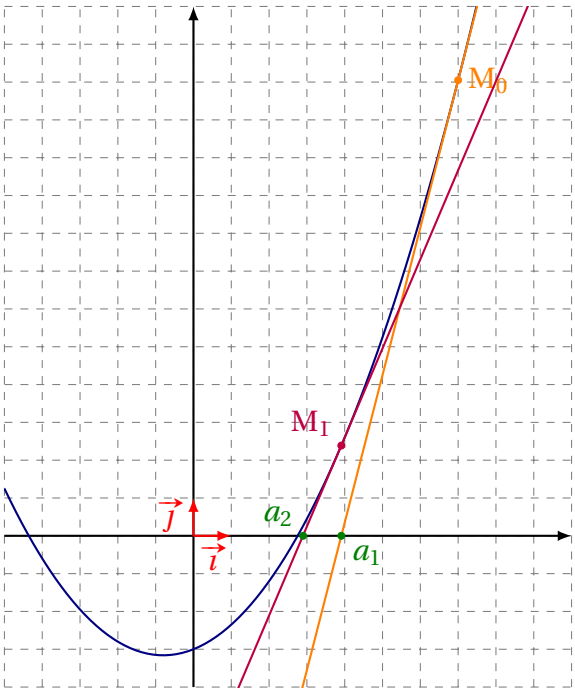
CHAPITRE 3

Suite numériques

Plan de ce chapitre

I. Rappels sur la méthode de Newton

Nous avons vu dans le chapitre précédent que la méthode de Newton a pour objectif de trouver des nombres qui se rapprochent de plus en plus d'une solution à l'équation $f(x) = 0$.



Les nombres a_1, a_2, ... se rapprochent de la solution.

On dit que les nombres a_1, a_2, ... sont les **termes successifs** de la **suite** (a_n).

II. Introduction

II. 1. Définition

Définitions 1

On appelle **suite numérique** une fonction dont la variable est un entier naturel.
Pour ne pas confondre avec les fonctions à variables réelles, la variable est mise en indice.

Si $(u_n)_{n\in\mathbb{N}}$ est une suite, alors n est l'**indice** du **terme** u_n.

On peut définir une suite de deux manières :

- par une fonction (si on connaît l'expression du terme général en fonction de l'indice) ;
- par récurrence (si on calcule un terme en fonction du ou des termes précédents).

Exemples 1

1. Si on définit la suite $(u_n)_{n\geqslant 1}$ par :

$$\forall\, n \geqslant 1, \quad u_n = \frac{1}{n},$$

alors $(u_n)_{n\geqslant 1}$ est définie par la fonction $n \mapsto \dfrac{1}{n}$; on dit qu'elle est définit de **façon explicite**.

2. Dans la méthode de Newton, les nombres a_n se calculent avec la relation :

$$\begin{cases} a_0 \text{ donné} \\ a_{n+1} = a_n - \dfrac{f(a_n)}{f'(a_n)}. \end{cases}$$

Dans la mesure où on calcule un terme à l'aide du terme précédent, la suite est définit de **façon récurrente.**

II. 2. Objectifs

L'étude des suites numériques consiste à :

- trouver leurs variations ;
- exprimer de façon explicite leur terme général (quand elles sont définies par récurrence).

Pour étudier les variations d'une suite, la dérivation ne sera d'aucune aide car cette notion est destinée aux cas où la variable est réelle (et non entière, comme c'est le cas pour les suites numérique).

Le second point est sans doute le plus problématique car, dans un cas général, il n'est pas chose aisée de passer de la forme par récurrence à la forme explicite.
En classe de 1re, nous verrons tout de même quelques exemples.

II. 3. Croissance, décroissance et constance

Définitions 2

- On dira qu'une suite (u_n) est **strictement croissante** à partir de n_0 si :

$$\forall\, n \geqslant n_0, \quad u_{n+1} > u_n.$$

- On dira qu'une suite (u_n) est **strictement décroissante** à partir de n_0 si :

$$\forall\, n \geqslant n_0, \quad u_{n+1} < u_n.$$

- On dira qu'une suite (u_n) est **constante** à partir de n_0 si :

$$\forall\, n \geqslant n_0, \quad u_{n+1} = u_n.$$

Exemples 2

1. **Cas d'une suite définie de façon explicite.**
   Posons $u_n = \sqrt{n+1}$. Quel que soit l'entier n,

   $$n+2 > n+1.$$

   La fonction $x \mapsto \sqrt{x}$ étant strictement croissante sur $\mathbb{R}_+$, on a :

   $$\sqrt{n+2} > \sqrt{n+1}$$

   (les images de deux nombres par une fonction croissante sont rangées dans le même ordre que les nombres)
   Ainsi :

   $$u_{n+1} > u_n$$

   ce qui signifie que la suite (u_n) est strictement croissante sur $\mathbb{N}$.
2. **Cas d'une suite définie par récurrence.**
   Posons pour tout entier naturel n :

   $$\begin{cases} v_0 = 9 \\ v_{n+1} = v_n^2 - v_n + 3 \end{cases}.$$

   Alors,

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

   Quel que soit la valeur de l'entier n, $v_n^2 \geqslant 0$ donc $v_n^2 + 3 \geqslant 3$, et donc $v_{n+1} - v_n > 0$.
   Ainsi, pour tout entier naturel n, $v_{n+1} > v_n$; la suite est donc croissante sur $\mathbb{N}$.

II. 4. Majorants et minorants

Déf. 3

- On appelle **majorant** d'une suite (u_n) tout nombre M tel que, pour tout entier naturel n, $u_n \leqslant \text{M}$.
- On appelle **minorant** d'une suite (u_n) tout nombre m tel que, pour tout entier naturel n, $u_n \geqslant m$.

Exemples 3

1. Soit $(u_n)_{n\geqslant 1}$ définie par $u_n = \dfrac{1}{n}$. « 0 » est un minorant de (u_n) car pour tout entier naturel n non nul, $u_n > 0$.
2. Soit (v_n) définie par $v_n = 3 - \dfrac{1}{n}$. « 3 » est un majorant de (v_n) car pour tout entier naturel n, $v_n \leqslant 3$.

III. Suites arithmétiques

III. 1. Définition

Définitions 4

On dit qu'une suite (u_n) est **arithmétique** si, pour tout entier naturel n,

$$u_{n+1} = u_n + r$$

où r est un nombre réel.

r est alors appelé la **raison** de la suite.

Exemple 4

Simon possède 30 €d'argent de poche.
Chaque semaine, il reçoit 5 € de la part de ses parents.

S'il ne dépense pas son argent petit à petit,

- après 1 semaine, il aura $30+5=35$ €;
- après 2 semaines, il aura $35+5=40$ €;
- après 3 semaines, il aura $40+5=45$ €;
- etc.

Si on note u_n la somme qu'il a après n semaine(s) alors :

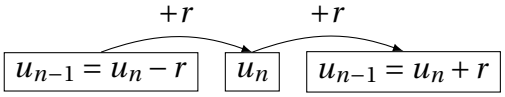
$$u_0 = 30 \quad ; \quad u_1 = 30+5 = 35 \quad ; \quad u_2 = 35+5 = 40 \quad ; \quad \ldots$$

que l'on peut aussi écrire :

$$\begin{cases} u_0 = 30 \\ u_{n+1} = u_n + 5 \end{cases}$$

La suite (u_n) est alors arithmétique de raison $r = 5$.

Remarque. Si on regarde trois termes consécutifs d'une suite arithmétique :



on constate que :

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

Ainsi, n'importe quel terme est la moyenne *arithmétique* des deux qui l'entourent; c'est la raison pour laquelle ces suites sont appelées *suites arithmétiques*.

III. 2. Variations

Propriété 1

Soit (u_n) une suite arithmétique de raison r.

- Si $r > 0$ alors (u_n) est strictement croissante.
- Si $r < 0$ alors (u_n) est strictement décroissante.
- Si $r = 0$ alors (u_n) est constante.

Démonstration

Pour tout entier naturel n,

$$u_{n+1} - u_n = (u_n + r) - u_n = r.$$

Donc,

- si $r > 0$, $u_{n+1} - u_n > 0$ et donc $u_{n+1} > u_n$; la suite (u_n) est donc strictement croissante.
- si $r < 0$, $u_{n+1} - u_n < 0$ et donc $u_{n+1} < u_n$; la suite (u_n) est donc strictement décroissante.
- si $r = 0$, $u_{n+1} - u_n = 0$ et donc $u_{n+1} = u_n$; la suite (u_n) est donc constante.

III. 3. Formule explicite

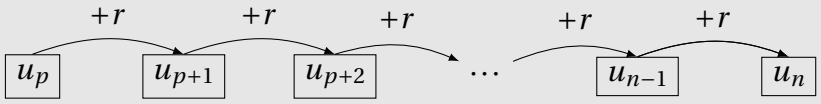
Propriété 2

Soit (u_n) une suite arithmétique de premier terme u_p et de raison r, $p \geqslant 0$. Alors,

$$\forall\, n \geqslant p, \quad u_n = u_p + (n-p)r.$$

Démonstration

Le schéma suivant illustre la situation dans laquelle nous sommes :



On peut écrire :

$$u_n = u_{p+(n-p)}$$

donc pour passer de u_p à $u_{p+(n-p)}$, on ajoute $(n-p)$ fois r, d'où :

$$u_n = u_p + (n-p)r.$$

Corollaire 1 (en prenant $p = 0$)

Soit (u_n) une suite arithmétique de premier terme u_0 et de raison r. Alors,

$$\forall\, n \geqslant 0, \quad u_n = u_0 + nr.$$

Corollaire 2 (en prenant $p = 1$)

Soit (u_n) une suite arithmétique de premier terme u_1 et de raison r. Alors,

$$\forall\, n \geqslant 1, \quad u_n = u_0 + (n-1)r.$$

Exemple 5

Reprenons l'exemple de Simon (exemple 4).
Nous avions établi l'égalité :

$$\forall\, n \geqslant 0, \quad u_{n+1} = u_n + 5 \qquad \text{avec } u_0 = 30.$$

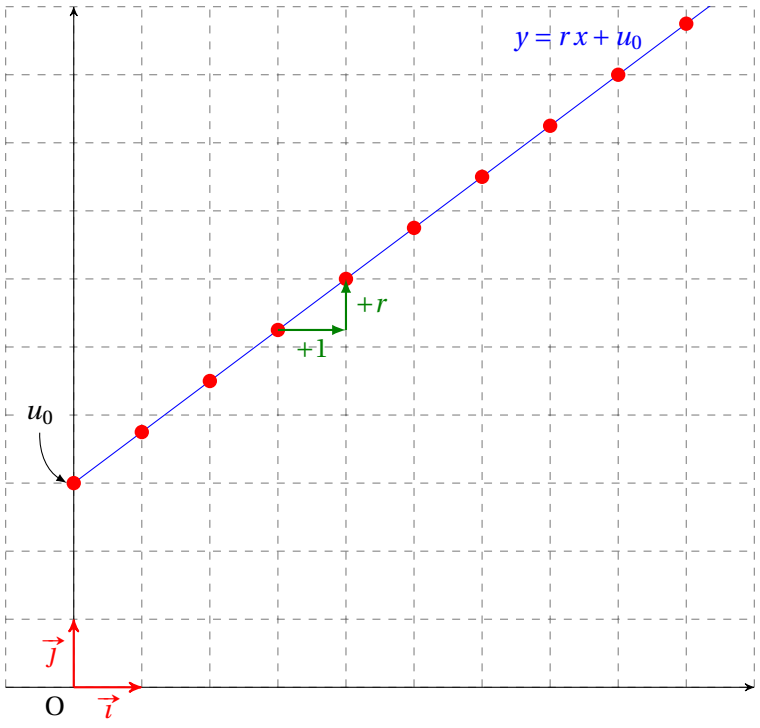
D'après le corollaire 1,

$$\forall\, n \geqslant 0, \quad u_n = 30 + 5n.$$

Cette formule explicite nous permet de connaître la somme que possède Simon après n semaines sans avoir à calculer tous les termes de la suite.

III. 4. Représentation graphique

Une suite peut être représentée par une série de points de coordonnées $(n\,;u_n)$.
Dans le cas d'une suite arithmétique de raison t, les points ont pour coordonnées $(n\,;u_0+nr)$.
Tous les points sont donc alignés sur la droite d'équation $y = r\,x + u_0$, où u_0 est l'ordonnée à l'origine et r, le coefficient directeur.



III. 5. Somme des premiers termes

III. 5. a. Somme des premiers entiers

Carl Friedrich Gauss (1777 – 1855) était un grand mathématicien allemand. On le surnomme le « prince des mathématiques » car l'histoire raconte que très jeune, il fit des prouesses en cette discipline.
On raconte par exemple que très jeune, il avait un instituteur qui voulait la paix et pour cela, il demanda à ses élèves de calculer la somme :

$$1+2+3+4+5+\cdots+98+99+100.$$

Mais il ne fallut que quelques secondes au petit Gauss pour donner la réponse : « 5 050 ».
Il remarqua en effet que :

$$1+100=2+99=3+98=\cdots=50+51=101\,,$$

et donc que la somme était égale à $50\times 101=5\,050$.

En s'inspirant de cette méthode, on peut généraliser pour trouver une expression qui donne la somme :

$$\mathrm{E}_n=1+2+3+\cdots+(n-2)+(n-1)+n.$$

En l'écrivant ainsi, puis à l'envers, et en ajoutant les deux égalités, on a :

$$\begin{array}{rcccccccccccccc}
\mathrm{E}_n & = & 1 & + & 2 & + & 3 & + & \cdots & + & (n-2) & + & (n-1) & + & n \\
\mathrm{E}_n & = & n & + & (n-1) & + & (n-2) & + & \cdots & + & 3 & + & 2 & + & 1 \\
\hline
2\mathrm{E}_n & = & (n+1) & + & (n+1) & + & (n+1) & + & \cdots & + & (n+1) & + & (n+1) & + & (n+1)
\end{array}$$

Ainsi,

$$2\mathrm{E}_n=n\times(n+1)$$

car il y a n fois le terme « $n+1$ », et donc :

$$\mathrm{E}_n=\frac{n(n+1)}{2}\,.$$

Propriété 3

$$1+2+3+\cdots+n=\frac{n(n+1)}{2}.$$

III. 5. b. Le symbole de sommation « $\sum$ »

Le symbole « $\sum$ » (c'est la lettre grecque *sigma majuscule*) est très souvent utilisé pour désigner une somme de termes.
Ainsi, quand on écrit $\sum_{k=0}^{100} u_k$, cela désigne la somme $u_0+u_1+u_2+u_3+u_4+\cdots+u_{99}+u_{100}$. La propriété 3 s'écrit alors :

$$\sum_{k=1}^{n} k=\frac{n(n+1)}{2}$$

III. 5. c. Somme des premiers termes d'une suite arithmétique

Dans ce paragraphe, nous souhaitons trouver une formule qui nous donne la somme :

$$S_n = \sum_{k=0}^{n} u_k = u_0 + u_1 + u_2 + \cdots + u_n$$

où (u_n) est une suite arithmétique de raison r.

D'après la propriété 2, nous savons que pour tout entier naturel k, $u_k = u_0 + kr$; donc :

$$\begin{aligned}
S_n &= u_0 + u_1 + u_2 + u_3 + \cdots + u_{n-1} + u_n \\
&= u_0 + (u_0 + r) + (u_0 + 2r) + (u_0 + 3r) + \cdots + \big[u_0 + (n-1)r\big] + (u_0 + nr) \\
&= \underbrace{(u_0 + u_0 + u_0 + \cdots + u_0)}_{(n+1)\text{ termes}} + \big[r + 2r + 3r + \cdots + (n-1)r + nr\big] \\
&= (n+1)u_0 + \underbrace{(1 + 2 + 3 + \cdots + n)}_{\text{propriété 3}} r \\
&= (n+1)u_0 + \frac{n(n+1)}{2} r. \qquad \text{(E)}
\end{aligned}$$

D'où la propriété suivante :

Propriété 4

Soit (u_n) une suite arithmétique de raison r. Alors,

$$u_0 + u_1 + u_2 + \cdots + u_n = (n+1)u_0 + \frac{n(n+1)}{2} r$$

$n+1$: nombre de termes dans la somme

Cette dernière formule n'est pas du goût de tout le monde ; en effet, elle peut paraître compliquée à retenir. C'est la raison pour laquelle nous allons la modifier légèrement.

Reprenons l'égalité (E) précédente et allons plus loin :

$$\begin{aligned}
S_n &= (n+1)u_0 + \frac{n(n+1)}{2} r \\
&= (n+1)\left[u_0 + \frac{nr}{2}\right] \\
&= (n+1)\left[\frac{2u_0 + nr}{2}\right] \\
&= (n+1)\left[\frac{u_0 + (u_0 + nr)}{2}\right] \\
&= (n+1) \times \frac{u_0 + u_n}{2}.
\end{aligned}$$

Cette dernière expression est plus facile à retenir car « n+1 » représente le nombre de termes dans la somme, u_0 désigne le premier terme de la somme et u_n, le dernier.

La somme est donc égale au « nombre de termes fois la moyenne arithmétique des termes extrêmes ».

Corollaire 3

Soit (u_n) une suite arithmétique de raison r. Alors,

$$u_0 + u_1 + u_2 + \cdots + u_n = (n+1) \times \frac{u_0 + u_n}{2}$$

ou encore :

$$\sum_{k=0}^{n} u_k = (\text{nombre de termes}) \times \frac{1^{\text{er}} \text{ terme} + \text{dernier terme}}{2}$$

Exemple 6

Considérons la suite (u_n) définie par :

$$\begin{cases} u_0 = 5 \\ u_{n+1} = u_n + 3 \end{cases}$$

C'est donc une suite arithmétique de raison 3.
Alors, d'après la propriété 2,

$$u_{200} = 5 + 200 \times 3 = 605$$

et donc, d'après le corollaire 3 :

$$u_0 + u_1 + u_2 + \cdots + u_{200} = 201 \times \frac{5 + 605}{2} = 201 \times 305 = 61\,305.$$

IV. Suites géométriques

IV. 1. Définition

Définitions 5

On dit qu'une suite (u_n) est **géométrique** si, pour tout entier naturel n,

$$u_{n+1} = q \times u_n$$

où q est un nombre réel non nul.

q est alors appelé la **raison** de la suite.

Exemple 7

Sophie a placé ses économies, à savoir 1 000 €, sur un livret qui lui rapporte 1,5 % par an, c'est-à-dire que d'une année à l'autre, ses économies augmenteront de 1,5 % du solde précédent.
Si on note $u_0 = 1\,000$ et u_n le montant de son livret après n années, alors :

$$u_{n+1} = u_n \times \left(1 + \frac{1,5}{100}\right) = 1,015u_n.$$

(u_n) est alors une suite géométrique de raison $q = 1,015$.

IV. 2. Formule explicite

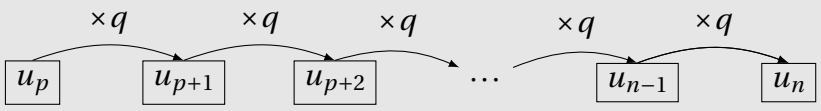
Propriété 5

Soit (u_n) une suite géométrique de premier terme u_p et de raison q, $q \geqslant 0$. Alors,

$$\forall\, n \geqslant p, \quad u_n = u_p \times q^{n-p}.$$

Démonstration

Le schéma suivant illustre la situation dans laquelle nous sommes :



On peut écrire :

$$u_n = u_{p+(n-p)}$$

donc pour passer de u_p à $u_{p+(n-p)}$, on multiplie par q, et ceci $(n-p)$ fois, d'où :

$$u_n = u_p \times \underbrace{q \times q \times \cdots \times q}_{(n-p) \text{ facteurs}} = u_p \times q^{n-p}.$$

Corollaire 4 (en prenant $p = 0$)

Soit (u_n) une suite géométrique de premier terme u_0 et de raison q. Alors,

$$\forall\, n \geqslant 0, \quad u_n = u_0 \times q^n.$$

Corollaire 5 (en prenant $p = 1$)

Soit (u_n) une suite géométrique de premier terme u_1 et de raison q. Alors,

$$\forall\, n \geqslant 1, \quad u_n = u_0 \times q^{n-1}.$$

Exemple 8

Reprenons l'exemple de Sophie (exemple 7) et calculons le solde de son livret après 10 ans (si elle ne retire pas d'argent entre temps). D'après le corollaire 4 :

$$u_{10} = u_0 \times q^{10} = 1\,000 \times 1{,}015^{10} \approx 1\,160{,}54.$$

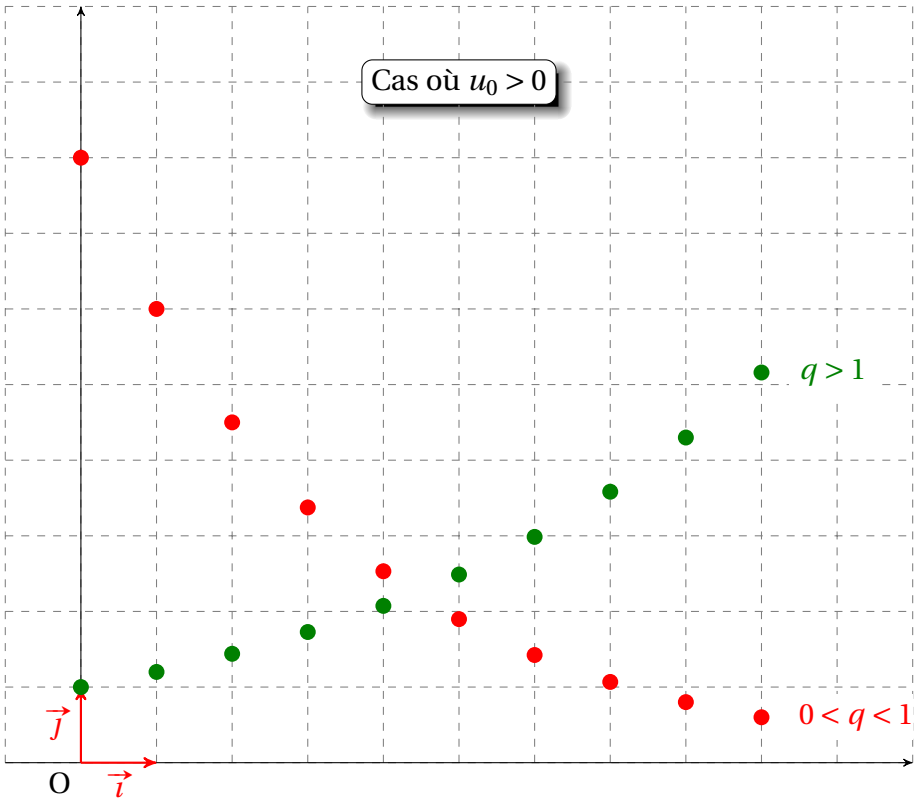
Ainsi, après 10 ans, elle possèdera 1 160,54 € sur son livret.

IV. 3. Représentation graphique

Selon les valeurs de la raison q de la suite géométrique, les points qui représentent les termes successifs « monteront » ou « descendront ».

Pour simplifier, prenons $\boldsymbol{u_0 > 0}$.

- Si $q = 1$ alors $u_{n+1} = u_n$, donc la suite (u_n) est constante (ce qui se traduira par une succession de points alignés sur une droite horizontale d'équation $y = u_0$).
- Si $q > 1$ alors l'égalité $u_{n+1} = qu_n$ nous dit que la suite est strictement croissante et donc que les points « monteront » (auront une ordonnée de plus en plus grande).
- Si $0 < q < 1$ alors l'égalité $u_{n+1} = qu_n$ nous dit que la suite est strictement décroissante et donc que les points « descendront » (auront une ordonnée de plus en plus petit en se rapprochant 0).



Lorsque $q > 1$, on s'aperçoit que les valeurs croissent de plus en plus vite ; on parle alors de **croissance exponentielle**.
Au contraire, quand $0 < q < 1$, les valeurs se rapprochent de plus en plus de 0. On parle alors de **limite nulle**.

IV. 4. Somme des premiers termes

IV. 4. a. Somme $1+q+q^2+\cdots q^n$

Propriété 6

Pour $q \neq 1$,

$$1+q+q^2+\cdots+q^n = \frac{1-q^n}{1-q}.$$

Démonstration

Posons $S_n = 1+q+q^2+\cdots+q^n$.
Alors,

$$qS_n = q+q^2+q^3+\cdots+q^{n+1}.$$

Ainsi,

$$qS_n = S_n - 1 + q^{n+1}$$

et donc :

$$qS_n - S_n = q^{n+1} - 1$$

soit :

$$(q-1)S_n = q^{n+1} - 1.$$

On en déduit alors que :

$$S_n = \frac{q^{n+1}-1}{q-1} \qquad \text{ou encore} \qquad S_n = \frac{1-q^{n+1}}{1-q}.$$

IV. 4. b. Somme des premiers termes d'une suite géométrique

Propriété 7

Soit (u_n) une suite géométrique de premier terme u_0 et de raison $q \neq 1$. Alors,

$$u_0+u_1+u_2+\cdots+u_n = u_0 \times \frac{1-q^{n+1}}{1-q}.$$

Démonstration

D'après la propriété 5 :

$$u_0+u_1+u_2+\cdots+u_n = u_0+qu_0+q^2u_0+\cdots+q^nu_0,$$

donc :

$$u_0+u_1+u_2+\cdots+u_n = u_0(1+q+q^2+\cdots+q^n.$$

Ainsi, d'après la propriété 6 :

$$u_0+u_1+u_2+\cdots+u_n = u_0 \times \frac{1-q^{n+1}}{1-q}.$$

V. Algorithmes

La notion de suites est propice aux algorithmes. En effet, une relation de récurrence définissant une suite implique nécessairement le même type de calculs à faire plusieurs fois. Il est donc important de savoir automatiser ces calculs.

V. 1. Suites arithmétiques

Prenons l'exemple de la suite (u_n) définie pour tout entier naturel n par :

$$\begin{cases} u_0 = 7 \\ u_{n+1} = u_n - 5 \end{cases}$$

V. 1. a. Calcul des termes successifs

On souhaite calculer et afficher tous les termes jusqu'à u_{50}.

```
u prend la valeur 7 (terme initial)
Pour n allant de 1 à 50:
    u prend la valeur u-5
    (on soustrait 5 à la valeur précédente)
    Afficher u
Fin du Pour
```

```
u = 7
for n in range(50):
    u = u-5
    print(u)
```

Une simple boucle suffit : on y calcule le nouveau terme en s'appuyant sur la relation de récurrence $u_{n+1} = u_n + r$, sachant qu'en algorithmique, les indices n'existent pas.
On stocke la valeur de u_0 dans la variable u et pour calculer u_1, on calcule la différence entre la valeur stockée dans u et 5.
À chaque passage dans la boucle, on calcule par rapport à la dernière valeur stockée dans la variable u.

V. 1. b. Calcul de la somme des termes

On souhaite ici calculer $u_0 + u_1 + \cdots + u_{50}$.

```
u prend la valeur 7 (terme initial)
s prend la valeur 7 (somme initiale)
Pour n allant de 1 à 50:
    u prend la valeur u-5
    s prend la valeur s+u
Fin du Pour
Afficher s
```

```
u,s = 7,7
for n in range(50):
    u = u-5
    s+=u
print(s)
```

La structure de l'algorithme est quasiment la même que celle de l'algorithme précédent, à ceci près que l'on ajoute une instruction dans la boucle pour calculer la somme.
Cette somme est calculée en prenant la valeur de la somme précédemment trouvée, et en lui ajoutant la valeur de u que l'on vient de calculer.

Le programme Python retourne « −6 018 » et la formule du corollaire 3 nous donne :

$$51 \times \frac{7 + (7 - 5 \times 50)}{2} = -6\,018.$$

V. 1. c. Calcul d'un seuil

On souhaite à présent savoir à partir de quel rang les termes sont inférieurs à −10 000.

```
u prend la valeur 7 (terme initial)
n prend la valeur 0 (indice initial) tant
que u>-10000:
    u prend la valeur u-5
    n prend la valeur n+1
Fin du Pour
Afficher n et u
```

```
u,n = 7,0
while u>=-10000:
    u-=5
    n+=1
print(n,u)
```

Ici, on recherche un indice, celui à partir duquel le terme est inférieur à −10 000. On ne va donc pas baser notre boucle sur les indices (comme dans les algorithmes précédents) mais sur la condition à remplir pour continuer à calculer les termes : on continue à calculer les termes TANT QUE le terme est plus grand que −10 000 (d'où l'utilisation de la boucle « Tant que », ou *while* en Python).

Le programme Python nous retourne l'indice $n = 2\,002$ (et le terme est alors $u_{2\,002} = -10\,003$).

Attention

Quand on utilise une boucle « Tant que » afin de déterminer un seuil, il ne faut pas oublier d'incrémenter l'indice à chaque passage dans la boucle (« `n prend la valeur n+1` »), sinon la boucle est infinie.
Cet oubli est assez fréquent quand on débute la programmation.

V. 2. Suites géométriques

Prenons l'exemple de la suite (v_n) définie pour tout entier naturel n par :

$$\begin{cases} v_0 = 10 \\ v_{n+1} = \dfrac{1}{2} v_n \end{cases}$$

V. 2. a. Calcul des termes successifs et de la somme des termes

On souhaite ici faire d'une pierre deux coups : calculer les termes successifs jusqu'à v_{20} ainsi que la somme $\sum_{k=0}^{20} v_k$.

```
v prend la valeur 10 (terme initial)
s prend la valeur 10 (somme initiale)
Pour n allant de 1 à 20:
    v prend la valeur 0.5*v
    Afficher v
    s prend la valeur s+v
Fin du Pour
Afficher s
```

```
v,s = 10,10
for n in range(20):
    v*=0.5
    print(v)
    s+=v
print("Somme = ",s)
```

Le programme Python retourne les valeurs de v_1 à v_{20} suivantes :

```
5.0
2.5
1.25
0.625
0.3125
0.15625
0.078125
0.0390625
0.01953125
0.009765625
0.0048828125
0.00244140625
0.001220703125
0.0006103515625
0.00030517578125
0.000152587890625
7.62939453125e-05
3.814697265625e-05
1.9073486328125e-05
9.5367431640625e-06
```

et affiche en fin la somme $v_0 + v_1 + \cdots + v_{20}$:
`Somme = 19.999990463256836.`

Si on a la curiosité de pousser la somme jusqu'à v_{100} (par exemple), le programme retourne :

`Somme = 20.0`

Et si on va jusqu'à v_{500} ou plus, la réponse est toujours `Somme = 20.0`.

On peut alors conjecturer (supposer) que cette somme se rapproche de plus en plus de 20. Mais il ne faut pas dire que toutes les sommes au-delà de v_{100} sont égales à 20... c'est impossible! Car si $\sum_{k=0}^{100} v_k = 20$ alors $\sum_{k=0}^{101} v_k = 20 + v_{101} \neq 20$ car $v_{101} \neq 0$.

Ce raisonnement est valable à n'importe quel rang : si on suppose que $\sum_{k=0}^{n} v_k = 20$ alors $\sum_{k=0}^{n+1} v_k = 20 + v_{n+1} > 20$ car $v_{n+1} > 0$.
Donc la somme n'est pas exactement égale à 20. La raison pour laquelle le programme affiche « 20 » est qu'il arrondi car il est arrivé au résultat « 19.999999999999999999... ».

V. 2. b. Calcul d'un seuil

Nous souhaitons savoir à partir de quel indice les termes sont inférieurs à 10^{-9}.

```
v prend la valeur 10 (terme initial)
n prend la valeur 0 Tan que u>=0.000000001:
    v prend la valeur 0.5*v
    n prend la valeur n+1
Fin du Pour
Afficher n et v
```

```
v,n=10,0
while v>=0.000000001:
    v*=0.5
    n+=1
print(n,v)
```

Le programme Python renvoie `n=34` et `v=5.820766091346741e-10`.
Ainsi, tous les termes à partir de v_{34} sont inférieurs à 10^{-9}.

VI. Autres types de suites

Les suites arithmétiques et géométriques ne sont pas les seuls types de suites, loin de là! Nous allons voir quelques exemples d'algorithmes affichant les premiers termes de diverses suites.

VI. 1. La suite factorielle

On définit la suite (u_n) par l'égalité :

$$\begin{cases} u_0 = 1 \\ u_n = n! = 1 \times 2 \times 3 \times \cdots \times n \end{cases}$$

« $n!$ » se dit : *factorielle n*.
L'algorithme et le programme suivants permettent de calculer $n!$.

```
fonction fact(n):
  Si n=0:
    Retourner 1
  Sinon:
    p = 1
    Pour k allant de 2 à n:
      p prend la valeur p*k
    Fin du Pour
    Retourner p
  Fin du Si

Afficher fact(10)
```

```
def fact(n):
    if n==0:
        return 1
    else:
        p = 1
        for k in range(1,n+1):
            p *= k
        return p

print(fact(10))
```

Le principe est le même que pour calculer la somme des premiers termes d'une suite, si ce n'est que l'on multiplie à la place d'ajouter.

Attention

La fonction `range` de Python est un peu spéciale. `range(n)` parcourt les nombres de 0 à $n-1$. Si on veut parcourir les nombres entiers de 1 à n, il faut procéder à un décalage comme dans le programme.

VI. 2. Suites de Syracuse

Pour un nombre u_0 initial donné, les suites de Syracuse sont définies ainsi :

$$u_{n+1} = \begin{cases} \dfrac{u_n}{2} & \text{si } u_n \text{ est pair} \\ 3u_n + 1 & \text{si } u_n \text{ est impair} \end{cases}$$

On souhaite calculer les 20 termes $u_1, u_2, \ldots, u_{20}$.

```
u prend une valeur quelconque
Pour n allant de 1 à 20:
  Si n est pair:
    u prend la valeur u/2
  Sinon:
    u prend la valeur 3*u+1
Fin du Si
Afficher u
Fin du Pour
```

```
u = int(input("Entrez un nombre : "))
for n in range(30):
    if u%2 == 0:
        u = u/2
    else:
        u = 3*u+1
    print(u,"\n")
```

```
226
113.0
340.0
170.0
85.0
256.0
128.0
64.0
32.0
16.0
8.0
4.0
2.0
1.0
4.0
2.0
1.0
4.0
2.0
1.0
4.0
2.0
1.0
4.0
2.0
1.0
4.0
```

Le programme retourne la série de nombres ci-contre.

La conjecture de Syracuse, autrement appelée *conjecture de Collatz*[a] ou encore *conjecture d'Ulam*[b] est l'hypothèse mathématique selon laquelle le terme « 1 » est toujours atteint, quel que soit le nombre de départ.

Malgré la simplicité de l'énoncé de cette conjecture, celle-ci défie depuis de nombreuses années les mathématiciens (on n'arrive pas à la démontrer).

Paul Erdős[c] dit d'ailleurs à son propos :

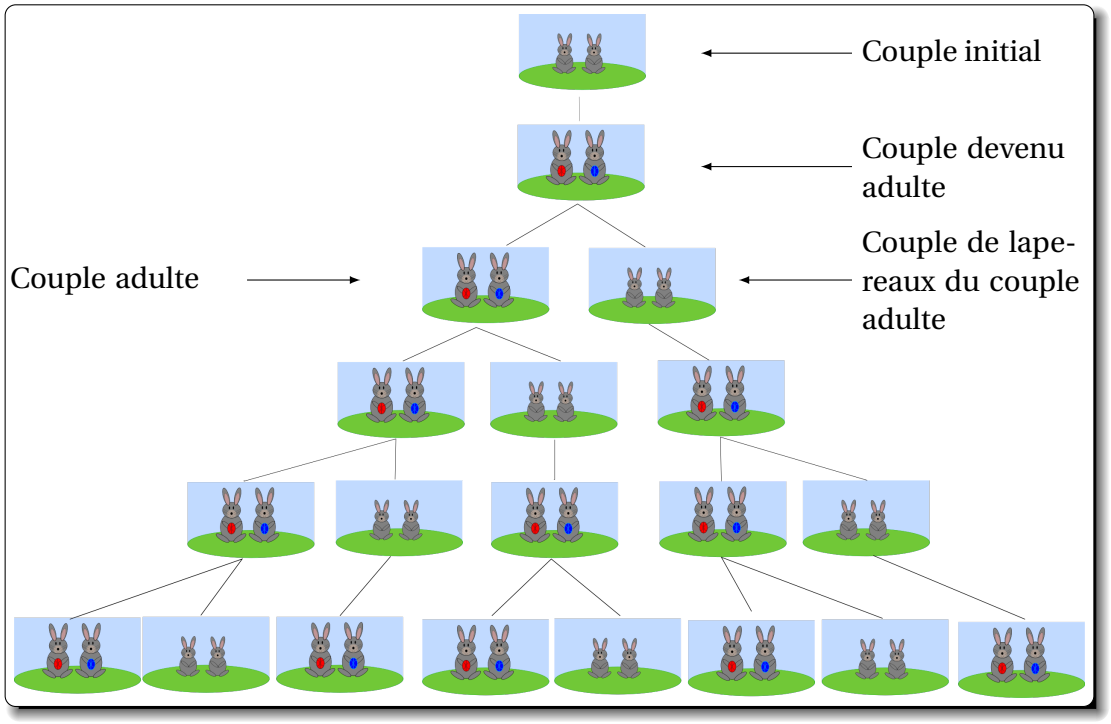
> « Les mathématiques ne sont pas encore prêtes pour de tels problèmes. »

a. Lothar Collatz (1910 – 1990) était un mathématicien allemand.
b. Stanislaw Ulam (1909 – 1984) était un mathématicien américain d'origine polonaise.
c. Paul Erdős (1913 – 1996) était un mathématicien hongrois

VI. 3. La suite de Fibonacci

Imaginez que l'on mette un couple de lapereaux sur une île déserte.
Il faut attendre un mois avant qu'ils deviennent adultes.
Un mois plus tard, ce couple donne naissance à un couple de lapereaux.
Imaginez que d'un mois à l'autre, un couple adulte donne naissance à un couple de lapereaux, sachant qu'il faut toujours attendre un mois pour qu'un lapereau devienne adulte.
Combien de couples y a-t-il sur l'île après n mois ?

C'est la question que s'est posée Leonardo Pisano, dit *Fibonacci* (1175 – 1250), mathématicien italien du Moyen-Âge.
On peut illustrer la situation à l'aide du schéma suivant :



Source : https://fr.wikipedia.org/wiki/Suite_de_Fibonacci

Notons $F_0 = F_1 = 1$, qui correspond au nombre de couple au mois initial et après 1 mois, et F_n le nombre de couples après n mois.
Ensuite, on a $F_2 = 1 + 1 = 2$ (le 1er couple et sa progéniture).
Ensuite, on a $F_3 = 2$ (couples adultes) $+ 1$ (couple de lapereaux) $= 3$.
Ensuite, on a $F_4 = 3$ (couples adultes) $+ 2$ (couples de lapereaux) $= 5$.

On s'aperçoit alors que $F_n = F_{n-1} + F_{n-2}$. C'est ainsi que l'on définit la **suite de Fibonacci**.

Déf. 6 La **suite de Fibonacci** est la suite $(F_n)_d$éfinie pour tout entier naturel n par :

$$F_0 = F_1 = 1 \quad , \quad F_{n+2} = F_{n+1} + F_n.$$

Contrairement aux suites arithmétiques et géométriques, pour calculer un terme quelconque, on fait appel à deux termes qui le précèdent. On dit que la suite est *d'ordre 2*.

Écrivons maintenant un algorithme qui calcule et affiche les termes successifs de cette suite :

```
u prend la valeur 1 (valeur de F_0)
v prend la valeur 1 (valeur de F_1)
Pour n allant de 1 à 20:
  w prend la valeur u+v (calcul du terme suivant)
  v prend la valeur u
  u prend la valeur w
  Afficher w
Fin du Pour
```

```
u,v = 1,1
print(u)
print(v)
for n in range(20):
    w = u+v
    v = u
    u = w
    print(w)
```

Le programme affiche :

```
1
1
2
3
5
8
13
21
34
55
89
144
233
377
610
987
1597
2584
4181
6765
10946
17711
```