussi la base de l'élaboration de programmes Python permettant d'obtenir une valeur approchée de la limite.

Approximation d'un réel par une suite

En effet, un terme u_n fournira une approximation de ℓ à ε près si $|u_n - \ell| \leqslant \varepsilon$.
Il faut donc calculer u_n jusqu'à ce que cet écart soit assez petit. Mais comme on ne sait pas a priori calculer ℓ, on utilise la majoration, et on calcule u_n jusqu'à ce que k^n soit plus petit que ε.

```
def valeur_approchee(erreur):
    u=u0 # terme initial
    n=0
    while majorant**n >= erreur:
        n=n+1
        u=f(u)
    return u
```

EXERCICE 1.6. On considère la suite $(u_n)_{n\in\mathbb{N}}$ définie par

$$u_0 = 1 \quad \text{et} \quad \forall n \in \mathbb{N}, \quad u_{n+1} = f(u_n) = u_n + \frac{1}{4}\left(2 - u_n^2\right).$$

(1) Étudier les variations de f et montrer que $f([1;2]) \subset [1;2]$.
(2) Montrer que pour tout entier n, $u_n \in [1;2]$.
(3) Montrer que si (u_n) converge vers ℓ, alors $\ell = \sqrt{2}$.
(4) Montrer que pour tout $t \in [1;2]$, $\left|f'(t)\right| \leqslant \frac{1}{2}$.
(5) Montrer que pour tout entier n dans $\mathbb{N}$

$$\left|u_{n+1} - \sqrt{2}\right| \leqslant \frac{1}{2}\left|u_n - \sqrt{2}\right|$$

puis que :

$$\left|u_n - \sqrt{2}\right| \leqslant \left(\frac{1}{2}\right)^n$$

(6) En déduire la convergence de la suite (u_n).
(7) Écrire alors un programme en Python permettant d'obtenir une valeur approchée de $\sqrt{2}$ à 10^{-5} près.

1.2 Suites implicites

DÉFINITION 1.4

Une suite implicite (x_n) est une suite définie par une certaine équation (E_n) qui dépend de n.
$\rightsquigarrow$ Bien souvent, le terme x_n est l'unique solution de l'équation $f_n(x) = 0$.
$\rightsquigarrow$ En général, c'est le théorème de bijection qui permet de conclure à l'existence des termes de la suite.

Exemple : comme, pour tout $n \in \mathbb{N}^*$, la fonction $f_n : x \mapsto e^x + x - n$ réalise une bijection de $\mathbb{R}_+$sur $[1-n, +\infty[$, 0 (qui est bien dans l'intervalle image) admet un unique antécédent par f_n sur $\mathbb{R}_+$, que l'on note x_n.

REMARQUE **1.10.**
Si l'équation de départ n'est pas sous la forme $f_n(x) = 0$ mais par exemple $f(x) = n$ on peut toujours s'y ramèner en considérant l'équation $g_n(x) = 0$ où $g_n(x) = f(x) - n$.

REMARQUE **1.11.**
1. Comme l'indique son nom, une suite implicite n'est pas explicite. A priori, elle ne vérifie pas de relation de récurrence et il n'existe pas d'expression en fonction de n.
2. Les méthodes de la section précédente ne sont donc pas applicables.
3. La seule information dont on dispose sur la suite (x_n) est qu'elle est solution de l'équation E_n, c'est-à -dire qu'elle vérifie l'équation $f_n(x_n) = 0$. Cette relation permet de déduire de nombreuses propriétés de la suite et on y revient toujours.
4. En Python, on utilise alors souvent un programme permettant d'obtenir une valeur approchée de x_n par **dichotomie**. Prenons l'exemple d'une fonction f_n strictement croissante qui s'annule entre a et b. Le programme suivant renvoie une valeur approchée de x_n à 10^{-3} près (en supposant qu'une fonction Python `def f(n,x) :` définissant la fonction f_n ait bien été écrite auparavant).

```python
def va_x(n):
    a= ...
    b= ... # on sait que la solution x_n est entre a et b
    c=(a+b)/2 # milieu de l'intervalle
    while b-a > 10**(-3) : # ou autre degre de precision
        if f(n,a)*f(n,c)>0: # si la solution est entre c et b
            a=c # on decale a  droite
        else :
            b=c # on decale a  gauche
        c=(a+b)/2
    return c
```

1.2.1 Existence de la suite. Encadrement des termes

Méthode 5 : Existence d'une suite implicite

Pour montrer que la suite $(x_n)_{n\in\mathbb{N}}$ existe, il faut montrer que l'équation $f_n(x) = 0$ admet une unique solution. Pour cela, on utilise le **théorème de la bijection** en précisant les intervalles de départ et d'arrivée.
$\rightsquigarrow$ On doit bien vérifier et mentionner le fait que 0 est dans l'intervalle image, sinon il n'admet pas (d'unique) antécédent!
$\rightsquigarrow$ Il est absolument nécessaire de dresser le tableau de variation de la fonction f_n qui va servir de support à l'étude de la suite implicite (x_n).

REMARQUE **1.12.**
On n'oublie pas que le théorème de bijection donne aussi les variations de la bijection réciproque, et cela s'avère parfois pratique et donne des résultats immédiats.

Méthode 6 : Majoration/minoration d'une suite implicite

Pour montrer que la suite (x_n) est majorée et/ou minorée, on utilise les variations de la fonction f_n en **raisonnant sur les images** : les antécédents seront rangés dans le même ordre que les images, ou dans l'ordre opposé, selon que f_n est (strictement) croissante ou décroissante.

EXERCICE 1.7. (D'après EDHEC 2000) Pour tout entier $n \in \mathbb{N}^*$, on définit la fonction f_n sur $\mathbb{R}_+$ par

$$f_n(x) = x^n + 9x^2 - 4$$

(1) Montrer que l'équation $f_n(x) = 0$ n'a qu'une seule solution strictement positive, notée u_n.
(2) Calculer u_1 et u_2 puis vérifier que : $\forall n \in \mathbb{N}^*, u_n \in \left]0, \frac{2}{3}\right[$.

Méthode 7 : Monotonie de la suite

Pour étudier la monotonie de la suite (x_n), il faut comparer x_n et x_{n+1}, ce qu'on ne peut pas faire directement. On va donc comparer $f_n(x_n)$ et $f_n(x_{n+1})$ (il s'agit de deux images par la **même** fonction f_n). Si on sait que $f_n(x_n) = 0$, il faut alors estimer $f_n(x_{n+1})$.
Une fois que c'est fait, on peut conclure avec le sens de variation de f_n.

EXERCICE 1.8. On reprend la suite implicite de l'Exercice 7.
(1) Montrer que, pour tout x élément de $]0,1[$, on a : $f_{n+1}(x) < f_n(x)$.
(2) En déduire le signe de $f_n(u_{n+1})$, puis les variations de la suite (u_n).

1.2.2 Convergence de la suite et limite

Comme la suite est implicite, le moyen de prouver la convergence de la suite est d'utiliser le **théorème de convergence monotone**.
Une fois l'existence de la limite établie (notée ℓ), on peut alors passer à la limite dans la relation vérifié par x_n, à savoir $f_n(x_n) = 0$, ce qui donne une équation en ℓ.

Lors du passage à la limite dans la relation $f_n(x_n) = 0$, certaines formes indéterminée subtiles peuvent apparaître, du type $\lim\limits_{n\longrightarrow+\infty} x_n^n$.
Ce calcul assez délicat fait souvent l'objet d'une question dédiée.

EXERCICE 1.9 (Exercice 7 suite et fin)**.**
(1) Montrer que la suite (u_n) est convergente. On note ℓ sa limite.
(2) Déterminer la limite de u_n^n lorsque $n \longrightarrow +\infty$. (3) Donner enfin la valeur de ℓ.