Introduction au calcul flottant

MT09
Vincent.Martin@utc.fr

UTC Compiègne
France

UTC, A2018
Plan

1. Introduction

2. Représentation des nombres

3. Calculs en précision limitée
1. Introduction
2. Représentation des nombres
3. Calculs en précision limitée
Une suite curieuse

Quelle est la limite de la suite $(u_n)_{n \in \mathbb{N}}$ ?

\[ u_n = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}}, \quad u_0 = 2, \ u_1 = -4 \]

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>18.5000000000000</td>
<td>17</td>
<td>7.2350211655349</td>
</tr>
<tr>
<td>4</td>
<td>9.37837837837838</td>
<td>18</td>
<td>22.0620784635258</td>
</tr>
<tr>
<td>5</td>
<td>7.80115273775217</td>
<td>19</td>
<td>78.5755748878722</td>
</tr>
<tr>
<td>6</td>
<td>7.15441448097533</td>
<td>20</td>
<td>98.3495031221654</td>
</tr>
<tr>
<td>7</td>
<td>6.80678473692481</td>
<td>21</td>
<td>99.8985692661829</td>
</tr>
<tr>
<td>8</td>
<td>6.59263276872179</td>
<td>22</td>
<td>99.9938709889028</td>
</tr>
<tr>
<td>9</td>
<td>6.44946593405393</td>
<td>23</td>
<td>99.9996303872863</td>
</tr>
<tr>
<td>10</td>
<td>6.34845206074662</td>
<td>24</td>
<td>99.999977306795</td>
</tr>
<tr>
<td>11</td>
<td>6.27443866272812</td>
<td>25</td>
<td>99.999986592167</td>
</tr>
<tr>
<td>12</td>
<td>6.21869676858216</td>
<td>26</td>
<td>99.999999193218</td>
</tr>
<tr>
<td>13</td>
<td>6.17585385581539</td>
<td>27</td>
<td>99.999999951478</td>
</tr>
<tr>
<td>14</td>
<td>6.14262717048101</td>
<td>28</td>
<td>99.999999997083</td>
</tr>
<tr>
<td>15</td>
<td>6.1204870457016</td>
<td>29</td>
<td>99.999999999825</td>
</tr>
<tr>
<td>16</td>
<td>6.16608655959810</td>
<td>30</td>
<td>99.999999999989</td>
</tr>
</tbody>
</table>
Une suite curieuse

Quelle est la limite de la suite \((u_n)_{n \in \mathbb{N}}\) ?

\[ u_n = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}}, \quad u_0 = 2, \quad u_1 = -4 \]

Pourtant :

\[ u_n = \frac{3 \cdot 6^{n+1} - 4 \cdot 5^{n+1}}{3 \cdot 6^n - 4 \cdot 5^n} \Rightarrow \lim_{n \to \infty} u_n = 6 \]
Plan

1. Introduction

2. Représentation des nombres

3. Calculs en précision limitée
Écriture des entiers en base 2

### En base 10

<table>
<thead>
<tr>
<th>Chiffres 0, 1, . . . , 9</th>
</tr>
</thead>
</table>

\[
n = d_p10^p + \ldots + d_110 + d_0, \quad 0 \leq d_i \leq 9, \quad d_p \neq 0
\]

<table>
<thead>
<tr>
<th>1789 = 1000 + 700 + 80 + 9 = 1 \cdot 10^3 + 7 \cdot 10^2 + 8 \cdot 10^1 + 9 \cdot 10^0</th>
</tr>
</thead>
</table>
Écriture des entiers en base 2

**En base 10**

Chiffres 0, 1, ..., 9

\[ n = d_p 10^p + \ldots + d_1 10 + d_0, \quad 0 \leq d_i \leq 9, \ d_p \neq 0 \]

\[ 1789 = 1000 + 700 + 80 + 9 = 1 \cdot 10^3 + 7 \cdot 10^2 + 8 \cdot 10^1 + 9 \cdot 10^0 \]

**En base 2**

Chiffres = 0, 1

\[ n = d_p 2^p + \ldots + d_1 2 + d_0, \quad 0 \leq d_i \leq 1, \ d_p \neq 0, \]

\[ (42)_{10} = 32 + 8 + 2 = 1 \cdot 2^5 + 0.2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \]

\[ = (101010)_2 \]
Écriture scientifique des réels

\[ x = \pm f \cdot 10^e, \quad 1/10 \leq f < 1 \]

### Exemple

- \( 825.34 = 0.82534 \cdot 10^3 \)  \( \text{écriture finie} \)
- \( 8.2534 = 0.82534 \cdot 10^1 \)
- \( 0.0082534 = 0.82534 \cdot 10^{-2} \)
- \( 1/2 = 0.5 \cdot 10^0 \)  \( \text{fraction, écriture finie} \)
- \( 1/3 = 0.333333333 \ldots \cdot 10^0 \)  \( \text{fraction, périodique} \)
- \( 4/7 = 0.5714285714285 \ldots \)  \( \text{fraction périodique} \)
- \( \pi = 0.314159265358 \ldots \cdot 10^1 \), infini, non périodique
Représentation des nombres flottants

\[ x = \pm f \cdot 2^e, \quad 2^{-1} \leq f < 1 \]

\( f \) mantisse, \( e \) exposant (entier, unique si \( x \neq 0 \))
Représentation des nombres flottants

\[ x = \pm f \cdot 2^e, \quad 2^{-1} \leq f < 1 \]

\( f \) mantisse, \( e \) exposant (entier, unique si \( x \neq 0 \))

**Nombres flottants : \( f \) sur \( t \) chiffres**

Taille mot mémoire limitée \( \Rightarrow x \in F \) ensemble fini (nombre flottants machine)

\[
f = \frac{d_1}{2} + \frac{d_2}{2^2} + \cdots + \frac{d_t}{2^t}, \quad 0 \leq d_i \leq 1, \quad d_1 \neq 0
\]

\[ = (0.d_1d_2\ldots d_t)_2 \]

Exposant \( L \leq e \leq U \)
Représentation des nombres flottants

\[ x = \pm f \cdot 2^e, \quad 2^{-1} \leq f < 1 \]

*f* mantisse, *e* exposant (entier, unique si \( x \neq 0 \))

**Nombres flottants : \( f \) sur *t* chiffres**

Taille mot mémoire limitée \( \Rightarrow x \in F \) ensemble fini (nombre flottants machine)

\[
f = \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots + \frac{d_t}{2^t}, \quad 0 \leq d_i \leq 1, \quad d_1 \neq 0
\]

\[= (0.d_1 d_2 \ldots d_t)_2\]

Exposant \( L \leq e \leq U \)

**Système caractérisé par**

Nombre de chiffres \( t \) (en base 2)

Exposants min et max \( L \) et \( U \)
Représentation des nombres flottants

\[ x = \pm f \cdot 2^e, \quad 2^{-1} \leq f < 1 \]

*f* mantisse, *e* exposant (entier, unique si \( x \neq 0 \))

**Nombres flottants : \( f \) sur \( t \) chiffres**

Taille mot mémoire limitée \( \Rightarrow x \in F \) ensemble fini (nombre flottants machine)

\[ f = \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots + \frac{d_t}{2^t}, \quad 0 \leq d_i \leq 1, \quad d_1 \neq 0 \]

\[ = (0.d_1d_2\ldots d_t)_2 \]

Exposant \( L \leq e \leq U \)

Système caractérisé par

Nombre de chiffres \( t \) (en base 2)

Exposants min et max \( L \) et \( U \)

\[ \text{Ensemble fini} \quad \text{card} F = 1 + 2^t(U - L + 1) \]
Quelques exemples

\[ \frac{3}{2} = 1 + \frac{1}{2} = 2 \left( \frac{1}{2} + \frac{1}{4} \right) = 2^1 \times 0.11 \]
Quelques exemples

- $3/2 = 1 + 1/2 = 2 \left(1/2 + 1/4\right) = 2^1 \times 0.11$
- $5/2 = 2 + 1/2 = 4 \left(1/2 + 1/8\right) = 2^2 \times 0.101$,
Quelques exemples

- $3/2 = 1 + 1/2 = 2 \left(1/2 + 1/4\right) = 2^1 \times 0.11$
- $5/2 = 2 + 1/2 = 4 \left(1/2 + 1/8\right) = 2^2 \times 0.101,$
- $1/10$, pas de représentation finie :

$$1/10 = \frac{1}{16} \frac{1}{10} = \frac{1}{16} \left(1 + \frac{3}{5}\right) = \frac{1}{16} \left(1 + \frac{9}{16} \frac{1}{1 - 1/16}\right)$$

$$= 2^{-4} \left(1 + \frac{9}{16} + \frac{9}{16^2} + \frac{9}{16^3} + \ldots\right)$$

$$= 2^{-3} \left(\frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \frac{0}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \frac{0}{2^7} + \frac{0}{2^8} + \frac{1}{2^9} + \ldots\right)$$

période
Nombre flottants

Exemple : $t = 3$, $L = -1$, $U = 2$, $\text{card} F = 33$
Nombre flottants

Exemple : \( t = 3, L = -1, U = 2, \) \( \text{card} F = 33 \)

Nombre positifs de \( F \) entre \( 1/2 \) et 1 \((e = 0)\) :

\[
\begin{align*}
1/2 &= (0.100)_2, & 3/4 &= 1/2 + 1/4 & = (0.110)_2 \\
5/8 &= 1/2 + 1/8 & = (0.101)_2, & 7/8 &= 1/2 + 1/4 + 1/8 & = (0.111)_2
\end{align*}
\]
Nombre flottants

Exemple : $t = 3$, $L = -1$, $U = 2$, $\text{card} F = 33$

Nombre positifs de $F$ entre $1/2$ et $1$ ($e = 0$) :

\[
\begin{align*}
1/2 &= (0.100)_2, & 3/4 &= 1/2 + 1/4 &= (0.110)_2 \\
5/8 &= 1/2 + 1/8 &= (0.101)_2, & 7/8 &= 1/2 + 1/4 + 1/8 &= (0.111)_2
\end{align*}
\]

Espacement variable (facteur 2 à chaque puissance de 2), mais distance relative constante $\varepsilon_{\text{mach}} = 1/8 = 1/2^3$

« Trou » important autour de 0, plus petit nombre positif de $F$ : $1/4$
Nombre flottants : le système IEEE 754

**Simple précision** (float) :

![](image)

32 bits, $t = 23 + 1$, $L = -126$, $U = 127$, $x_{\text{max}} \approx 10^{38}$, $x_{\text{min}} \approx 10^{-38}$
Nombre flottants : le système IEEE 754

**Simple précision** (*float*) :

- 8 bits pour le nombre significatif `e`
- 23 bits pour l'exposant `f`

32 bits, $t = 23 + 1$, $L = -126$, $U = 127$, $x_{\text{max}} \approx 10^{38}$, $x_{\text{min}} \approx 10^{-38}$

**Double précision** (*double*) :

- 11 bits pour le nombre significatif `e`
- 52 bits pour l'exposant `f`

64 bits, $t = 52 + 1$, $L = -1022$, $U = 1023$, $x_{\text{max}} \approx 10^{308}$, $x_{\text{min}} \approx 10^{-308}$
Propriétés de la norme IEEE 754

Utilisé par Java, processeurs Intel, PowerPC (norme internationale)

Bit caché gagne en précision

Norme précise règles d’arrondi (au plus proche, vers 0, vers $\pm\infty$)

Il existe $\pm 0$, $\pm\infty$

Nombres dénormalisés (entre 0 et $x_{\text{min}}$)

NaN = « Not a Number » pour $0/0$, $\infty/\infty$, fonction $\text{isnan}(x)$
Approcher $x \in \mathbb{R}$ par $fl(x) \in F : (x > 0)$

Arrondi au plus proche $fl(x)$ est l’élément de $F$ le plus proche de $x$

\[
\frac{|x - fl(x)|}{|x|} \leq 2^{-t}
\]
Arrondi – epsilon machine

Approcher $x \in \mathbb{R}$ par $fl(x) \in F : (x > 0)$

Arrondi au plus proche $fl(x)$ est l’élément de $F$ le plus proche de $x$

\[
\frac{|x - fl(x)|}{|x|} \leq 2^{-t}
\]

Exemple (5 chiffres significatifs) : $x = \sqrt{7} \approx 2.6457513 \ldots$
Arrondi $fl(x) = 2.6458$
Arrondi – epsilon machine

Approcher $x \in \mathbb{R}$ par $fl(x) \in F : (x > 0)$

Arrondi au plus proche $fl(x)$ est l'élément de $F$ le plus proche de $x$

\[
\frac{|x - fl(x)|}{|x|} \leq 2^{-t}
\]

Exemple (5 chiffres significatifs) : $x = \sqrt{7} \approx 2.6457513 \ldots$

Arrondi $fl(x) = 2.6458$

$\varepsilon_{mach} = 2^{-t}$ caractéristique de l'arithmétique.

$fl(x) = x(1 + \varepsilon)$, $|\varepsilon| \leq \varepsilon_{mach}$

Calculatrice $\varepsilon_{mach} \approx 10^{-10}$

Avec Scilab $\varepsilon_{mach} = 2^{-53} \approx 1.11 \times 10^{-16} \approx 16$ chiffres

(Scilab : "%eps" = $2\varepsilon_{mach} \approx 2.22 \times 10^{-16}$).
1. Introduction

2. Représentation des nombres

3. Calculs en précision limitée
Calcul sur les nombres flottants

En général le résultat exact d’une opération sur deux flottants n’est pas un flottant machine

Exemple

En base 2, avec 3 chiffres significatifs ($t = 3$) :

$$\frac{5}{8} + \frac{3}{4} = \frac{11}{8} = 1 + \frac{3}{8} \notin F$$

$$0.101 + 0.110 = 0.1011$$

Le dernier chiffre (en rouge) ne peut pas être pris en compte.
Calcul sur les nombres flottants

En général le résultat exact d’une opération sur deux flottants n’est pas un flottant machine

Exemple

En base 2, avec 3 chiffres significatifs ($t = 3$) :

\[
\begin{align*}
5/8 + 3/4 &= 11/8 = 1 + 3/8 \\
\text{et } 1.011_2 + 0.110_2 &= 1.011_2
\end{align*}
\]

Le dernier chiffre (en rouge) ne peut pas être pris en compte.

Axiome  Pour $(x, y) \in F^2 : (x \otimes y)$ est l’arrondi de la valeur exacte de $x \ast y$, si $\ast \in \{+,-,\times,\div,\sqrt{\cdot}\}$

\[
(x \otimes y) = fl(x \ast y)
\]

Permet de raisonner sur les calculs flottants
L’arithmétique flottante est commutative, et non associative.

Exemple (arithmétique base 10, 7 chiffres) :

\[
a = 0.1234567, \quad b = 0.4711325 \cdot 10^4, \quad c = -b
\]

\[
b \oplus c = 0, \quad (a \oplus (b \oplus c)) = a = 0.1234567
\]

\[
\begin{array}{ccc}
0.47113250000 & 10^4 & 0.4711448 & 10^4 \\
+ 0.00001234567 & 10^4 & -0.4711325 & 10^4 \\
\hline
0.47114484567 & 10^4 & 0.0000123 & 10^4
\end{array}
\]

\[
(a \oplus b) = 0.4711448 \cdot 10^4, \quad (a \oplus b) \oplus c = 0.123
\]
Propriétés de l’arithmétique flottante

L’arithmétique flottante est commutative, et non associative

Exemple (arithmétique base 10, 7 chiffres) :

\[ a = 0.1234567, \quad b = 0.4711325 \cdot 10^4, \quad c = -b \]

\[ b \oplus c = 0, \quad (a \oplus (b \oplus c)) = a = 0.1234567 \]

\[
\begin{array}{c@{\quad}c@{\quad}c}
0.47113250000 & 10^4 & 0.4711448 & 10^4 \\
+ 0.00001234567 & 10^4 & - 0.4711325 & 10^4 \\
\hline
0.47114484567 & 10^4 & 0.0000123 & 10^4
\end{array}
\]

\[ (a \oplus b) = 0.4711448 \cdot 10^4, \quad (a \oplus b) \oplus c = 0.123 \]

Soustraction de deux nombres voisins

\[ a = 0.1234567, \quad b = 0.1234560, \quad a \ominus b = 0.7 \cdot 10^{-6} \text{ (exact).} \]

Si \( a \) et \( b \) sont connus à 6 chiffres près, \( a \ominus b \) n’a qu’un chiffre significatif : révèle une perte de précision dans un calcul précédent.
Annulation destructrice

$a = 123456, \ b = 12.3456, \ c = 123450$, arithmétique (décimale) avec 6 chiffres. Calcul de $a + b - c = 18.3456$ (résultat exact).

\[
\begin{array}{c}
123456.0000 \\
+ \quad 12.3456 \\
\hline
123468.0000 \\
\end{array}
\]
puis
\[
\begin{array}{c}
123468. \\
- \quad 123450. \\
\hline
18. \\
\end{array}
\]

L'annulation révèle une perte d'information précédente (même résultat pour $b \in [11.5, 12.5]$).

Autre ordre $a \ominus c = 6$, puis $b \oplus (a \ominus c) = 18.3456$, exact.
Annulation destructrice

\[ a = 123456, \ b = 12.3456, \ c = 123450, \ \text{arithmétique (décimale) avec 6 chiffres. Calcul de } a + b - c = 18.3456 (\text{résultat exact}). \]

\[
\begin{array}{c}
123456.0000 \\
+ 12.3456 \\
\hline
123468.0000
\end{array}
\quad \text{puis} \quad
\begin{array}{c}
123468.0000 \\
- 123450.0000 \\
\hline
18.0000
\end{array}
\]

Annulation destructrice : seulement deux chiffres exacts. Erreur d’arrondi dans la première opération, la seconde est exacte. L’annulation révèle une perte d’information précédente (même résultat pour \( b \in [11.5, 12.5[ \)).
Annulation destructrice

\( a = 123456, \ b = 12.3456, \ c = 123450, \) arithmétique (décimale) avec 6 chiffres. Calcul de \( a + b - c = 18.3456 \) (résultat exact).

\[
\begin{array}{c}
123456.0000 \\
+ 12.3456 \\
\hline
123468.0000
\end{array}
\quad \text{puis} \quad
\begin{array}{c}
123468.0000 \\
- 123450.0000 \\
\hline
18.0000
\end{array}
\]

Annulation destructrice : seulement deux chiffres exacts. Erreur d'arrondi dans la première opération, la seconde est exacte. L'annulation révèle une perte d'information précédente (même résultat pour \( b \in [11.5, 12.5] \)).

Autre ordre
\( a \ominus c = 6, \) puis \( b \oplus (a \ominus c) = 18.3456, \) exact.
Équations du second degré

Calcul des racines de $x^2 - 2px + 1$, quand $p \gg 1$ (ex : $p = 10^7$)
Équations du second degré

Calcul des racines de $x^2 - 2px + 1$, quand $p \gg 1$ (ex : $p = 10^7$)

**Algorithme 1**

$x^+ = p + \sqrt{p^2 - 1}$

$x^- = p - \sqrt{p^2 - 1}$

**Algorithme 2**

$x^+ = p + \sqrt{p^2 - 1}$

$x^- = 1/(p + \sqrt{p^2 - 1})$
Équations du second degré

Calcul des racines de \( x^2 - 2px + 1 \), quand \( p \gg 1 \) (ex : \( p = 10^7 \))

**Algorithme 1**

\[
\begin{align*}
x^+ &= p + \sqrt{p^2 - 1} \\
x^- &= p - \sqrt{p^2 - 1}
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
x^+ &= 2.000000000000000 \times 10^7 \\
x^- &= 5.0291419029236 \times 10^{-8}
\end{align*}
\]

**Algorithme 2**

\[
\begin{align*}
x^+ &= p + \sqrt{p^2 - 1} \\
x^- &= 1/(p + \sqrt{p^2 - 1})
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
x^+ &= 2.000000000000000 \times 10^7 \\
x^- &= 5.000000000000000 \times 10^{-8}
\end{align*}
\]
Équations du second degré

Calcul des racines de $x^2 - 2px + 1$, quand $p \gg 1$ (ex : $p = 10^7$)

**Algorithme 1**

\[
\begin{align*}
    x^+ &= p + \sqrt{p^2 - 1} \\
    x^- &= p - \sqrt{p^2 - 1}
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
    x^+ &= 2.00000000000000 10^7 \\
    x^- &= 5.0291419029236 10^{-8}
\end{align*}
\]

Algorithme **instable**

**Algorithme 2**

\[
\begin{align*}
    x^+ &= p + \sqrt{p^2 - 1} \\
    x^- &= 1/(p + \sqrt{p^2 - 1})
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
    x^+ &= 2.00000000000000 10^7 \\
    x^- &= 5.00000000000000 10^{-8}
\end{align*}
\]

Algorithme **stable**
Équations du second degré

Calcul des racines de $x^2 - 2px + 1$, quand $p \gg 1$ (ex : $p = 10^7$)

**Algorithm 1**

\[
\begin{align*}
x^+ &= p + \sqrt{p^2 - 1} \\
x^- &= p - \sqrt{p^2 - 1}
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
x^+ &= 2.000000000000000 \times 10^7 \\
x^- &= 5.0291419029236 \times 10^{-8}
\end{align*}
\]

Algorithme instable

**Algorithm 2**

\[
\begin{align*}
x^+ &= p + \sqrt{p^2 - 1} \\
x^- &= 1/(p + \sqrt{p^2 - 1})
\end{align*}
\]

Avec Scilab

\[
\begin{align*}
x^+ &= 2.000000000000000 \times 10^7 \\
x^- &= 5.000000000000000 \times 10^{-8}
\end{align*}
\]

Algorithme stable

Solutions exactes :

\[
\begin{align*}
x^+ &= 1.999999999999995 \times 10^7, \\
x^- &= 5.0000000000000125 \times 10^{-8}
\end{align*}
\]
Une suite curieuse (très simple)

\[ u_{n+1} = \alpha u_n + \beta, \quad n = 0, 1, \ldots, \quad u_0 \text{ donné} \]

Solution : \( u_n = \alpha^n u_0 + \frac{\alpha^{n-1} \beta}{\alpha - 1} \).

On prend : \( \alpha = 4, \beta = -1 : u_n = 1/3 + 4^n (u_0 - 1/3) \).

Si \( u_0 = 1/3 \), alors la suite est constante : \( u_n = 1/3, \forall n \). Pourtant...
Une suite curieuse (très simple)

\[ u_{n+1} = \alpha u_n + \beta, \quad n = 0, 1, \ldots, \quad u_0 \text{ donné} \]

Solution : \( u_n = \alpha^n u_0 + \frac{\alpha^{n-1} - 1}{\alpha - 1} \beta. \)

On prend : \( \alpha = 4, \quad \beta = -1 : u_n = \frac{1}{3} + 4^n (u_0 - \frac{1}{3}). \)

Si \( u_0 = \frac{1}{3}, \) alors la suite est *constante* : \( u_n = \frac{1}{3}, \forall n. \) Pourtant...

\[
\begin{array}{cccc}
0 & 0.333333333333 & 24 & 0.328125 \\
1 & 0.333333333333 & 25 & 0.3125 \\
2 & 0.333333333333 & 26 & 0.25 \\
2 & \ldots. & 27 & 0.0 \\
11 & 0.33333333333255 & 28 & -1.0 \\
23 & 0.33203125 & 29 & -5.0 \\
30 & -21.0 \\
\end{array}
\]
Une suite curieuse (très simple)

\[ u_{n+1} = \alpha u_n + \beta, \quad n = 0, 1, \ldots, \quad u_0 \text{ donné} \]

Solution : \( u_n = \alpha^n u_0 + \frac{\alpha^n - 1}{\alpha - 1} \beta \).

On prend : \( \alpha = 4, \beta = -1 : u_n = 1/3 + 4^n (u_0 - 1/3) \).

Si \( u_0 = 1/3 \), alors la suite est constante : \( u_n = 1/3, \forall n \). Pourtant...

\[
\begin{array}{cccc}
0 & 0.333333333333 & 24 & 0.328125 \\
1 & 0.333333333333 & 25 & 0.3125 \\
2 & 0.333333333333 & 26 & 0.25 \\
\vdots & \vdots & 27 & 0.0 \\
11 & 0.333333333333 & 28 & -1.0 \\
\vdots & \vdots & 29 & -5.0 \\
23 & 0.33203125 & 30 & -21.0 \\
\end{array}
\]

Si \( u_0 = 1/3(1 - \delta) \) avec \( \delta \approx \varepsilon_{\text{mach}} \), alors \( u_n = 1/3(1 - 4^n \delta) \rightarrow -\infty !! \)