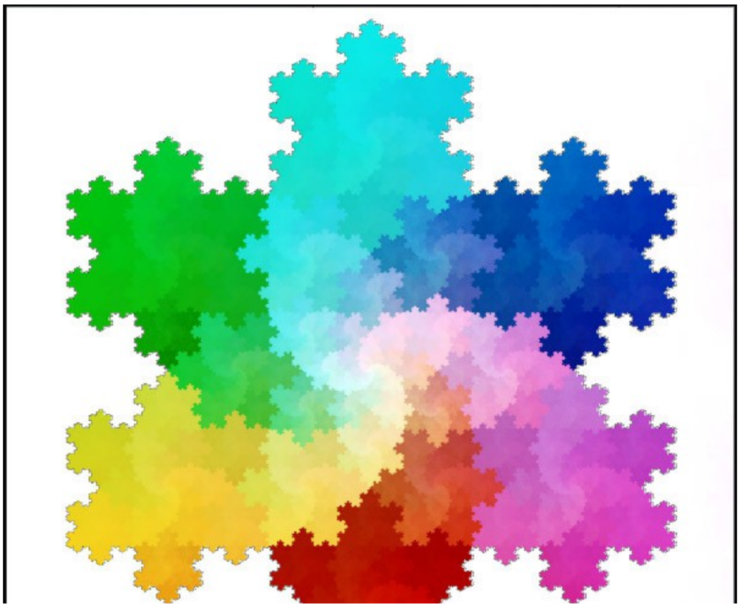
Terminale S

Les suites - Partie I : Raisonnement par récurrence

1.0



Olivier Lecluse

Juillet 2013

Table des matières

Introduction

Dans notre quotidien, placements, évolution de population, crédits etc... sont également autant de situations impliquant les suites. Par exemple, lorsque l'on contracte un crédit pour un projet immobilier, le capital restant dû est modélisé par une suite arithmético-géométrique dont nous verrons un exemple dans ce chapitre.

Ce chapitre sera l'occasion de découvrir un nouvel outil très puissant pour les démonstrations : le raisonnement par récurrence.

Celui-ci peut être illustré de manière très simple en pensant à une suite de domino dans laquelle, si un domino tombe, alors le suivant tombera. Il suffit alors que le premier domino tombe pour que tous les dominos tombent.

Ce principe très intuitif peut être formalisé de manière rigoureuse et permet de faire rapidement des démonstrations mathématiques.

Nous répondrons également à la question de savoir comment en ajoutant une infinité de nombres on peut aboutir à une somme finie. Cette question a été évoquée dès 500 avant J.C. par le philosophe Zénon d'Elée lorsqu'il a soumis le paradoxe d'Achille et la tortue. (cf - p.41*lien* - p.41). Ce sera l'occasion de découvrir la notion de limite.

Pour la bonne compréhension de ce chapitre, il peut être utile de revoir ce qui a été abordé en classe de première dans le *chapitre des suites*[1], en particulier les suites **arithmétiques** et **géométriques**.

1 - http://lcs.allende.lyc14.ac-caen.fr/~lecluseo/1ES/Suites_web/web/

Rappels de la classe de première

I

A. Définition

Définition : Suite numérique

Une suite numérique est une liste de nombres, rangés et numérotés :

- à l'entier 0 correspond le nombre noté u_0
- à l'entier 1 correspond le nombre noté u_1
- ...
- à l'entier n correspond le nombre noté u_n (appelé terme de la suite de rang n).

La suite est notée $(u_n)_{n\in\mathbb{N}}$ ou plus simplement (u_n).

Attention : Attention aux notations

Ne pas confondre le terme de rang n de la suite noté u_n avec la suite elle même notée (u_n).

Exemple : Placement

On place 5000€ sur un livret d'épargne. Le taux d'intérêts est de 2,5%. On s'intéresse à la somme disponible à l'année n.

La suite (u_n) représente la somme disponible en fonction du nombre d'années de placement.

Le n[ième] terme de la suite : u_n représente la somme disponible à l'année n.

B. Suite définie de façon explicite

Dans ce cas, on dispose d'une formule permettant de calculer directement Un en fonction de n.

C'est à dire qu'il existe une fonction f définie sur $[0;+\infty[$ telle que, pour tout entier n, $u_n = f(n)$.

Exemple

Soit (u_n) la suite définie par $u_n = 3n + 4$

Le premier terme de la suite u_0 est $3 \times 0 + 4 = 4$. On remplace n par 0.

Le second terme u_1 vaut $3 \times 1 + 4 = 7$

$u_{n+1} = 3 \times (n+1) + 4 = 3n + 3 + 4 = 3n + 7$ pour tout n

C. Suite définie par récurrence

Parfois, on ne dispose pas de formule directe permettant de calculer u_n en fonction de n, mais d'une formule permettant de calculer u_n en fonction des termes précédents. On calcule ainsi u_n en calculant systématiquement tous les termes de la suite de proche en proche à l'aide de la formule donnée.

Exemple

Soit la suite définie par la relation :

- $u_0 = 2$
- $u_{n+1} = 2 \times u_n - 3n$

La formule permet de dire que :

- $u_1 = 2 \times u_0 - 3 \times 0 = 2 \times 2 - 3 = 4$
- $u_2 = 2 \times u_1 - 3 \times 1 = 2 \times 4 - 3 = 5$
- $u_3 = ...$

Définition

On dit dans ce cas que la suite (u_n) est définie par une *relation de récurrence*.

Fondamental : Initialisation de la récurrence

Dans le cas de suites définies par récurrence, on a absolument besoin de connaître le (ou les) premier(s) terme(s) de la suite afin de pouvoir appliquer la formule de récurrence.

En effet, la seule formule $u_{n+1} = 2 \times u_n - 3n$ ne permet pas de calculer u_1 et encore moins les termes suivants.

D. Synthèse sur suites arithmétiques et géométriques

Rappel : Ce qu'il faut retenir de la classe de première

	Suite arithmétique de raison r, de premier terme u_0	**Suite géométrique de raison q de premier terme u_0**
Définition par récurrence	$u_{n+1} = u_n + r$	$u_{n+1} = q \times u_n$
Définition explicite	$u_n = u_0 + n \times r$	$u_n = u_0 \times q^n$
Relation entre deux termes u_n et u_p	$u_n = u_p + (n-p) \times r$	$u_n = u_p \times q^{n-p}$
$S_n = \sum_{k=0}^{n} u_k = u_0 + u_1 + \ ... \ +$	$S_n = \frac{n+1}{2}(u_0 + u_n)$	Si $q \neq 1$ $S_n = u_0 \times \frac{1-q^{n+1}}{1-q}$
$S_n = \sum_{k=p}^{n} u_k = u_p + u_{p+1} + \ ...$	$S_n = \frac{n-p+1}{2}(u_p + u_n)$	Si $q \neq 1$ $S_n = u_p \times \frac{1-q^{n-p+}}{1-q}$
$S_n = \sum u_k$	(nb de termes) $\times$ (moyenne (	(1er terme) $\frac{1 - \text{raiso}}{1 -}$

E. Dépasser un seuil

Une somme de 10000 euros est placée à un taux annuel de 3,5%. On note u_n le capital au bout de *n* années. Au bout de combien d'années ce capital double-t-il ?

Il y a plusieurs méthodes pour répondre à cette question. Nous allons en voir deux qui utilisent la calculatrice mais de manière différente.

Question 1

[*Solution n°1 p 29*]

Donner une formule de récurrence permettant de calculer la suite u_n

Question 2

[*Solution n°2 p 29*]

A l'aide de la fonction suites de la calculatrice, dresser un tableau de valeur de la suite u_n et en déduire la réponse à la question posée.

Question 3

[*Solution n°3 p 30*]

On considère l'algorithme suivant :

```
Initialisation :
 ... n prend la valeur 0
... u prend la valeur 10
Traitement :
... Tant que u < 20 Faire
... ... n prend la valeur n+1
... ... u prend la valeur u * 1.035
... Fin Tant que
```

```
Sortie :
... Afficher n
```

Compléter le tableau suivant :

	Etape 0	Etape 1	Etape 2
variable n	0		
variable u	10		
Condition u<20			

A quoi sert cet algorithme ?

Quel est le rôle de chacune des variables ?

Expliquer son fonctionnement.

Question 4

[*Solution n°4 p 31*]

Programmer cet algorithme et répondre à la question posée initialement.

Indice :

On pourra le programmer sur Python ou sur sa calculatrice.

La maîtrise des éléments de programmation[2] *sera nécessaire à partir de maintenant. Dans cette activité, on consultera plus particulièrement la section relative à la boucle Tant que sur Python - p.42, Casio - p.43 ou TI - p.43.*

F. Étude d'une suite arithmético-géométrique

Dans un parc naturel, la population de chamois diminue de 20% chaque année mais on introduit aussi 120 nouvelles bêtes. On note u_n le nombre de chamois à l'année n. On suppose qu'il y a 1000 chamois à l'année 0.

Question 1

[*Solution n°5 p 31*]

Donner l'expression de la suite par récurrence

Question 2

[*Solution n°6 p 32*]

Trouver le réel α solution de l'équation $\alpha = 0,8\alpha + 120$

Question 3

[*Solution n°7 p 32*]

Montrer que la suite $(v_n)_{n\geq 0}$ définie pour tout *n* par $v_n = u_n - \alpha$ est une suite géométrique de raison et de premier terme à préciser.

Indices :

On pourra exprimer v_{n+1} *en fonction de* u_n

On remarquera ensuite que $480 = 600 \times 0,8$

Mettre 0,8 en facteur dans l'expression v_{n+1}

2 - http://lcs.allende.lyc14.ac-caen.fr/~lecluseo/Algo/

Question 4

[Solution n°8 p 32]

En déduire une expression explicite de v_n puis de u_n

Indice :

On se rappellera que $v_n = v_0 \times q^n$ *est la formule explicite d'une suite géométrique.*

Question 5

[Solution n°9 p 32]

A l'aide de la calculatrice, conjecturer ce que deviendra le nombre de chamois dans les années qui viennent.

Raisonnement par récurrence

II

« Un voyage de mille lieues commence toujours par un premier pas »

Lao Tseu, Doo De Jing (-600 Av J.C)

Il a fallu plus de 3 siècles pour aboutir au raisonnement par récurrence tel qu'il va vous être présenté dans cette partie. Le raisonnement par récurrence a été inventé par Fermat et Pascal au XVIIe siècle, le principe de démonstration a été axiomatisé par Péano à la fin du XIXè siècle et son nom définitif lui a probablement été donné par Poincarré en 1902.

A. Le raisonnement par récurrence

Principe du raisonnement par récurrence

On peut se représenter le principe de raisonnement utilisé dans l'activité précédente comme une chaîne de dominos.

Si on veut faire tomber toute la chaîne, il faut s'assurer

- que le premier domino tombe. C'est ce qu'on appellera *la phase d'initialisation*.
- que les dominos sont placés de telle façon que lorsqu'un domino tombe, le suivant tombe aussi. C'est ce qu'on appellera la *propriété d'hérédité*.

Si ces deux conditions sont réunies, alors on est assuré que tous les dominos tomberont, aussi longue soit la chaîne.

On peut se donner d'autres représentations de ce principe comme monter à une échelle ou un escalier. Les bébés apprennent très vite à grimper d'une marche à une autre et se retrouvent souvent en haut d'un escalier sans savoir redescendre... Pour les empêcher de se retrouver dans une situation dangereuse, on place une barrière au niveau de la première marche, ce qui revient à les priver de la *phase d'initialisation*. Il sont ainsi dans l'incapacité de monter à l'escalier même s'ils maîtrise la *propriété d'hérédité* consistant à passer d'une marche à l'autre !

Fondamental : Le principe de récurrence

Soit $n_0 \in \mathbb{N}$ un entier.

On veut démontrer que pour tout rang $n \geq n_0$, une propriété $\mathcal{P}(n)$ (propriété au rang *n*) est vraie.

Pour cela on utilise la méthode suivante :

1. Initialisation : Vérifier que la propriété est vraie au rang n_0 ($\mathcal{P}_{n_0}$ est vraie)
2. Hérédité :
   - On **suppose** que la propriété est **vraie au rang n** ($\mathcal{P}_n$ est vraie)
   - On **démontre** alors que la propriété est **vraie au rang suivant** ($\mathcal{P}_{n+1}$ est vraie)
3. Conclusion : On conclut alors que la propriété est vraie à partir du rang n_0 (Pour tout $n \geq n_0$, $\mathcal{P}_n$ est vraie)

Exemple

On définit la suite (u_n) par

- $u_0 = 0$
- $u_{n+1} = u_n + 2n - 11$ pour tout $n \in \mathbb{N}$

Montrer que la suite (u_n) a pour forme explicite pour tout *n* : $u_n = n^2 - 12n$

Utilisons un raisonnement par récurrence :

Soit $\mathcal{P}_n$ la propriété $u_n = n^2 - 12n$

1. Initialisation :

   Si $n = 0, u_0 = 0$ est donnée par la formule de récurrence, mais aussi $u_0 = 0^2 - 12 \times 0$

   Donc $\mathcal{P}_0$ **est vraie au rang** $n = 0$. L'initialisation est réalisée.
2. Hérédité

Supposons qu'à un certain rang *n*, la propriété $\mathcal{P}_n$ soit vraie. Nous devons la **démontrer** au rang *n+1*

$u_{n+1} = u_n + 2n - 11$ d'après la définition de l'énoncé.

Mais alors $u_{n+1} = n^2 - 12n + 2n - 11$ en utilisant la propriété $\mathcal{P}_n$

En simplifiant on a $u_{n+1} = n^2 - 10n - 11$

Pour vérifier la formule au rang *n+1*, nous allons développer et réduire l'expression recherchée au rang *n+1* :

$(n+1)^2 - 12(n+1) = n^2 + 2n + 1 - 12n - 12 = n^2 - 10n - 11$. On reconnaît l'expression simplifiée de u_{n+1}

Donc on a bien $u_{n+1} = (n+1)^2 - 12(n+1)$. La propriété $\mathcal{P}_{n+1}$ est vraie au rang *n+1*. Elle est donc héréditaire.

3. Conclusion

Par un *raisonnement par récurrence*, on a **prouvé** que pour tout $n \geq 0$, la propriété $\mathcal{P}_n$ est **vraie**.

Donc Pour tout $n \in \mathbb{N}, u_n = n^2 - 12n$

B. Retrouver un résultat connu...

Question

[*Solution n°10 p 32*]

Montrer par récurrence que si une suite vérifie $u_{n+1} = r + u_n$ pour tout $n \in \mathbb{N}$, alors $u_n = r \times n + u_0$ pour tout $n \in \mathbb{N}$

C. Importance de l'initialisation

Attention : Attention à la phase d'initialisation

La première étape d'initialisation de la récurrence est souvent très simple à réaliser.

Elle semble parfois tellement simple qu'on peut être tenté de l'oublier pour se focaliser sur l'hérédité qui est souvent plus délicate à montrer.

l'exemple suivant montre qu'une récurrence non initialisée peut mener à des **résultats absurdes**.

Exemple : Récurrence non initialisée

Soit $\mathcal{P}_n$ la propriété « 2^n *est un multiple de 3* »

Cette propriété est **héréditaire**. En effet :

Si on suppose que $\mathcal{P}_n$ est **vraie**, alors $2^n = 3 \times k$ où *k* est un entier.

Mais alors $2^{n+1} = 2 \times 2^n = 2 \times 3 \times k$ puisque \mathcal P_n est vraie.

Cette dernière écriture montre que 2^{n+1} est un multiple de 3, donc que $\mathcal{P}_{n+1}$ est **vraie**.

Pourtant il est bien évident qu'une puissance de 2 ne pourra jamais être divisible par 3 ! ! La propriété **n'est pas initialisable** et on **ne peut tirer aucune conclusion** de l'hérédité.

D. Pour aller plus loin : calcul de la somme des carrés

Question

[Solution n°11 p 33]

Montrer par récurrence que pour tout $n \geq 1, \sum_{k=1}^{n} k^2 = 1^2 + 2^2 + 3^2 + \ ... \ + n^2 = \frac{n(n+1)(2n+1)}{6}$

Indices :

Attention ici, il s'agit de montrer une formule à partir du rang 1. Il faut donc prendre n=1 pour initialiser la récurrence.

On pourra remarquer que $\sum_{k=1}^{n+1} k^2 = \sum_{k=1}^{n} k^2 + (n+1)^2$

Un logiciel de calcul formel pourra être utile dans les calculs un peu longs.

Limite d'une suite

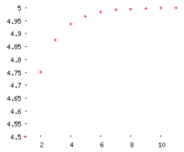
III

A. Exercice : Classer les suites selon leur limite

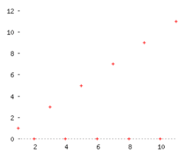
[*Solution n°12 p 34*]

En vous aidant de votre intuition, essayez de classer les suites dans la bonne catégorie.

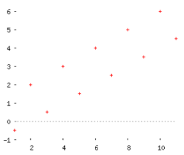
1 -



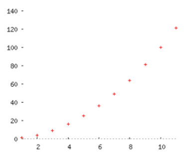
2 -



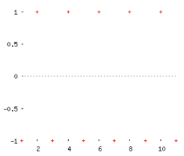
3 -



4 -



5 -



La suite tend vers l'infini

La suite tend vers un nombre

La suite n'a pas de limite, elle ne tend vers rien du tout

B. Limite infinie

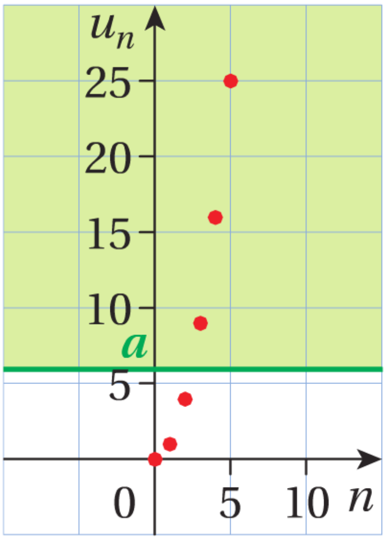
Définition

On dit que la suite (u_n) admet pour limite $+\infty$ si tout intervalle de la forme $]A\,;+\infty[$ (où A est un réel) contient tous les termes u_n de la suite à partir d'un certain rang.
On dit que la suite (u_n) admet pour limite $-\infty$ si tout intervalle de la forme $]-\infty\,;B[$ (où B est un réel) contient tous les termes u_n de la suite à partir d'un certain rang.
Cela revient à dire que $(-u_n)$ tend vers $+\infty$
On note alors $\lim\limits_{n\to+\infty} u_n = +\infty$ (respectivement $\lim\limits_{n\to+\infty} u_n = -\infty$).

Exemple

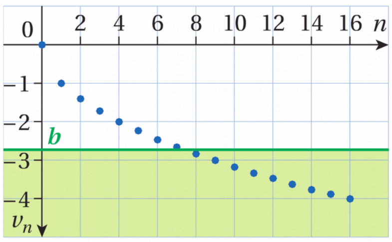
La suite (u_n) définie sur $\mathbb{N}$ par $u_n = n^2$ a pour limite $+\infty$

En effet, soit A un réel positif, l'intervalle $]A\,;+\infty[$ contient tous les termes de la suite dès lors que $n > \sqrt{A}$ (puisque la fonction carré est **croissante** sur $\mathbb{R}^+$)



Exemple

La suite (v_n) définie sur $\mathbb{N}$ par $v_n = -\sqrt{n}$ a pour limite $-\infty$



En effet soit B un réel négatif.
$-\sqrt{n} < B \Longleftrightarrow n > B^2$ en élevant au carré. En effet la fonction carré est **décroissante** sur $\mathbb{R}^-$
Par conséquent, tous les termes v_n de rang supérieur à B^2, seront dans l'intervalle $]-\infty\,;B[$

C. Exercice

Soit la suite (v_n) définie par $v_n = 3 - n^2$.

Question 1

[Solution n°13 p 34]

A l'aide de la calculatrice, conjecturer qu'il existe un rang *N* au delà duquel $v_n < -1000$.

Quel est ce rang ?

Question 2

[Solution n°14 p 34]

Démontrer par le calcul la conjecture de la question précédente.

Question 3

[Solution n°15 p 35]

Généraliser le résultat de la question précédente à n'importe quel seuil B.

Montrer ainsi que (v_n) est divergente et que sa limite est $-\infty$

Question 4

[Solution n°16 p 35]

On considère l'algorithme suivant :

```
Initialisation
... Saisir B
... N prend la valeur 0
... V prend la valeur 3
Traitement
... Tant que V>B Faire
... ... N prend la valeur N+1
... ... V prend la valeur 3-N^2
... Fin Tant que
Sortie
... Afficher N
```

Cet algorithme s'arrête t-il à un moment ? Justifier

Question 5

[Solution n°17 p 35]

Programmer l'algorithme ci-dessus sur Python ou votre calculatrice et donner la valeur retournée par celui-ci. lorsqu'on saisit la valeur -100

Ce résultat est-il cohérent avec le travail fait dans la première question ?

D. Introduction de la notion de limite finie

On considère la suite (u_n) définie par $u_0 = 10$ et $u_{n+1} = \frac{3}{5}u_n + 2$.

Question 1

[Solution n°18 p 36]

En utilisant une calculatrice ou un ordinateur, donner des valeurs approchées de u_1 à u_{10} puis des valeurs approchées de u_{20} et de u_{50} .

Émettre une conjecture sur la valeur limite de la suite (u_n)

Question 2

[Solution n°19 p 36]

Soit h un réel strictement positif.
Montrer que si $5 - h < u_n < 5 + h$, alors $5 - h < u_{n+1} < 5 + h$

Question 3

[Solution n°20 p 37]

Démontrer par récurrence que pour tout $n \geqslant 8$ on a $4,9 \leqslant u_n \leqslant 5,1$

Question 4

[Solution n°21 p 37]

Déterminer un entier *N* tel que pour tout $n \geqslant N$ on ait $4,999 \leqslant u_n \leqslant 5,001$

Importance de l'initialisation !

Nous avons démontré que

- pour un nombre *h* donné **si il existe** un rang n_0 tel que $5 - h < u_{n_0} < 5 + h$ alors pour tout entier $n \geqslant n_0$ il en sera de même.
- pour $h = 0,1$, le tableau de valeurs de la calculatrice nous donne l'existence d'un tel n_0 (8)
- pour $h = 0,001$, il suffit de prendre $n_0 = 20$

Par contre rien ne nous assure que pour n'importe quelle valeur de $h > 0$, un tel rang n_0 existe !

On peut toutefois conjecturer l'existence d'un tel rang n_0 pour n'importe quelle valeur de $h > 0$ au moyen d'un algorithme, comme le suggère la question suivante :

Question 5

[Solution n°22 p 37]

Écrire un algorithme prenant en entrée un nombre $h > 0$ et retournant le plus petit entier n_0 tel que $5 - h < u_{n_0} < 5 + h$

Programmer cet algorithme sur la calculatrice pour en déduire à partir de quel rang les termes de la suite (u_n) seront à une distance inférieure à 0,000001 de la limite $\ell = 5$

Pour démontrer rigoureusement l'existence d'un tel rang n_0 pour n'importe quelle valeur de $h > 0$, une démonstration mathématique s'impose. Une possibilité est de passer par nos connaissances sur les suites géométriques.

Question 6

[Solution n°23 p 38]

Soit (v_n) la suite définie pour tout entier naturel *n* par $v_n = u_n - 5$
Démontrer que (v_n) est géométrique et préciser sa raison.
En déduite la limite de (v_n)

Question 7

[Solution n°24 p 38]

En déduire la limite de (u_n)

E. Limite finie

Définition

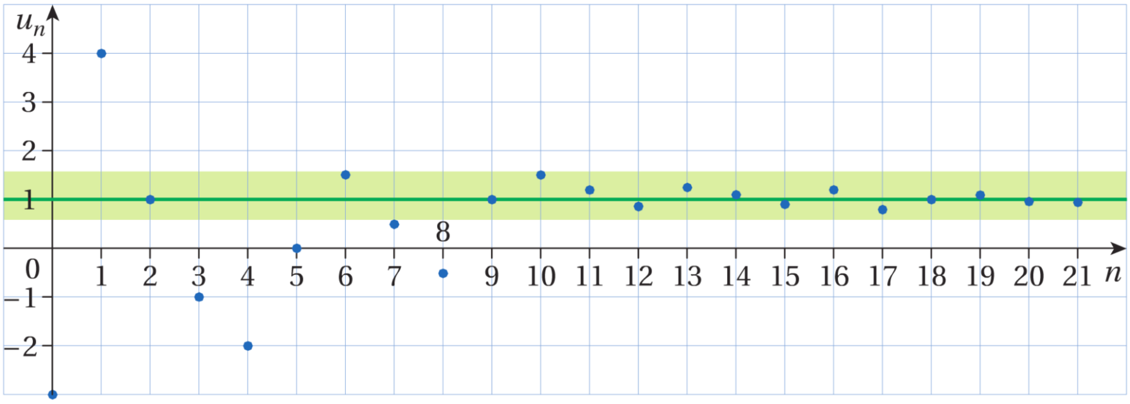
On dit que la suite (u_n) admet pour limite le réel ℓ si tout intervalle ouvert contenant ℓ contient **tous les termes de la suite à partir d'un certain rang**.

On dit alors que la suite (u_n) est *convergente* et *converge vers* ℓ.

On note alors $\lim\limits_{n \to +\infty} u_n = \ell$

Complément : Illustration

Dans le cas de la suite représentée ci-dessous, on voit que tous les termes **à partir du rang 9** appartiennent à un intervalle ouvert contenant 1 (matérialisé par la bande colorée en vert). Et plus l'intervalle ouvert se "**resserre**" autour de 1, plus il faut **aller chercher loin** le rang du terme à partir duquel tous les termes de la suite sont dans l'intervalle.



La suite représentée ici semble avoir pour limite 1

Remarque : Intervalle ouvert

Tout intervalle ouvert contenant *l* contient un intervalle ouvert centré en *l* de la forme $]\ell - \epsilon\,;\ell + \epsilon[$. On peut donc se contenter de chercher si tout intervalle ouvert de type $]l - \epsilon\,; l + \epsilon[$ contient tous les termes de la suite à partir d'un certain rang.

Remarque

$\lim\limits_{n \to +\infty} u_n = \ell \Longleftrightarrow \lim\limits_{n \to +\infty} u_n - \ell = 0$

Fondamental : Unicité de la limite

Si une suite (u_n) converge vers une limite, alors cette limite est unique.

F. Exercice

Démontrer que la suite (u_n) définie par $u_n = 1 + \dfrac{1}{n^2}$ est convergente et de limite 1.

Question 1

[*Solution n°25 p 38*]

On considère un intervalle $I_\epsilon =]1 - \epsilon\,; 1 + \epsilon[$ ouvert centré autour de 1.

Démontrer que si $n \geqslant \mathbb{E}\left(\dfrac{1}{\sqrt{\epsilon}}\right) + 1$, tous les termes de la suite sont dans I_ϵ

Indice :

$\mathbb{E}\left(\frac{1}{\sqrt{\epsilon}}\right)$ *désigne la partie entière de* $\frac{1}{\sqrt{\epsilon}}$

Question 2

[*Solution n°26 p 39*]

En déduire que la suite (u_n) converge et donner sa limite.

G. Limite d'une suite du type u(n+1)=f(un)

On considère la fonction $f \; : x \longmapsto \frac{3}{x+1}$

On définit la suite (u_n) par

- $u_0 = 2$
- $u_{n+1} = f(u_n)$

Question

[*Solution n°27 p 39*]

Tracer sur un même graphique

la droite d'équation $y = x$

la courbe C_f représentative de la fonction *f*

Utiliser ces éléments pour

- déterminer graphiquement les 5 premiers termes de la suite (u_n)
- conjecturer la limite de la suite (u_n)

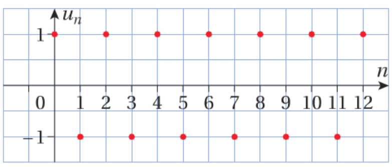
H. Des suites sans limites

Attention

Une suite n'a pas nécessairement de limite ! !

C'est notamment le cas de suites qui oscillent ou qui changent de signe de manière régulière.

Exemple

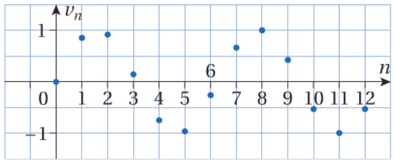


La suite définie par $u_n = (-1)^n$ alterne entre la valeur -1 et la valeur 1.

Un intervalle ouvert contenant 1 mais pas -1 ne contiendrait pas toutes les valeurs de la suite à partir d'un certain rang. On ne peut pas dire que cette suite ait pour limite 1. De même -1. On ne peut pas dire non plus que cette suite s'en va à l'infini.

Exemple

Les termes de la suite définie par $v_n = \sin n$ se répartissent uniformément dans l'intervalle [-1 ;1]. La suite (v_n) n'a pas non plus de limite.



Test final partie I

IV

Pour ce test d'auto-évaluation final, vous devez obtenir un minimum de 80% de bonnes réponses. En cas d'échec, révisez la section du cours qui vous a posé des difficultés et retentez à nouveau le test.

Exercice 1

Les suites ci-dessous sont définies pour tout entier n. Lesquels sont des suites croissantes ?

- [ ] $u_n = 4n + 5$
- [ ] $v_n = 3^n$
- [ ] $w_n = 10 \times 0.5^n$
- [ ] $z_n = 2 \times 1,01^n$

Exercice 2

Soit q un nombre réel tel que $1 + q + q^2 + ... + q^5 = 19608$*, alors q est égal à*

- ○ 4
- ○ 5
- ○ 6
- ○ 7

Exercice 3

Les suites ci-dessous sont définies pour tout entier n. Lesquelles ont une limite finie ?

- ☐ $u_n = \left(\frac{5}{2}\right)^n$
- ☐ $v_n = 0,5^n - 2^n$
- ☐ $w_n = 2 - \left(\frac{1}{10}\right)^n$
- ☐ $z_n = 3 \times 0,5^n$

Exercice 4

Les suites ci-dessous sont définies pour tout entier n*. Lesquelles ont une limite finie ?*

- ☐ $u_n = 1 + 2 + 2^2 + ... + 2^n$
- ☐ $v_n = 1 + 0,1 + 0,1^2 + ... + 0,1^n$
- ☐ $w_n = 2,5 + 2,5^2 + ... + 2,5^{n+1}$
- ☐ $z_n = 1 + \frac{9}{10} + \left(\frac{9}{10}\right)^2 + ... + \left(\frac{9}{10}\right)^n$

Exercice 5

cocher la ou les bonnes réponses

Exercice

Pour tout entier n*, on considère la proposition* $\mathcal{P}_n$ *: «* $6^n - 1$ *est un multiple de 5 »*

- ☐ La proposition $\mathcal{P}$ est héréditaire
- ☐ La proposition $\mathcal{P}$ est vraie sur $\mathbb{N}$
- ☐ Il existe un entier n tel que $\mathcal{P}_n$ est fausse

Exercice

Pour tout entier n*, on considère la proposition* $\mathcal{Q}_n$ *: «* $6^n + 1$ *est un multiple de 5 »*

- [ ] La proposition Q est héréditaire
- [ ] La proposition Q est vraie sur $\mathbb{N}$
- [ ] Il existe un entier *n* tel que Q_n est fausse

Solution des exercices

> Solution n°1 *(exercice p. 9)*

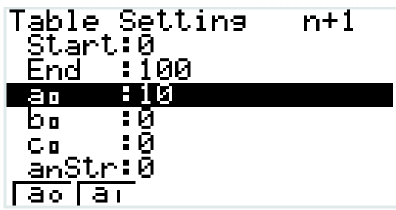
On peut choisir d'exprimer les termes de la suite u_n en milliers d'euros (mais ce n'est pas une obligation)

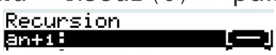
$u_0 = 10$

$u_{n+1} = u_n \times 1.035$

> Solution n°2 *(exercice p. 9)*

Méthode : Tabuler une suite définie par récurrence sur Casio

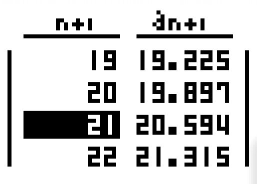


Sélectionner `Menu` `Recur(8)` puis sélectionner a_{n+1} : 

Dans `SET` dur la touche $\boxed{\text{F5}}$, définir le nombre de termes calculés et le terme initial de la suite : a0=10

De retour sur l'écran `Recursion`, appuyer sur $\boxed{\text{n.an}}$ ($\boxed{\text{F4}}$) pour saisir la formule de récurrence exprimant a_{n+1} en fonction de a_n.

Vous ferez appel à a_n au moyen de la touche $\boxed{\text{F2}}$.



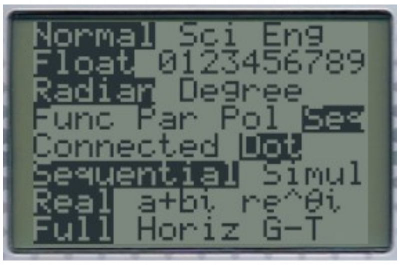
Les valeurs de la suite s'obtiennent par la fonction `TABL` (touche $\boxed{\text{F6}}$)

On lit ainsi sur l'écran la valeur de $u_{20} = 19,9$ et $u_{21} = 20,5$

Le capital a donc doublé au bout de 21 ans.

Méthode : Tabuler une suite définie par récurrence sur TI

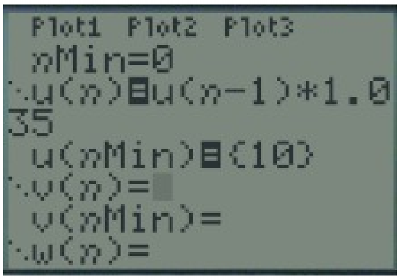
Choisir `mode` puis positionnez le mode suite (`Seq`)



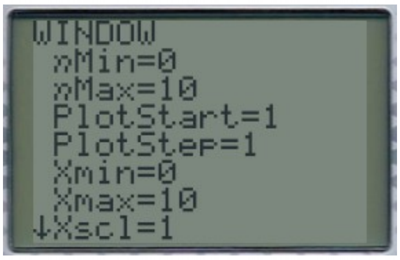
par la touche f(x), entrer la formule de récurrence donnant le terme u_n en fonction de u_{n-1}

L'expression u_{n-1} s'obtient par 2nde 7 (x,t,o,n - 1)

Renseignez la valeur de u_0 dans `u(nMin)` et 0 dans `nMin`



Définir dans `fenêtre` les valeurs min et max pour n



Les valeurs de la suite s'obtiennent par la fonction 2nde graphe

On lit ainsi sur l'écran la valeur de $u_{20} = 19,9$ et $u_{21} = 20,5$

Le capital a donc doublé au bout de 21 ans.

> Solution n°3 *(exercice p. 9)*

	Etape 0	Etape 1	Etape 2
variable n	0	1	2
variable u	10	10.35	10.71
Condition u<20	VRAI	VRAI	VRAI

On voit donc dans l'exécution pas à pas de l'algorithme que la variable n permet de comptabiliser le nombre d'années et que u contient la valeur de u_n.

Tant que le capital n'a pas doublé, on continue à augmenter de dernier de 3,5%

Dès qu'il a doublé, on dort de la boucle et on affiche le nombre d'année nécessaire au doublement du capital.

> Solution n°4 *(exercice p. 10)*

Méthode : A l'aide du langage python

Le *programme suivant*[3] implémente l'algorithme en utilisant le langage Python. On peut l'exécuter pas à pas.

La valeur retournée est 21, ce qui répond à la question initialement posée : Le capital a donc doublé au bout de 21 ans.

Méthode : Programmation sur Casio

Il faut bien savoir situer sur Casio les différentes fonctions et l'accès à l'éditeur de programmes. On se référera à la page de programmation sur Casio.

```
======SEUIL    ======
0→N↵
10→U↵
While U<20↵
N+1→N↵
U×1.035→U↵
WhileEnd↵
N◢
COM CTL JUMP ? ◢ ▷
```

La valeur retournée par le programme est 21. Le capital a donc doublé au bout de 21 ans.

Méthode : Programmation sur TI

Il faut bien savoir situer sur TI les différentes fonctions et l'accès à l'éditeur de programmes. On se référera à la page de programmation sur TI.

```
PROGRAM:SEUIL
:0→N
:10→U
:While U<20
:N+1→N
:U*1.035→U
:End
:Disp N
```

La valeur retournée par le programme est 21. Le capital a donc doublé au bout de 21 ans.

> Solution n°5 *(exercice p. 10)*

Au départ, on a 1000 chamois. à l'année *n+1*, le nombre de chamois est celui de l'année *n* que l'on diminue de 20% et auquel on ajoute 120. On obtient ainsi la définition par récurrence suivante :

- $u_0 = 1000$
- $u_{n+1} = 0,8 \times u_n + 120$

> Solution n°6 *(exercice p. 10)*

3 - http://www.pythontutor.com/visualize.html#code=n%3D0%0Au%3D10%0Awhile+u%3C20%3A%0A++++n%3Dn%2B1%0A++++u%3Du*1.035%0Aprint(n)&mode=edit&cumulative=false&heapPrimitives=false&drawParentPointers=false&textReferences=false&showOnlyOutputs=false&py=3

Nommons l'équation (E) : $\alpha = 0,8\alpha + 120$
$(E) \iff 0,2\alpha = 120$
$(E) \iff \alpha = \dfrac{120}{0,2}$
$(E) \iff \alpha = 600$

> Solution n°7 (exercice p. 10)

$v_{n+1} = u_{n+1} - 600$. En remplaçant u_{n+1} par sa valeur dans la formule de récurrence, on a
$v_{n+1} = 0,8 \times u_n + 120 - 600$
$v_{n+1} = 0,8 \times u_n - 480$. On remarque alors que $480 = 600 \times 0,8$ donc
$v_{n+1} = 0,8(u_n - 600)$ ce qui permet de conclure que
$v_{n+1} = 0,8 \times v_n$
De plus $v_0 = u_0 - 600 = 1000 - 600 = 400$
Donc la suite $(v_n)_{n\geqslant 0}$ est une **suite géométrique**

- de premier terme 400
- de raison 0,8.

> Solution n°8 (exercice p. 11)

On sait d'après le cours que $v_n = v_0 \times q^n = 400 \times 0,8^n$
De plus $u_n = v_n + 600$
On en déduit que $u_n = 400 \times 0,8^n + 600$

> Solution n°9 (exercice p. 11)

On sait que la suite (v_n) est décroissante car la raison est plus petite que 1. La suite (u_n) est donc décroissante également car elle s'obtient à partir de (v_n) par un simple décalage constant de 600.

Le nombre de chamois va donc diminuer dans les années à venir.

Par contre, $0,8^n$ est un nombre positif. On ne peut donc pas avoir un nombre de chamois inférieur à 600.

A l'aide de la calculatrice, on a $u_{10} \approx 707$ et $u_{30} \approx 601$

On peut donc conjecturer à partir de ces éléments, que le nombre de chamois va progressivement diminuer pour s'approcher de 600, sans jamais l'atteindre ni passer en dessous de ce seuil.

Cette conjecture fera l'objet d'une étude plus précise dans la suite de ce chapitre lorsque nous étudierons la notion de limite d'une suite.

> Solution n°10 (exercice p. 15)

Soit $\mathcal{P}_n$ la propriété $u_n = r \times n + u_0$

1. Initialisation
   $u_0 = 0 \times r + u_0$ donc $\mathcal{P}_0$ est **vraie**
2. hérédité
   Supposons que $\mathcal{P}_n$ soit vraie. Regardons au rang *n+1*:
   $u_{n+1} = u_n + r$ par définition
   $u_{n+1} = r \times n + u_0 + r$ en remplaçant u_n par sa valeur donnée par $\mathcal{P}_n$

Donc en regroupant les termes, $u_{n+1} = r \times (n+1) + u_0$, ce qui montre que $\mathcal{P}_{n+1}$ est **vraie**

3. conclusion

La propriété $\mathcal{P}_n$ est initialisée pour $n = 0$ et héréditaire. Le principe de démonstration par récurrence montre donc que pour tout $n \geq 0$, on a la formule $u_n = n \times r + u_0$

> Solution n°11 *(exercice p. 15)*

Soit $\mathcal{P}_n$ l'hypothèse de récurrence suivante : $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

1. Initialisation

Écrivons la formule au rang 1 :

$$\sum_{k=1}^{1} k^2 = 1^2 = 1$$

$$\frac{1(1+1)(2 \times 1 + 1)}{6} = \frac{2 \times 3}{6} = 1$$

Donc $\mathcal{P}_1$ est **vraie**. La récurrence est initialisée.

2. Hérédité

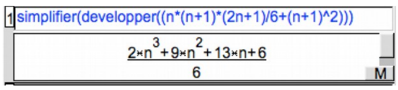
Supposons que $\mathcal{P}_n$ soit vraie au rang *n*. **Démontrons** la formule au rang *n+1*

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

en séparant la somme avec d'une part les n premiers termes et d'autre part le dernier terme.

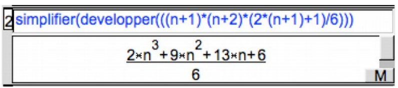
Or la première partie de la somme est connue par l'hypothèse de récurrence :

on obtient $\sum_{k=1}^{n+1} k^2 = \frac{n(n+1)(2n+1)}{6} + (n+1)^2$



A ce stade, utilisons un logiciel de calcul formel pour gagner du temps sur les calculs fastidieux.

Celui-ci nous montre en développant que $\frac{n(n+1)(2n+1)}{6} + (n+1)^2 = \frac{2n^3 + 9n^2 + 13n + 6}{6}$



Dans le même ordre d'idée, écrivons le membre de droite de notre formule de récurrence au rang n+1 et développons cette expression à nouveau à l'aide du logiciel de calcul formel :

$$\frac{(n+1)(n+1+1)(2(n+1)+1)}{6} = \frac{2n^3 + 9n^2 + 13n + 6}{6}$$

On tombe dans les deux cas sur la même formule développée. On peut donc en déduire que

$\sum_{k=1}^{n+1} k^2 = \frac{(n+1)(n+1+1)(2(n+1)+1)}{6}$ ce qui prouve que notre hypothèse de récurrence au rang *n+1* ($\mathcal{P}_{n+1}$) est vraie.

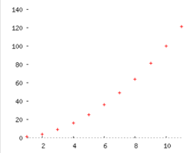
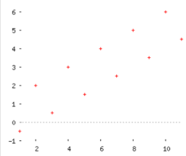
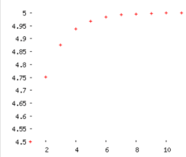
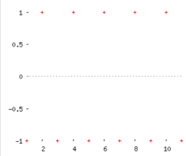
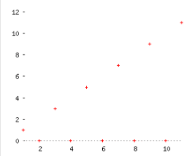
3. Conclusion

L'hypothèse $\mathcal{P}_1$ est vraie (**initialisation**)

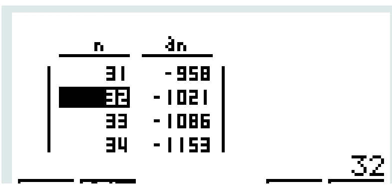
Nous avons vu de plus que si $\mathcal{P}_n$ est vraie alors $\mathcal{P}_{n+1}$ est vraie (**hérédité**)

Donc pour tout entier $n \geq 1$, nous avons la formule $\sum_{k=1}^{n} k^2 = \frac{n(n+1)(2n+1)}{6}$

> Solution n°12 *(exercice p. 17)*

La suite tend vers l'infini	 
La suite tend vers un nombre	
La suite n'a pas de limite, elle ne tend vers rien du tout	 

> Solution n°13 *(exercice p. 19)*



Il semble que pour $n \geqslant 32$, les valeurs de la suite (u_n)descendent en dessous du seuil de -1000.

> Solution n°14 *(exercice p. 19)*

$3 - n^2 < -1000 \Longleftrightarrow -n^2 < -1003$

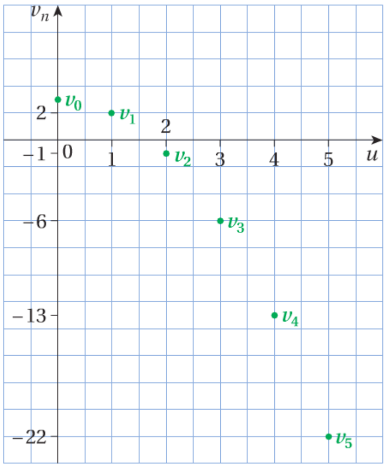
$3 - n^2 < -1000 \Longleftrightarrow n^2 > 1003$ (on renverse l'inégalité par passage à l'opposé)

$3 - n^2 < -1000 \Longleftrightarrow n > \sqrt{1003} \approx 31,7$ (on conserve l'inégalité car la fonction racine est croissante sur $\mathbb{R}^+$)

Par conséquent, à partir d'un rang *n* supérieur à 32, tous les termes de la suite (v_n)

sont dans l'intervalle $]-\infty\,;-1000[$

> Solution n°15 *(exercice p. 19)*



Soit un réel B négatif.
$3-n^2<B \iff -n^2<B-3$
$3-n^2<B \iff n^2>3-B$ (on renverse l'inégalité par passage à l'opposé)
$3-n^2<B \iff n>\sqrt{3-B}$ (si B<0, 3-B est positif et la fonction racine est croissante sur $\mathbb{R}^+$)
Par conséquent, à partir d'un rang *n* supérieur à $\sqrt{3-B}$, tous les termes de la suite (v_n) sont dans l'intervalle $]-\infty\,;B[$

> Solution n°16 *(exercice p. 19)*

On sait que la suite (v_n) tend vers $-\infty$. Quelle que soit la valeur de *B* saisie, on sait qu'à partir d'un certain rang *N*, les valeurs *v_N* seront inférieures à B, ce qui mettra fin à la boucle Tant que. L'algorithme s'arrête donc en affichant la plus petite valeur de *n* où le seuil *B* est dépassé.

> Solution n°17 *(exercice p. 19)*

Voici le *lien vers le programme*[4] réalisé en Python. Le code du programme est écrit ci-dessous.

```
# Initialisation
B=int(input("Saisir une valeur pour B"))
N=0
V=3
# Traitement
while V>B:
    N=N+1
    V=3-N**2
# Sortie
print(N)
```

La valeur retournée pour $B=-100$ est $N=11$.

Nous avions déterminé en première question que lorsque $n>\sqrt{3-B}$, alors $v_n<B$.

Or $\sqrt{3-B}=\sqrt{103}\approx 10,15$ donc pour $n\geq 11$, le seuil de -100 est dépassé, ce qui est bien le résultat donné par le programme.

4 - http://www.pythontutor.com/visualize.html#code=%23+Initialisation%0AB%3Dint(input(%22Saisir+une+valeur+pour+B%22))%0AN%3D0%0AV%3D3%0A%0A%23+Traitement%0Awhile+V%3EB%3A%0A++++N%3DN%2B1%0A++++V%3D3-N**2%0A%0A%23+Sortie%0Aprint(N)&mode=display&cumulative=false&heapPrimitives=false&drawParentPointers=false&textReferences=false&showOnlyOutputs=false&py=3&curInstr=0

> Solution n°18 (exercice p. 19)

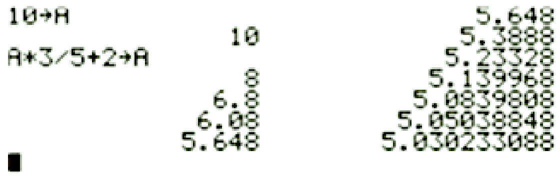
On peut obtenir les valeurs approchées des termes de la suite en utilisant une calculatrice ou un ordinateur.

Par exemple avec une calculatrice TI82, on pourra faire

10 [STO▶] A [ENTER] (on stocke la valeur initiale u_0 dans la mémoire A)

A ∗ 3/5 + 2 [STO▶] A [ENTER] (on fait le calcul du terme suivant et on le stocke dans A)

Il suffit ensuite d'appuyer plusieurs fois sur la touche [ENTER] pour obtenir les valeurs approchées successives des termes de la suite



Avec un tableur on peut entrer la valeur initiale 10 dans la cellule A1, la formule =A1*3/5+2 dans la cellule B1 et on recopie cette formule vers la droite autant de fois que nécessaire.

On obtient les valeurs suivantes :

$u_1 = 8$	$u_2 = 6{,}8$	$u_3 = 6{,}08$	$u_4 \approx 5{,}65$	$u_5 \approx 5{,}39$	$u_6 \approx 5{,}23$
$u_7 \approx 5{,}14$	$u_8 \approx 5{,}08$	$u_9 \approx 5{,}05$	$u_{10} \approx 5{,}03$	$u_{20} \approx 5{,}0001$	$u_{50} \approx 5$

Les valeurs approchées obtenues permettent de penser que la suite (u_n) a pour valeur limite 5.

> Solution n°19 (exercice p. 20)

Soit h un réel strictement positif .

Supposons que pour un entier n, on a $5 - h < u_n < 5 + h$

alors $\frac{3}{5}(5 - h) < \frac{3}{5}u_n < \frac{3}{5}(5 + h)$ donc $\frac{3}{5}(5 - h) + 2 < \frac{3}{5}u_n + 2 < \frac{3}{5}(5 + h) + 2$

c'est-à-dire $3 - \frac{3}{5}h + 2 < u_{n+1} < 3 + \frac{3}{5}h + 2$ donc $5 - \frac{3}{5}h < u_{n+1} < 5 + \frac{3}{5}h$

Sachant que h est un réel strictement positif, on a $\frac{3}{5}h < h$ et $-h < -\frac{3}{5}h$

On obtient donc $5 - h < 5 - \frac{3}{5}h < u_{n+1} < 5 + \frac{3}{5}h < 5 + h$ donc $5 - h < u_{n+1} < 5 + h$

Si $5 - h < u_n < 5 + h$ alors $5 - h < u_{n+1} < 5 + h$.

> Solution n°20 (exercice p. 20)

Considérons la proposition P_n : $4{,}9 \leqslant u_n \leqslant 5{,}1$
On a vu que $u_8 \approx 5{,}08$, donc $4{,}9 \leqslant u_8 \leqslant 5{,}1$, donc P_8 est vraie. (initialisation)
D'après la question précédente, en prenant $h = 0{,}1$ on peut dire que
si $5 - 0{,}1 < u_n < 5 + 0{,}1$ alors $5 - 0{,}1 < u_{n+1} < 5 + 0{,}1$
c'est-à-dire si $4{,}9 < u_n < 5{,}1$ alors $4{,}9 < u_{n+1} < 5{,}1$
ou encore si P_n est vraie alors P_{n+1} est vraie. (hérédité)
On en déduit que P_n est vraie pour tout $n \geqslant 8$.
Donc pour tout $n \geqslant 8$ on a $4{,}9 \leqslant u_n \leqslant 5{,}1$.

> Solution n°21 *(exercice p. 20)*

Considérons la proposition Q_n : $4{,}999 \leqslant u_n \leqslant 5{,}001$
On a vu que $u_{20} \approx 5{,}0001$, donc $4{,}999 \leqslant u_{20} \leqslant 5{,}001$, c'est-à-dire que Q_{20} est vraie. (initialisation)
En prenant $h = 0{,}001$ on a $5 - h = 4{,}999$ et $5 + h = 5{,}001$
La question 2 permet alors d'affirmer que si Q_n est vraie alors Q_{n+1} est vraie. (hérédité)
On en déduit que Q_n est vraie pour tout $n \geqslant 20$.
Donc pour tout $n \geqslant 20$ on a $4{,}999 \leqslant u_n \leqslant 5{,}001$.
NB : Le raisonnement a été initialisé à $n = 20$ en utilisant les résultats de la question 1, mais on aurait pu faire une initialisation à $n = 17$ car $u_{17} \approx 5{,}0008$ donc $4{,}999 < u_{17} < 5{,}001$

> Solution n°22 *(exercice p. 20)*

```
Entrées
... n=0
 ... U=10
... h prend la valeur 0
... Tant que h<=0 faire
 ... ... Saisir h
... Fin tant que
Traitement
... Tant que abs(U-5)>h Faire
... ... n prend la valeur n+1
... ... U prend la valeur 3/5*U+2
... Fin Tant que
Sortie
... Afficher n
```

Complément

Dans cet algorithme, une boucle « *Tant que* » a été introduite dans la partie entrée pour demander à l'utilisateur de saisir une valeur pour *h* tant qu'il s'obstine à rentrer une valeur négative ou nulle. Cela protège le programme l'empêchant de tourner indéfiniment si la valeur entrée n'est pas strictement positive.

Complément : Valeur absolue

Une astuce a été utilisée dans la condition en utilisant la fonction *Valeur absolue*.
Cette fonction $x \longmapsto abs(x)$ renvoie

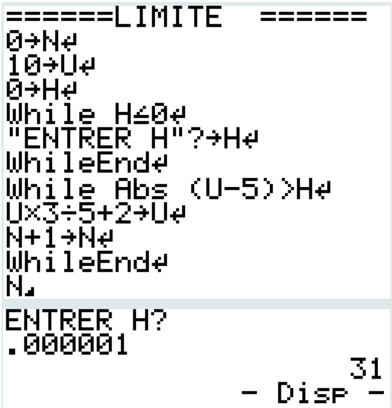
- x si x est **positif**

- -*x* si *x* est **négatif**

Ainsi $abs(-5) = 5$ et $abs(3) = 3$

La condition $abs(U-5) > h$ revient à dire que la distance entre U et 5 est supérieure à h donc $U < 5-h$ **ou** $U > 5+h$

Simulateur : Programme Casio



```
======LIMITE  ======
0→N↵
10→U↵
0→H↵
While H≤0↵
"ENTRER H"?→H↵
WhileEnd↵
While Abs (U-5)>H↵
U×3÷5+2→U↵
N+1→N↵
WhileEnd↵
N◢
ENTRER H?
.000001
                31
          - Disp -
```

Voici le programme correspondant sur Casio. La valeur absolue se trouve dans $\boxed{\text{OPTN}}$ $\boxed{\text{NUM}}$. Ceux qui préfère saisir $U < 5-h$ **ou** $U > 5+h$ trouveront les relations logiques dans $\boxed{\text{OPTN}}$ $\boxed{\text{LOGIC}}$

> Solution n°23 *(exercice p. 20)*

Montrons que (vn) est géométrique

$$v_{n+1} = u_{n+1} - 5 = \frac{3}{5}u_n + 2 - 5 = \frac{3}{5}u_n - 3 = \frac{3}{5}(u_n - 5) = \frac{3}{5}v_n$$

Donc la suite (v_n) est géométrique de raison $\frac{3}{5}$

Complément : En déduire sa limite.

On sait depuis la classe de première que la limite des suites géométriques de raison $0 < q < 1$ est 0.

Donc $\lim\limits_{n \to +\infty} v_n = 0$

> Solution n°24 *(exercice p. 20)*

Puisque $u_n = v_n + 5$, et puisque (v_n) tend vers 0, on en déduit bien que (u_n) tend vers $0+5=5$

> Solution n°25 *(exercice p. 21)*

si $n \geqslant \mathbb{E}\left(\frac{1}{\sqrt{\epsilon}}\right) + 1$, alors $n > \frac{1}{\sqrt{\epsilon}}$ et donc $n^2 > \frac{1}{\epsilon}$

Par passage à l'inverse, puisque la fonction $x \longmapsto \frac{1}{x}$ est strictement décroissante, on en déduit que $\frac{1}{n^2} < \epsilon$

Mais on sait aussi que $\frac{1}{n^2} > 0$ et $-\epsilon < 0$ donc $-\epsilon < \frac{1}{n^2} < \epsilon$

en ajoutant 1 à cette inégalité, on en déduit que si $n \geqslant \mathbb{E}\left(\frac{1}{\sqrt{\epsilon}}\right) + 1$, alors $1 - \epsilon < 1 + \frac{1}{n^2} < 1 + \epsilon$, ce qui revient à dire que $u_n \in I_\epsilon$

> Solution n°26 *(exercice p. 22)*

On vient de démontrer qu'un intervalle ouvert centré en 1 de type $]1 - \epsilon\,; 1 + \epsilon[$ contient tous les termes de la suite (u_n) à partir d'un rang $n \geqslant \mathbb{E}\left(\frac{1}{\sqrt{\epsilon}}\right) + 1$.
C'est exactement la définition d'une suite convergente de limite 1.

> Solution n°27 *(exercice p. 22)*

Simulateur

partant de u_k on utilise la courbe C_f pour déterminer $u_{k+1} = f(u_k)$ puis la droite $y = x$ pour reporter u_{k+1} sur l'axe des abscisses et réitérer la procédure.

D'après la simulation, on peut émettre la conjecture que la suite (u_n) converge et que sa limite est 1,3

Remarque

En déplaçant le curseur, on s'aperçoit **sur cet exemple** que la convergence ne semble pas dépendre du premier terme de la suite. Il n'en est pas toujours de même. Cela va dépendre de la forme de la courbe C_f

Contenus annexes

- Paradoxe de Zénon

Zénon : Philosophe grec qui a émis un certain de nombre de paradoxes lié à cette illusion du mouvement.

Dans le paradoxe d'Achille et de la tortue:

Il est dit qu'un jour, le fameux héros grec Achille a disputé une course à pied avec une tortue.

Comme Achille était réputé être un coureur très rapide, il avait accordé gracieusement à la tortue une avance de mille mètres (1 km).

Zénon affirme alors que le rapide Achille n'a jamais pu rattraper la tortue:

En effet, supposons pour simplifier le raisonnement que chaque concurrent court à vitesse constante, l'un très rapidement, et l'autre très lentement

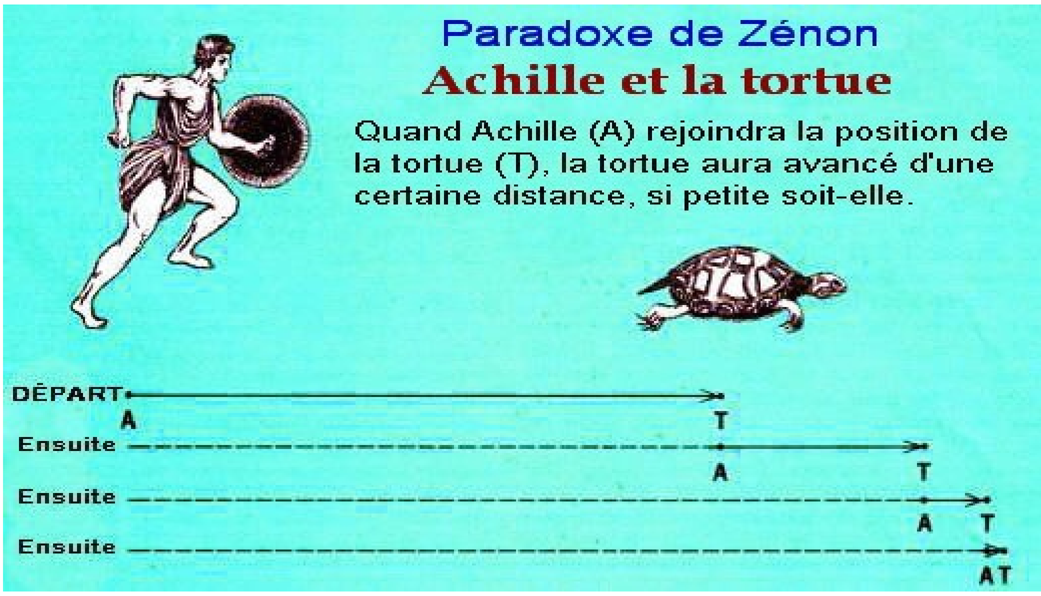
Au bout d'un certain temps, Achille aura comblé ses mille mètres de retard et atteint le point de départ de la tortue ; mais pendant ce temps, la tortue aura parcouru une certaine distance, certes beaucoup plus courte, mais non nulle, disons 100 mètres.

Cela demandera alors à Achille un temps supplémentaire pour parcourir cette distance, pendant lequel la tortue avancera encore plus loin.

Et puis une autre durée avant d'atteindre ce troisième point, alors que la tortue aura encore progressé.

Ainsi, toutes les fois où Achille atteint l'endroit où la tortue se trouvait, elle se retrouve encore plus loin.

Par conséquent, Achille n'a jamais pu et ne pourra jamais rattraper la tortue.



- Boucle Tant que

Définition

Cet type de boucles est utilisé lorsque l'on **ne connaît pas** a priori le nombre de fois que l'on veut répéter une partie du programme.

Méthode

La structure algorithmique Tant que... Faire est représentée sous Python par

while *condition* **:**

... instructions à répéter

Attention : Bloc de programme

Sous Python, le commencement de la partie du programme se situant dans le *Tant que* est marqué par les deux points **":"**

La partie du code qui est située dans le bloc *Tant que* est **décalée par rapport à la marge**. C'est ce décalage qui permet de marquer également la fin du bloc. Il n'y a donc pas de mot particulier pour marquer la fin du *Tant que*.

Exemple

```
a=10
n=0
while a<50 :
 ... a=a*1.02
... n=n+1
print ("50 a été atteint au bout de ",n," etapes")
```

Dans ce programme, les ... désignent un décalage par rapport à la marge.

- Boucle Tant que

Définition

Cet type de boucles est utilisé lorsque l'on **ne connaît pas** a priori le nombre de fois que l'on veut répéter une partie du programme.

Méthode

La structure algorithmique Tant que... Faire est représentée sous Casio par

While *condition*

instructions à répéter

WhileEnd

Contrairement à Python, il est nécessaire de préciser ici la fin des instructions contenues dans le while à l'aide du mot clé **WhileEnd**.

Les instructions While ... WhileEnd se trouvent sur Casio accessibles depuis le menu [PRGM] sous l'entrée [COM] en alland chercher le troisième sous-menu([F6]).

Vous obtenez ceci en bas de l'écran : While WEnd Do Lp-W

Exemple

```
======WHILE    ======
10→A↵
0→N↵
While A<50↵
A×1.02→A↵
N+1→N↵
WhileEnd↵
N◢
```

- Boucle Tant que sur TI

Définition

Cet type de boucles est utilisé lorsque l'on **ne connaît pas** a priori le nombre de fois que l'on veut répéter une partie du programme.

Méthode

La structure algorithmique Tant que... Faire est représentée sous TI par

While *condition*

instructions à répéter

End

Contrairement à Python, il est nécessaire de préciser ici la fin des instructions contenues dans le while à l'aide du mot clé **End**.

Les instructions **While** ... **End** se trouvent sur TI accessibles depuis le menu prgm en mode d'édition. Le menu ci-contre apparaît sous l'onglet CTL

Exemple