Introduction au calcul flottant

MT09
Vincent.Martin@utc.fr

UTC Compiègne
France

UTC, A2019

Plan

Plan

Une suite curieuse

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

$$u_n = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}}, \qquad u_0 = 2,\ u_1 = -4$$

3	18.5000000000000	17	7.2350211655349
4	9.37837837837838	18	22.0620784635258
5	7.80115273775217	19	78.5755748878722
6	7.15441448097533	20	98.3495031221654
7	6.80678473692481	21	99.8985692661829
8	6.59263276872179	22	99.9938709889028
9	6.44946593405393	23	99.9996303872863
10	6.34845206074662	24	99.9999777306795
11	6.27443866272812	25	99.9999986592167
12	6.21869676858216	26	99.9999999193218
13	6.17585385581539	27	99.9999999951478
14	6.14262717048101	28	99.9999999997083
15	6.12024870457016	29	99.9999999999825
16	6.16608655959810	30	99.9999999999989

Une suite curieuse

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

$$u_n = 111 - \frac{1130}{u_{n-1}} + \frac{3000}{u_{n-1}u_{n-2}}, \qquad u_0 = 2,\ u_1 = -4$$

3	18.5000000000000	17	7.2350211655349
4	9.37837837837838	18	22.0620784635258
5	7.80115273775217	19	78.5755748878722
6	7.15441448097533	20	98.3495031221654
7	6.80678473692481	21	99.8985692661829
8	6.59263276872179	22	99.9938709889028
9	6.44946593405393	23	99.9996303872863
10	6.34845206074662	24	99.9999777306795
11	6.27443866272812	25	99.9999986592167
12	6.21869676858216	26	99.9999999193218
13	6.17585385581539	27	99.9999999951478
14	6.14262717048101	28	99.9999999997083
15	6.12024870457016	29	99.9999999999825
16	6.16608655959810	30	99.9999999999989

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

Écriture des entiers en base 2

En base 10

Chiffres $0, 1, \ldots, 9$

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

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

Écriture des entiers en base 2

En base 10

Chiffres $0, 1, \ldots, 9$

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

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

En base 2

Chiffres $= 0, 1$

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

$$\begin{aligned}(42)_{10} &= 32 + 8 + 2 = 1 \cdot 2^5 + 0.2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 \\ &= (101010)_2\end{aligned}$$

Écriture scientifique des réels

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

Exemple

- $825.34 = 0.82534\,10^3$ écriture finie
- $8.2534 = 0.82534\,10^1$
- $0.0082534 = 0.82534\,10^{-2}$

- $1/2 = 0.5\,10^0$ fraction, écriture finie
- $1/3 = 0.333333333\ldots\,10^0$ fraction, périodique
- $4/7 = 0.5714285714285\ldots$ fraction périodique

- $\pi = 0.314159265358\ldots\,10^1$, 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)

$$\begin{aligned} f &= \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots + \frac{d_t}{2^t}, \qquad 0 \leq d_i \leq 1, \qquad d_1 \neq 0 \\ &= (0.d_1 d_2 \ldots d_t)_2 \end{aligned}$$

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)

$$f = \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots + \frac{d_t}{2^t}, \qquad 0 \leq d_i \leq 1, \qquad d_1 \neq 0$$
$$= (0.d_1 d_2 \ldots d_t)_2$$

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)

$$f = \frac{d_1}{2} + \frac{d_2}{2^2} + \ldots + \frac{d_t}{2^t}, \qquad 0 \leq d_i \leq 1, \qquad d_1 \neq 0$$
$$= (0.d_1 d_2 \ldots d_t)_2$$

Exposant $L \leq e \leq U$

Système caractérisé par

Nombre de chiffres t (en base 2)

Exposants min et max L et U

Ensemble fini

$$\text{card} F = 1 + 2^t(U - L + 1)$$

Quelques exemples

- $3/2 = 1 + 1/2 = 2\,(1/2 + 1/4) = 2^1 \times 0.11$

Quelques exemples

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

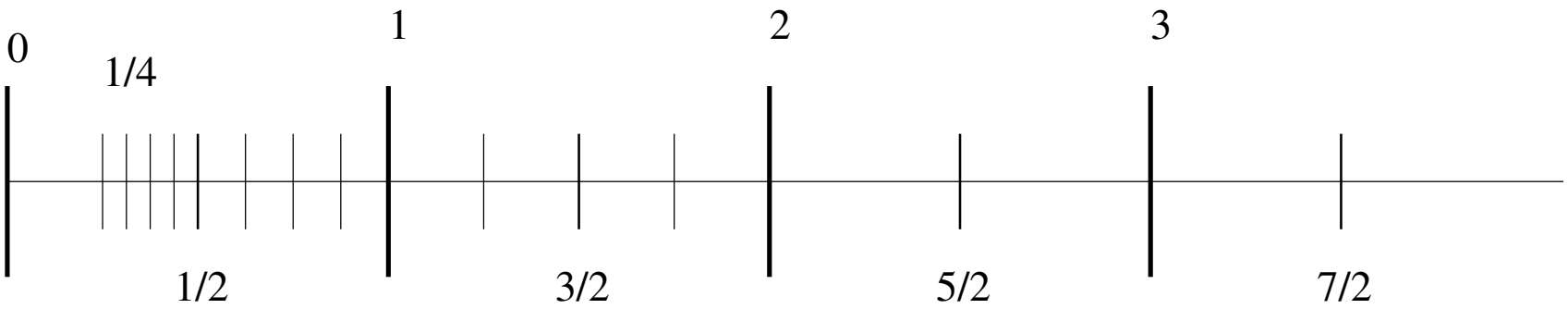
Quelques exemples

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

$$
\begin{aligned}
1/10 = \frac{1}{16}\frac{16}{10} &= \frac{1}{16}(1 + \frac{3}{5}) = \frac{1}{16}(1 + \frac{9}{16}\,\frac{1}{1 - 1/16}) \\
&= 2^{-4}\left(1 + \frac{9}{16} + \frac{9}{16^2} + \frac{9}{16^3} + \cdots\right) \\
&= 2^{-3}\left(\frac{1}{2} + \frac{1}{2^2} + \frac{0}{2^3} + \underbrace{\frac{0}{2^4} + \frac{1}{2^5} + \frac{1}{2^6} + \frac{0}{2^7} + \frac{0}{2^8} + \frac{1}{2^9}}_{\text{période}} + \cdots\right)
\end{aligned}
$$

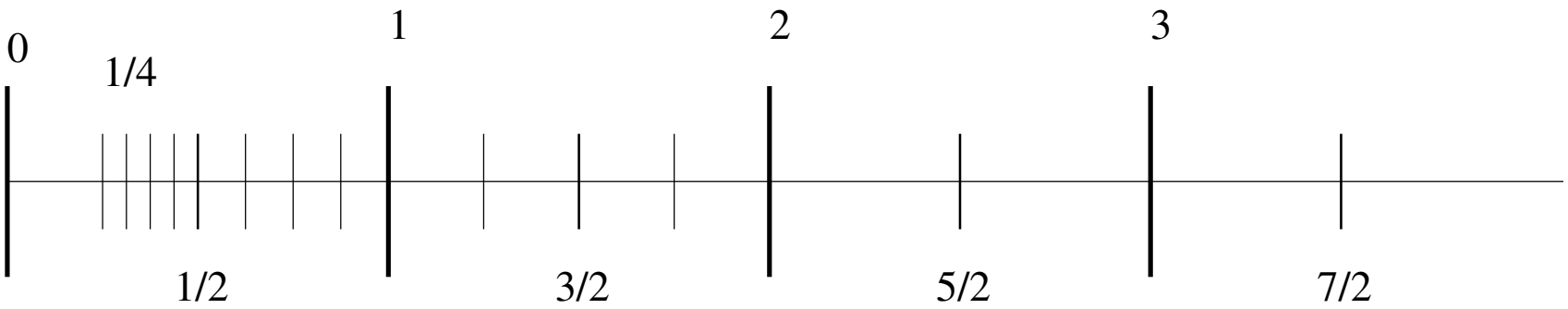
Nombre flottants

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



Nombre flottants

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

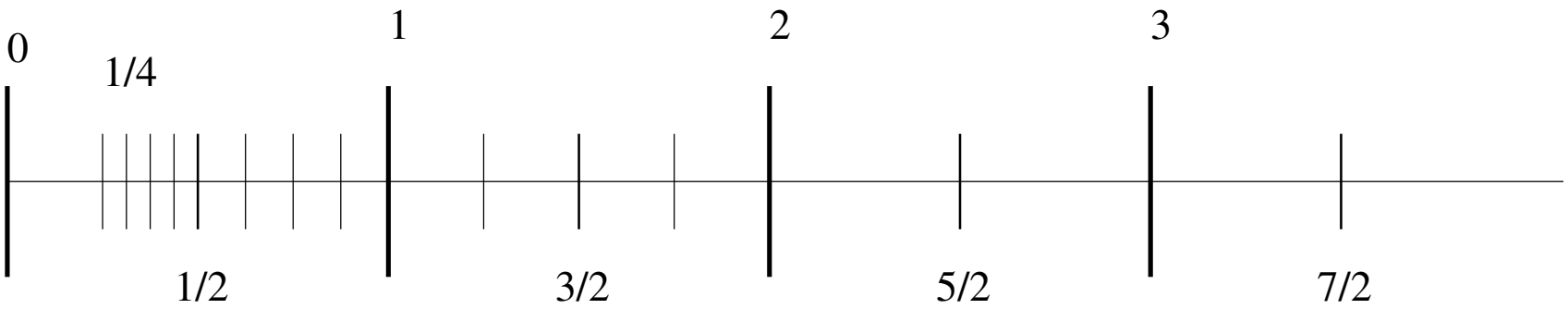


Nombre positifs de F entre 1/2 et 1 ($e = 0$) :

$$1/2 = (0.100)_2, \quad 3/4 = 1/2 + 1/4 = (0.110)_2$$
$$5/8 = 1/2 + 1/8 = (0.101)_2, \quad 7/8 = 1/2 + 1/4 + 1/8 = (0.111)_2$$

Nombre flottants

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



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

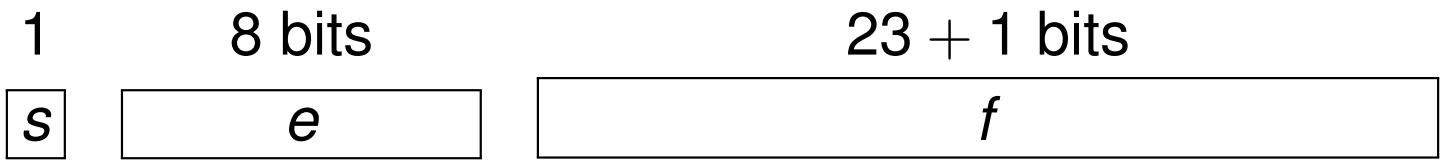
$$1/2 = (0.100)_2, \quad 3/4 = 1/2 + 1/4 = (0.110)_2$$
$$5/8 = 1/2 + 1/8 = (0.101)_2, \quad 7/8 = 1/2 + 1/4 + 1/8 = (0.111)_2$$

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

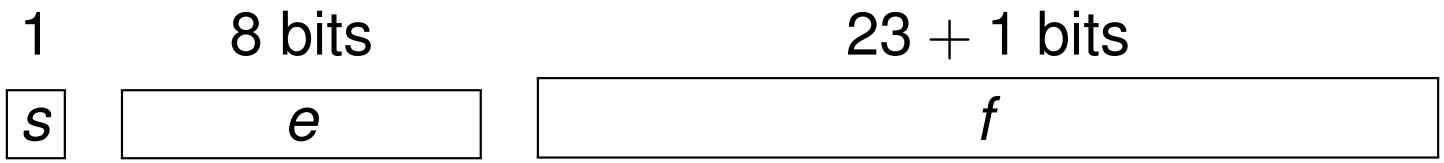
Simple précision (`float`) :



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

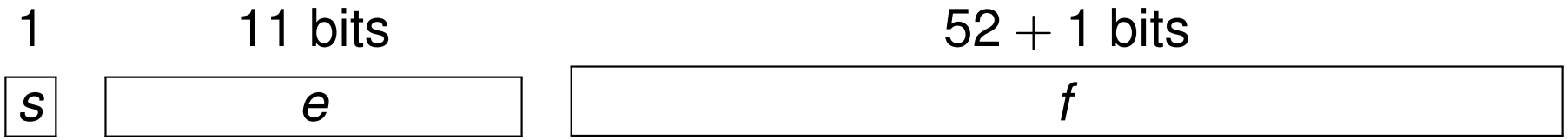
Nombre flottants : le système IEEE 754

Simple précision (`float`) :



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

Double précision (`double`) :



64 bits, $t = 52 + 1$, $L = -1022$, $U = 1023$, $x_{\max} \approx 10^{308}$, $x_{\min} \approx 10^{-308}$

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 ± 0, $\pm\infty$

Nombres dénormalisés (entre 0 et x_{min})

`NaN` = « Not a Number » pour $0/0$, ∞/∞, fonction `isnan(x)`

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

$$\boxed{\frac{|x - fl(x)|}{|x|} \leq 2^{-t}}$$

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

$$\boxed{\frac{|x - fl(x)|}{|x|} \leq 2^{-t}}$$

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

$$\boxed{\frac{|x - fl(x)|}{|x|} \leq 2^{-t}}$$

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

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

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

Calculatrice $\varepsilon_{\text{mach}} \approx 10^{-10}$
Avec Scilab $\varepsilon_{\text{mach}} = 2^{-53} \approx 1.11\,10^{-16} \approx$ 16 chiffres
(Scilab : $''\%eps'' = 2\varepsilon_{\text{mach}} \approx 2.22\,10^{-16}$).

Plan

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 ($t = 3$) :

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

$$\begin{array}{rl} & 0.101 \\ + & 0.110 \\ \hline & 1.011 \\ = & 0.1011\ 2^1 \end{array}$$

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 ($t = 3$) :

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

$$\begin{array}{rl} & 0.101 \\ + & 0.110 \\ \hline & 1.011 \\ = & 0.1011\ 2^1 \end{array}$$

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

Axiome Pour $(x, y) \in F^2$: $(x \circledast y)$ est l'arrondi de la valeur exacte de $x * y$, si $* \in \{+, -, \times, \backslash, \sqrt{}\}$

$$(x \circledast y) = fl(x * y)$$

Permet de raisonner sur les calculs flottants

Propriétés de l'arithmétique flottante

L'arithmétique flottante est commutatitve, et non associative

Exemple (arithmétique base 10, 7 chiffres) :
$a = 0.1234567,\ b = 0.4711325 \cdot 10^4,\ c = -b$
$b \oplus c = 0, \quad (a \oplus (b \oplus c)) = a = 0.1234567$

$$\begin{array}{rrr} 0.47113250000 & & 10^4 \\ + \ 0.00001234567 & & 10^4 \\ \hline 0.4711448\,4567 & & 10^4 \end{array} \qquad \begin{array}{rrr} 0.4711448 & & 10^4 \\ - \ 0.4711325 & & 10^4 \\ \hline 0.0000123 & & 10^4 \end{array}$$

$(a \oplus b) = 0.4711448\,10^4, \quad (a \oplus b) \oplus c = 0.123$

Propriétés de l'arithmétique flottante

L'arithmétique flottante est commutatitve, et non associative

Exemple (arithmétique base 10, 7 chiffres) :
$a = 0.1234567,\ b = 0.4711325 \cdot 10^4,\ c = -b$
$b \oplus c = 0, \quad (a \oplus (b \oplus c)) = a = 0.1234567$

$$\begin{array}{rll} 0.47113250000 & & 10^4 \\ + \ 0.00001234567 & & 10^4 \\ \hline 0.4711448\,4567 & & 10^4 \end{array} \qquad \begin{array}{rll} 0.4711448 & & 10^4 \\ - \ 0.4711325 & & 10^4 \\ \hline 0.0000123 & & 10^4 \end{array}$$

$(a \oplus b) = 0.4711448\,10^4, \quad (a \oplus b) \oplus c = 0.123$

Soustraction de deux nombres voisins
$a = 0.1234567,\ b = 0.1234560,\ a \ominus b = 0.7\,10^{-6}$ (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).

$$\begin{array}{rr} & 123456.0000 \\ + & 12.3456 \\ \hline & 123468.0000 \end{array}$$

puis

$$\begin{array}{rr} & 123468. \\ - & 123450. \\ \hline & 18. \end{array}$$

Annulation destructrice

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

$$\begin{array}{rr} & 123456.0000 \\ + & 12.3456 \\ \hline & 123468.0000 \end{array} \quad \text{puis} \quad \begin{array}{rr} & 123468. \\ - & 123450. \\ \hline & 18. \end{array}$$

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).

$$\begin{array}{rr} & 123456.0000 \\ + & 12.3456 \\ \hline & 123468.0000 \end{array} \quad \text{puis} \quad \begin{array}{rr} & 123468. \\ - & 123450. \\ \hline & 18. \end{array}$$

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

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0291419029236\,10^{-8}$$

Algorithme 2

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0000000000000\,10^{-8}$$

Équations du second degré

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

Algorithme 1

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0291419029236\,10^{-8}$$

Algorithme instable

Algorithme 2

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0000000000000\,10^{-8}$$

Algorithme stable

Équations du second degré

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

Algorithme 1

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0291419029236\,10^{-8}$$

Algorithme instable

Algorithme 2

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

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

Avec Scilab

$$x^+ = 2.0000000000000\,10^7$$

$$x^- = 5.0000000000000\,10^{-8}$$

Algorithme stable

Solutions exactes :

$x^+ = 1.9999999999999 5\,10^7$, $x^- = 5.0000000000001250\,10^{-8}$

Une suite curieuse (très simple)

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

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...

Une suite curieuse (très simple)

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

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...

```
0  0.333333333333
1  0.333333333333
2  0.333333333333
     ....
11  0.333333333255
   .....
23  0.33203125
```

```
24   0.328125
25   0.3125
26   0.25
27   0.0
28  -1.0
29  -5.0
30  -21.0
```

Une suite curieuse (très simple)

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

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...

```
0  0.333333333333
1  0.333333333333
2  0.333333333333
     ....
11  0.333333333255
   .....
23  0.33203125
```

```
24   0.328125
25   0.3125
26   0.25
27   0.0
28  -1.0
29  -5.0
30  -21.0
```

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