IEE 754 : niveau 2

Méthodes numériques pour l'info - semaines 17 & 18

Guillaume CONNAN

IUT de Nantes - Département INFO

avril 2016

UNIVERSITÉ DE NANTES

Sommaire

UNIVERSITÉ DE NANTES

UNIVERSITÉ DE NANTES

UNIVERSITÉ DE NANTES

UNIVERSITÉ DE NANTES

```
In [1]: 3 * 0.1
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 3 * 0.1
Out[1]: 0.30000000000000004
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 3 * 0.1
Out[1]: 0.30000000000000004
In [2]: sum([0.1 for k in range(10000000)])
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 3 * 0.1
Out[1]: 0.30000000000000004
In [2]: sum([0.1 for k in range(10000000)])
Out[2]: 999999.9998389754
```

UNIVERSITÉ DE NANTES

LES NOMBRES RÉELS N'EXISTENT PAS.

UNIVERSITÉ DE NANTES

LES NOMBRES RÉELS N'EXISTENT PAS.

TOUT CE QUE VOUS AVEZ VU EN MATHS N'EST QU'ILLUSION !

UNIVERSITÉ DE NANTES

William Kahan (1933)

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32* $(\#E, \#m) = (8, 24)$
- *binary64*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32* $(\#E, \#m) = (8, 24)$
- *binary64*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32* $(\#E, \#m) = (8, 24)$
- *binary64* $(\#E, \#m) = (11, 53)$.
- *toy7*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32* $(\#E, \#m) = (8, 24)$
- *binary64* $(\#E, \#m) = (11, 53)$.
- *toy7*

UNIVERSITÉ DE NANTES

$$v = (-1)^s \times m \times 2^E$$

- un bit de signe s qui vaut 0 si le nombre est positif, 1 sinon ;
- 11 bits d'exposant décalé e ;
- 53 bits de mantisse (ou « significande » en franglais) m, $\quad 1 \leqslant m$.

- *binary32* $(\#E, \#m) = (8, 24)$
- *binary64* $(\#E, \#m) = (11, 53)$.
- *toy7* $(\#E, \#m) = (3, 4)$.

UNIVERSITÉ DE NANTES

Oups ! $1 + 11 + 53 = 65$...En fait, on représente un nombre de $\mathbb{V}_{64}$ sous **forme normale** :

$$v = (-1)^s \times 1, f \times 2^E$$

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.
Elle correspond aux exposants réels de E_{min} jusqu'à 0.

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.
Elle correspond aux exposants réels de E_{min} jusqu'à 0.
La deuxième de $2^{\#E-1}$ à $2^{\#E} - 1$.

UNIVERSITÉ DE NANTES

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.
Elle correspond aux exposants réels de E_{min} jusqu'à 0.
La deuxième de $2^{\#E-1}$ à $2^{\#E} - 1$.
Elle correspond aux exposants de 1 jusqu'à E_{max}.

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.
Elle correspond aux exposants réels de E_{min} jusqu'à 0.
La deuxième de $2^{\#E-1}$ à $2^{\#E} - 1$.
Elle correspond aux exposants de 1 jusqu'à E_{max}.
Il suffit donc de translater les exposants réels de $2^{\#E-1} - 1$...

Sur $\#E$ bits on peut coder $2^{\#E}$ nombres.
La première moitié va donc de 0 à $2^{\#E-1} - 1$.
Elle correspond aux exposants réels de E_{min} jusqu'à 0.
La deuxième de $2^{\#E-1}$ à $2^{\#E} - 1$.
Elle correspond aux exposants de 1 jusqu'à E_{max}.
Il suffit donc de translater les exposants réels de $2^{\#E-1} - 1$...

$$e_{\text{stocké}} = E_{\text{réel}} + 2^{\#E-1} - 1$$

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E$$

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E = -1$$

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E = -1 \leftrightarrow e = -1 + 3 = 2 = 10_2$$

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E = -1 \leftrightarrow e = -1 + 3 = 2 = 10_2$$

s (1 bit)	e (3 bits)			f (3 bits)		
0	0	1	0	1	0	0

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E = -1 \leftrightarrow e = -1 + 3 = 2 = 10_2$$

s (1 bit)	e (3 bits)			f (3 bits)		
0	0	1	0	1	0	0

UNIVERSITÉ DE NANTES

0,75 en toy7

$$0,75_{10} = \frac{3_{10}}{2^2_{10}} = 11_2 \times 2^{-2} = 1,100_2 \times 2^{-1}$$

$$e - \left(2^{3-1} - 1\right) = E = -1 \leftrightarrow e = -1 + 3 = 2 = 10_2$$

s (1 bit)	e (3 bits)			f (3 bits)		
0	0	1	0	1	0	0

UNIVERSITÉ DE NANTES

0,75 en toy7 : méthode générale

Comment écrire en base 2 la partie fractionnaire d'un nombre en base 10 ?

UNIVERSITÉ DE NANTES

0,75 en toy7 : méthode générale

Comment écrire en base 2 la partie fractionnaire d'un nombre en base 10 ?

$$0,75 = b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots$$

UNIVERSITÉ DE NANTES

0,75 en toy7 : méthode générale

Comment écrire en base 2 la partie fractionnaire d'un nombre en base 10 ?

$$0,75 = b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots$$

$$2 \times 0,75 = b_1 + b_2 \times 2^{-1} + \cdots$$

UNIVERSITÉ DE NANTES

0,75 en toy7 : méthode générale

Comment écrire en base 2 la partie fractionnaire d'un nombre en base 10 ?

$$0,75 = b_1 \times 2^{-1} + b_2 \times 2^{-2} + \cdots$$

$$2 \times 0,75 = b_1 + b_2 \times 2^{-1} + \cdots$$

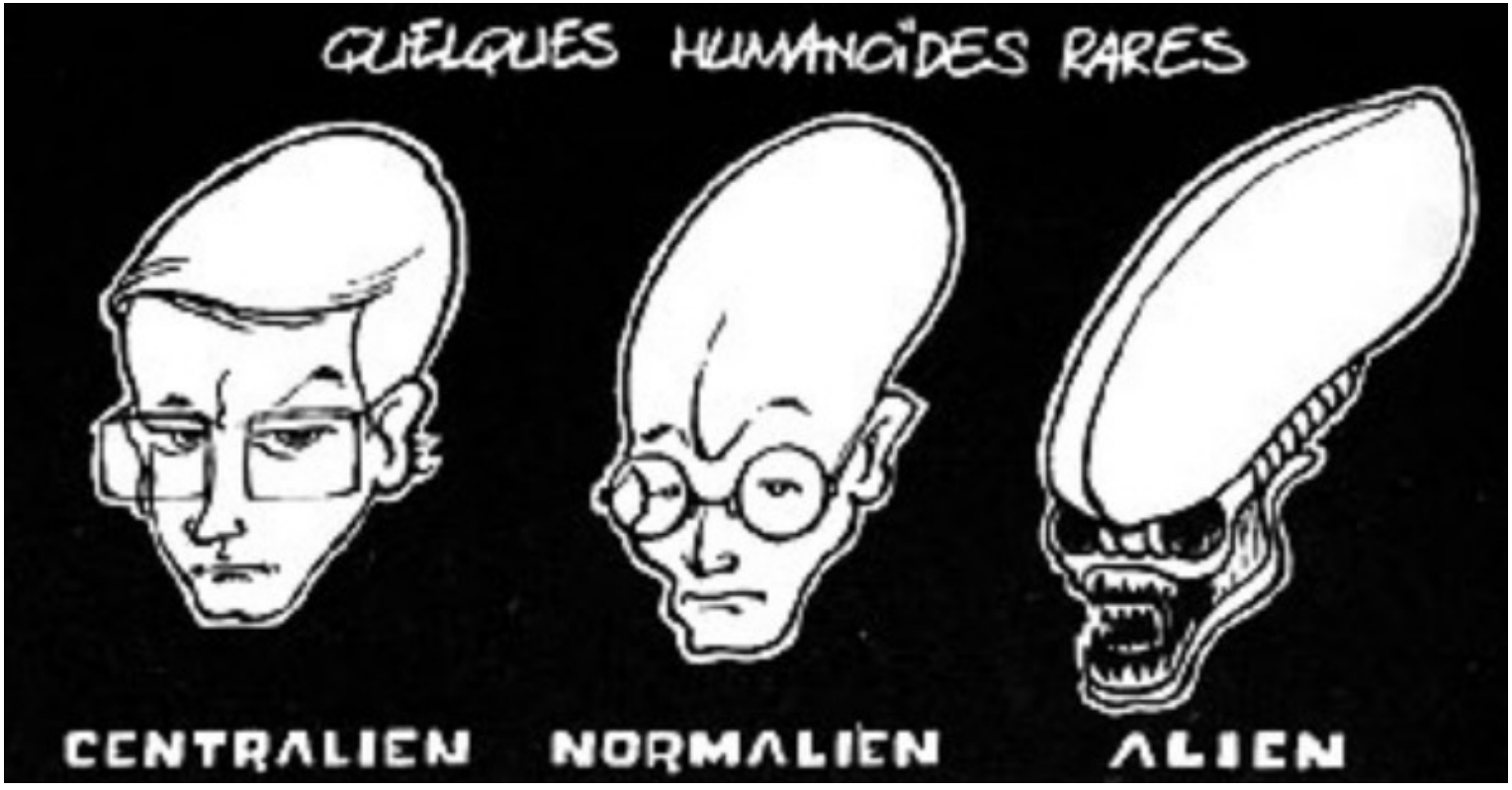
On voit poindre un bel algo...

UNIVERSITÉ DE NANTES

Exercice 1

Pourquoi le Patriot a raté son Scud de 600m ?
Représentez 1/10 avec une mantisse de 24 bits et tronquez le résultat.
Calculez l'erreur en seconde puis après 100 heures d'utilisation.
Sachant qu'un Scud vole à $1676\,m\,s^{-1}$*, quelle est environ l'erreur commise en mètres ?*
Et si on avait arrondi au plus proche ?

UNIVERSITÉ DE NANTES



UNIVERSITÉ DE NANTES

s (1 bit)	e (3 bits)			f (3 bits)		
0	0	0	0	0	0	0

UNIVERSITÉ DE NANTES

s (1 bit)	e (3 bits)			f (3 bits)		
0	0	0	0	0	0	0

s (1 bit)	e (3 bits)			f (3 bits)		
1	0	0	0	0	0	0

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

.
.
.
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
.
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
Out[3]: inf
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
Out[3]: inf
In [4]: 1/x
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
Out[3]: inf
In [4]: 1/x
Out[4]: 0.0
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
Out[3]: inf
In [4]: 1/x
Out[4]: 0.0
In [5]: 4 - x
.
```

UNIVERSITÉ DE NANTES

```
In [1]: x = 1e500

In [2]: 1+x
Out[2]: inf
In [3]: x**3
Out[3]: inf
In [4]: 1/x
Out[4]: 0.0
In [5]: 4 - x
Out[5]: -inf
```

UNIVERSITÉ DE NANTES

$$\lim_{x \to +\infty} x^2 - x$$

$$\lim_{x \to +\infty} x^2 - x$$

```
In [9]: x**2 - x
.
```

UNIVERSITÉ DE NANTES

$$\lim_{x \to +\infty} x^2 - x$$

```
In [9]: x**2 - x
Out[9]: nan
```

UNIVERSITÉ DE NANTES


NAN (PAIN INDIEN FOURRÉ AU FROMAGE)
MENU
cheese NAN KEBAB
MENU
cheese NAN CHIK'N

```
In [10]: x
Out[10]: inf

In [11]: x - x
Out[11]: nan

In [12]: x / x
Out[12]: nan

In [13]: x - x == x - x
Out[13]: False

In [14]: x**2 - x
Out[14]: nan

In [15]: x * (x - 1)
Out[15]: inf

In [16]: x * (x - 1) == x**2 - x
Out[16]: False
```

UNIVERSITÉ DE NANTES

Exercice 2

1. *Comment expliquer les résultats suivants :*

```
*Main> let f(x) = x^2 / sqrt(x^3 + 1)
*Main> f(1e100)
1.0e50
*Main> f(1e150)
0.0
*Main> f(1e200)
NaN
```

2. *Comment éviter le dernier NaN ?*

UNIVERSITÉ DE NANTES

```
In [16]: f = lambda x: x**2 / sqrt(x**3 + 1)

In [17]: f(1e100)
Out[17]: 1e+50

In [18]: f(1e150)
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call
         ↪   last)
<ipython-input-18-79775d85f31a> in <module>()
----> 1 f(1e150)

<ipython-input-16-eed0b81ef932> in <lambda>(x)
----> 1 f = lambda x: x**2 / sqrt(x**3 + 1)

OverflowError: (34, 'Numerical result out of range')

In [19]: f(1e200)
---------------------------------------------------------------------------
OverflowError                             Traceback (most recent call
         ↪   last)
<ipython-input-19-c1d077e9d28e> in <module>()
----> 1 f(1e200)
```

```
In [53]: x = 1e-500

In [54]: x
Out[54]: 0.0

In [55]: x / x
---------------------------------------------------------------------------
ZeroDivisionError                         Traceback (most recent call
        ↪   last)
<ipython-input-55-fd52a7f8b5f1> in <module>()
----> 1 x / x

ZeroDivisionError: float division by zero
```

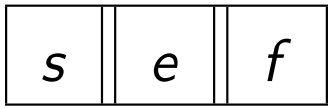
```
In [56]: sqrt(-1.0)
---------------------------------------------------------------------------
ValueError                                Traceback (most recent call
        ↪  last)
<ipython-input-56-d1c09f21b443> in <module>()
----> 1 sqrt(-1.0)

ValueError: math domain error
```

UNIVERSITÉ DE NANTES

```
*Main> let x = 1e500
*Main> x - x
NaN
*Main> x / x
NaN
*Main> sqrt(-1)
NaN
*Main> 0 / 0
NaN
```

UNIVERSITÉ DE NANTES

Arbre de lecture



UNIVERSITÉ DE NANTES

Arbre de lecture

s	e	f

$e = 0...0$	$f = 0$	$\rightarrow$	$v = (-1)^s \times 0.0$
$e = 0...0$	$f \neq 0$	$\rightarrow$	$v = (-1)^s \times 0.f \times 2^{1-E_{\max}}$
$e = 1...1$	$f = 0$	$\rightarrow$	$v = (-1)^s \times \infty$
$e = 1...1$	$f \neq 0$	$\rightarrow$	NaN
$0...0 < e < 1...1$	$f \neq 0$	$\rightarrow$	$v = (-1)^s \times 1.f \times 2^{e-E_{\max}}$

UNIVERSITÉ DE NANTES

Successeur

$$n = \#f$$

Successeur

$$n = \#f$$

toy7,

Successeur

$$n = \#f$$

$toy7,\ n = 3$

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32,

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32, $n = 23$

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32, $n = 23$
binary64,

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32, $n = 23$
binary64, $n = 52$

UNIVERSITÉ DE NANTES

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32, $n = 23$
binary64, $n = 52$

$$v = M(v) \times 2^{E(v)-n}$$

UNIVERSITÉ DE NANTES

Successeur

$$n = \#f$$

toy7, $n = 3$
binary32, $n = 23$
binary64, $n = 52$

$$v = M(v) \times 2^{E(v)-n}$$

$$\text{succ}(v) = (M(v) + 1) \times 2^{E(v)-n} = v + 2^{E(v)-n}$$

UNIVERSITÉ DE NANTES

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1
- On notera λ le plus petit VF normal positif

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1
- On notera λ le plus petit VF normal positif
- On notera μ le plus petit VF sous-normal positif

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1
- On notera λ le plus petit VF normal positif
- On notera μ le plus petit VF sous-normal positif
- On notera Ω le plus grand VF normal.

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1
- On notera λ le plus petit VF normal positif
- On notera μ le plus petit VF sous-normal positif
- On notera Ω le plus grand VF normal.

Tableau récapitulatif

- On notera ε_m l'*epsilon* de la machine, c'est-à-dire le successeur de 1
- On notera λ le plus petit VF normal positif
- On notera μ le plus petit VF sous-normal positif
- On notera Ω le plus grand VF normal.

Quelle relation existe-t-il entre μ, ε_m et λ ?

UNIVERSITÉ DE NANTES

Nom	E	f	$E_{\min}$	$E_{\max}$	ε_m	λ	μ	Ω
toy7	3	3	-2	3	$2^{-3} = 1/8$	$2^{-2} = 1/4$	$2^{-5} = 1/32$	$1,111 \times 2^3 = 15$
bin32	8	23	-126	127	2^{-23} $\approx 1,2 \times 10^{-7}$	2^{-126} $\approx 1,2 \times 10^{-38}$	$2^{-126-23}$ $\approx 1,4 \times 10^{-45}$	$(2^{24}-1) \times 2^{12}$ $\approx 3,4 \times 10^{38}$
bin64	11	52	-1022	1023	2^{-52} $\approx 2,2 \times 10^{-16}$	2^{-1022} $\approx 2,2 \times 10^{-308}$	2^{-1074} $\approx 5 \times 10^{-324}$	$(2^{53}-1)2^{1023-}$ $\approx 1,8 \times 10^{308}$

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
.
.
.
.
.
.
.
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True
.
.
.
.
.
.
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16
.
.
.
.
.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
.
.
.
.
.
.
.
```

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
Out[3]: 0.0

.
.
.
.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
Out[3]: 0.0

In [4]: x - 1e-16
.

.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
Out[3]: 0.0

In [4]: x - 1e-16
Out[4]: 0.9999999999999999

.
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
Out[3]: 0.0

In [4]: x - 1e-16
Out[4]: 0.9999999999999999

In [5]: 1e-16 + 1e-18
.
```

UNIVERSITÉ DE NANTES

```
In [1]: 1 + 1e-16 == 1
Out[1]: True

In [2]: x = 1 + 1e-16

In [3]: x - 1
Out[3]: 0.0

In [4]: x - 1e-16
Out[4]: 0.9999999999999999

In [5]: 1e-16 + 1e-18
Out[5]: 1.01e-16
```

UNIVERSITÉ DE NANTES

Sommaire

UNIVERSITÉ DE NANTES

$$\mathbb{V}_b$$

$$\mathbb{V}_b$$

$$\overline{\mathbb{V}_b}$$

Comparaison

Il est très simple et rapide de comparer deux VF : comment la machine procède-t-elle ? Quel est l'avantage de ce stockage des VF ?

Addition

1. on commence par ramener les deux nombres au même exposant, en l'occurrence le plus grand des deux ;

Addition

1. on commence par ramener les deux nombres au même exposant, en l'occurrence le plus grand des deux ;
2. on ajoute les deux mantisses *complètes* en tenant compte du signe ;

Addition

1. on commence par ramener les deux nombres au même exposant, en l'occurrence le plus grand des deux ;
2. on ajoute les deux mantisses *complètes* en tenant compte du signe ;
3. on renormalise le nombre obtenu.

Addition

En *toy7* : $1,1 + 0,0111$

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^{0} + 1,11 \times 2^{-2}$

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^{0} + 1,11 \times 2^{-2}$

1,1 :

0	0	1	1	1	0	0	

UNIVERSITÉ DE NANTES

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^{0} + 1,11 \times 2^{-2}$

1,1 :

0	0	1	1	1	0	0	

0,0011 :

0	0	0	1	1	1	0	

UNIVERSITÉ DE NANTES

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^{0} + 1,11 \times 2^{-2}$

1,1 :

0	0	1	1	1	0	0

0,0011 :

0	0	0	1	1	1	0

1,100

UNIVERSITÉ DE NANTES

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^{0} + 1,11 \times 2^{-2}$

1,1 :

0	0	1	1	1	0	0

0,0011 :

0	0	0	1	1	1	0

1,100
0,001*11*
– – ––

Addition

En *toy7* : $1,1 + 0,0111 \quad \rightarrow \quad 1,1 \times 2^0 + 1,11 \times 2^{-2}$

1,1 :

0	0	1	1	1	0	0

0,0011 :

0	0	0	1	1	1	0

1,100
0,001*11*
– – ––
1,101*11*

UNIVERSITÉ DE NANTES

Les arrondis

Définition 1

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;
2. l'arrondi vers 0 (RZ) qui arrondit à la valeur de plus petite valeur absolue : c'est la troncature ;

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;
2. l'arrondi vers 0 (RZ) qui arrondit à la valeur de plus petite valeur absolue : c'est la troncature ;
3. l'arrondi vers $+\infty$ (RU) qui arrondit à la valeur supérieure la plus petite ;

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;
2. l'arrondi vers 0 (RZ) qui arrondit à la valeur de plus petite valeur absolue : c'est la troncature ;
3. l'arrondi vers $+\infty$ (RU) qui arrondit à la valeur supérieure la plus petite ;
4. l'arrondi vers $-\infty$ (RD) qui arrondit à la valeur inférieure la plus grande.

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;
2. l'arrondi vers 0 (RZ) qui arrondit à la valeur de plus petite valeur absolue : c'est la troncature ;
3. l'arrondi vers $+\infty$ (RU) qui arrondit à la valeur supérieure la plus petite ;
4. l'arrondi vers $-\infty$ (RD) qui arrondit à la valeur inférieure la plus grande.

Les arrondis

Définition 1

1. l'arrondi au plus proche (RN) qui arrondit au VF...le plus proche. En cas d'égalité, on choisit la valeur paire (donc qui se termine par un 0 en binaire) ;
2. l'arrondi vers 0 (RZ) qui arrondit à la valeur de plus petite valeur absolue : c'est la troncature ;
3. l'arrondi vers $+\infty$ (RU) qui arrondit à la valeur supérieure la plus petite ;
4. l'arrondi vers $-\infty$ (RD) qui arrondit à la valeur inférieure la plus grande.

Le mode d'arrondi par défaut est le premier.

1,100
0,001*11*
– – ––
1,101*11*

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

$$1,10100 \leqslant \underbrace{1,10111}_{x} \leqslant 1,11000$$

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

$$1,10100 \leqslant \underbrace{1,10111}_{x} \leqslant 1,11000$$

$$\underbrace{1,100 + 1,00\,ulp(x)}_{v} \leqslant \underbrace{1,100 + 1,11\,ulp(x)}_{x} \leqslant \underbrace{1,100 + 10,00\,ulp(x)}_{\mathsf{succ}(v)}$$

$|x - v| = 0,11\,ulp(x)$

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

$$1,10100 \leqslant \underbrace{1,10111}_{x} \leqslant 1,11000$$

$$\underbrace{1,100 + 1,00ulp(x)}_{v} \leqslant \underbrace{1,100 + 1,11ulp(x)}_{x} \leqslant \underbrace{1,100 + 10,00ulp(x)}_{\mathrm{succ}(v)}$$

$|x - v| = 0,11ulp(x)$ $|x - \mathrm{succ}(v)| = 0,01ulp(x)$

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

$$1,10100 \leqslant \underbrace{1,10111}_{x} \leqslant 1,11000$$

$$\underbrace{1,100 + 1,00ulp(x)}_{v} \leqslant \underbrace{1,100 + 1,11ulp(x)}_{x} \leqslant \underbrace{1,100 + 10,00ulp(x)}_{\mathrm{succ}(v)}$$

$|x - v| = 0,11ulp(x)$ $|x - \mathrm{succ}(v)| = 0,01ulp(x)$ donc
$RN(x) = \mathrm{succ}(v)$

1,100
0,001*11*
– – ––
1,101*11*

$x = 1,10111$ VF en *toy7* ?

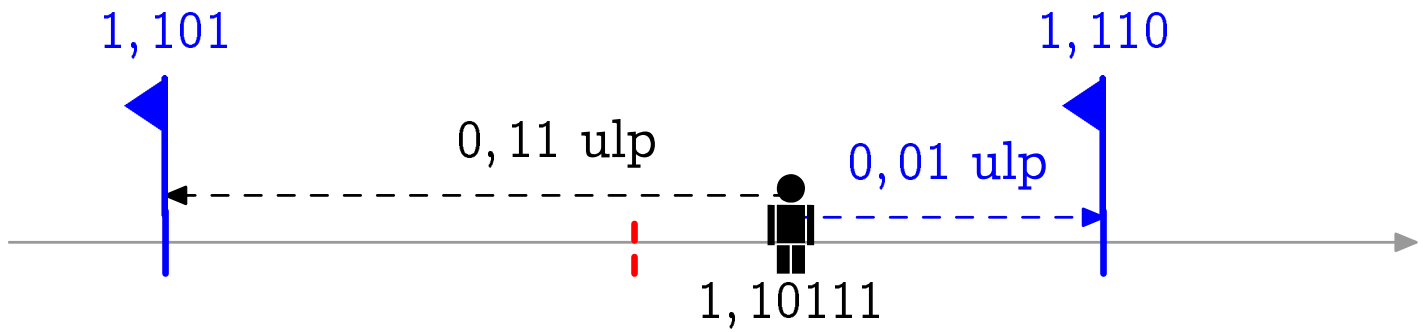
$$1,10100 \leqslant \underbrace{1,10111}_{x} \leqslant 1,11000$$

$$\underbrace{1,100 + 1,00ulp(x)}_{v} \leqslant \underbrace{1,100 + 1,11ulp(x)}_{x} \leqslant \underbrace{1,100 + 10,00ulp(x)}_{\mathrm{succ}(v)}$$

$|x - v| = 0,11ulp(x)$ $|x - \mathrm{succ}(v)| = 0,01ulp(x)$ donc
$RN(x) = \mathrm{succ}(v)$

$$1,1 \oplus 0,00111 = 1,110$$

UNIVERSITÉ DE NANTES


1, 101
1, 110
0, 11 ulp
0, 01 ulp
1, 10111

1. Est-ce que l'addition des VF est associative ?

1. Est-ce que l'addition des VF est associative ?

```
In [14]: (1 + 1e-16) + 1e-16
Out[14]: 1.0

In [15]: 1 + (1e-16 + 1e-16)
Out[15]: 1.0000000000000002
```

UNIVERSITÉ DE NANTES

1. Est-ce que l'addition des VF est associative ?

```
In [14]: (1 + 1e-16) + 1e-16
Out[14]: 1.0

In [15]: 1 + (1e-16 + 1e-16)
Out[15]: 1.0000000000000002
```

2. Est-on sûr de l'ordre dans le quel un compilateur calcule $a + b + c + d$?

UNIVERSITÉ DE NANTES

Multiplication

1. on « xore » les bits de signe ;

Multiplication

1. on « xore » les bits de signe ;
2. on additionne les exposants réels et on décale ou plutôt on additionne les exposants décalés et on retire la valeur d'un décalage ;

Multiplication

1. on « xore » les bits de signe ;
2. on additionne les exposants réels et on décale ou plutôt on additionne les exposants décalés et on retire la valeur d'un décalage ;
3. on multiplie les mantisses ;

Multiplication

1. on « xore » les bits de signe ;
2. on additionne les exposants réels et on décale ou plutôt on additionne les exposants décalés et on retire la valeur d'un décalage ;
3. on multiplie les mantisses ;
4. on normalise.

$$101,1 \times (-10,01)$$

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

UNIVERSITÉ DE NANTES

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. 101 + 100 − 11 = 1001 − 11 = 110 : l'exposant décalé est 110 donc l'exposant est 3 ;

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$$1,100 < x < 1,101$$

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$$1,100 < x < 1,101$$

avec $|x - 1,100| =$

UNIVERSITÉ DE NANTES

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$$1,100 < x < 1,101$$

avec $|x - 1,100| = 0,011 ulp(x)$ et $|x - 1,101| =$

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$$1,100 < x < 1,101$$

avec $|x - 1,100| = 0,011\,ulp(x)$ et $|x - 1,101| = 0,101\,ulp(x)$

$101,1 \times (-10,01)$

101,1 :	0	1	0	1	0	1	1
-10,01 :	1	1	0	0	0	0	1

1. 0 xor 1 donne 1 : le bit de signe est 1 ;
2. $101 + 100 - 11 = 1001 - 11 = 110$: l'exposant décalé est 110 donc l'exposant est 3 ;
3. $1,011 \times 1,001 = 1,011 + 1,011 \times 0,001 = 1,011 + 0,001011 = 1,100011$;
4. le produit est donc $1,100011 \times 2^3$. Le passage à la forme normale va arrondir le produit. On a

$$1,100 < x < 1,101$$

avec $|x - 1,100| = 0,011\,ulp(x)$ et $|x - 1,101| = 0,101\,ulp(x)$ donc

$$RN(101,1 \times (-10,01)) = -1,100 \times 2^3$$

1	1	1	0	1	0	0

UNIVERSITÉ DE NANTES

1. Est-ce que la multiplication des VF est associative ?

1. Est-ce que la multiplication des VF est associative ?

```
In [1]: (1.00000001 * 1e-15) * 1e15
Out[1]: 1.0000000100000002

In [2]: 1.00000001 * (1e-15 * 1e15)
Out[2]: 1.00000001
```

1. Est-ce que la multiplication des VF est associative ?

```
In [1]: (1.00000001 * 1e-15) * 1e15
Out[1]: 1.0000000100000002

In [2]: 1.00000001 * (1e-15 * 1e15)
Out[2]: 1.00000001
```

2. Est-ce que la multiplication des VF est distributive sur l'addition ?

UNIVERSITÉ DE NANTES

1. Est-ce que la multiplication des VF est associative ?

```
In [1]: (1.00000001 * 1e-15) * 1e15
Out[1]: 1.0000000100000002

In [2]: 1.00000001 * (1e-15 * 1e15)
Out[2]: 1.00000001
```

2. Est-ce que la multiplication des VF est distributive sur l'addition ?

```
In [3]: 1e15 * (1e-16 + 1)
Out[3]: 1000000000000000.0

In [4]: (1e15 * 1e-16) + (1e15 * 1)
Out[4]: 1000000000000000.1
```

UNIVERSITÉ DE NANTES

Sommaire

UNIVERSITÉ DE NANTES

```python
def base1(a):
    if (a + 1.0) - a != 1.0:
        return a
    else:
        return base1(2.0 * a)

def base2(a, b):
    if (a + b) - a == b:
        return b
    else:
        return base2(a, b + 1.0)
```

```python
def base1(a):
    if (a + 1.0) - a != 1.0:
        return a
    else:
        return base1(2.0 * a)

def base2(a, b):
    if (a + b) - a == b:
        return b
    else:
        return base2(a, b + 1.0)
```

? ? ? ? ! ! ! ? ? ? ? ! ! ? ? ?

95 % of the folks out there are completely clueless about floating-point

James Gosling *(M. Java) - 28 février 1998*

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre.

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre.

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre. Par exemple `3`, `3.0`, `3.0000`, `3.0e0` n'ont pas la même précision, précision qui dépend en plus du langage.

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre. Par exemple `3`, `3.0`, `3.0000`, `3.0e0` n'ont pas la même précision, précision qui dépend en plus du langage.

Exactitude (*accuracy*) : c'est ce qui relie un nombre au contexte dans lequel il est employé.

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre. Par exemple `3`, `3.0`, `3.0000`, `3.0e0` n'ont pas la même précision, précision qui dépend en plus du langage.

Exactitude (*accuracy*) : c'est ce qui relie un nombre au contexte dans lequel il est employé.

Précision (*precision*) : c'est le nombre de bits utilisés pour représenter un nombre. La précision concerne donc le format utilisé pour écrire ou stocker ou arrondir un nombre. Par exemple `3`, `3.0`, `3.0000`, `3.0e0` n'ont pas la même précision, précision qui dépend en plus du langage.

Exactitude (*accuracy*) : c'est ce qui relie un nombre au contexte dans lequel il est employé.

$3,177777777777777$ est une approximation plutôt précise (16 décimales) mais inexacte (2 décimales) de π.

UNIVERSITÉ DE NANTES

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

l'erreur absolue $|x - \widehat{x}|$

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

l'erreur absolue $|x - \widehat{x}|$

l'erreur relative $\eta = \frac{x-\widehat{x}}{x}$

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

l'erreur absolue $|x - \widehat{x}|$

l'erreur relative $\eta = \frac{x-\widehat{x}}{x}$

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

l'erreur absolue $|x - \widehat{x}|$

l'erreur relative $\eta = \frac{x-\widehat{x}}{x}$ alors $\widehat{x} = x(1+\eta)$

Soit x un nombre et $\widehat{x}$ le nombre qui le représente. On distingue :

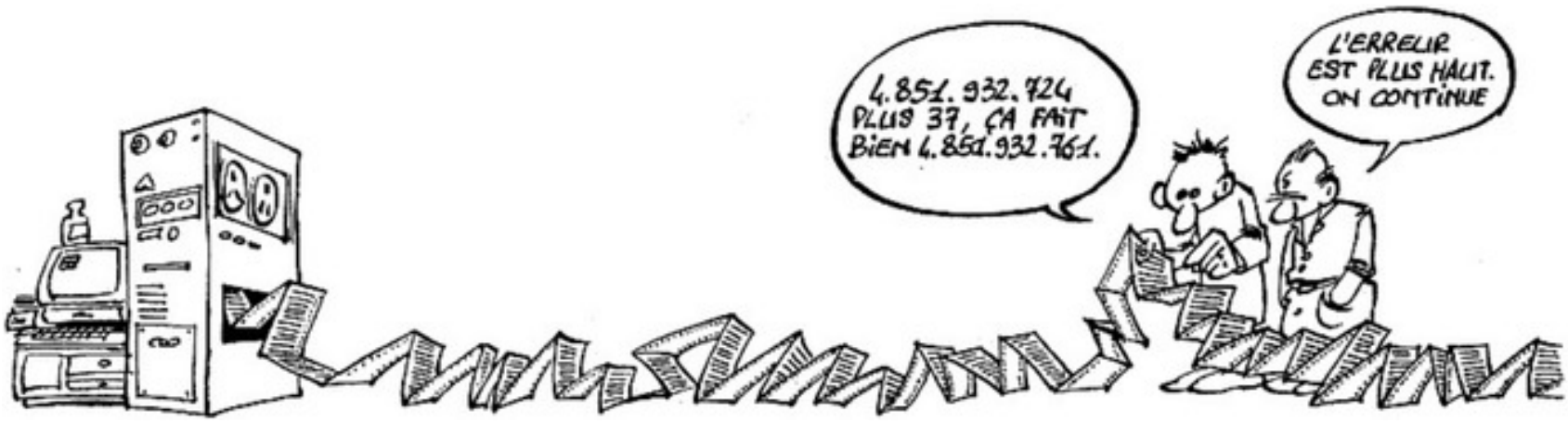
l'erreur absolue $|x - \widehat{x}|$

l'erreur relative $\eta = \frac{x-\widehat{x}}{x}$ alors $\widehat{x} = x(1+\eta)$

commise en prenant $\widehat{x}$ à la place de x.

Feel nervous, but feel in control. It's not dark magic, it's science.

Florent DE DINECHIN



UNIVERSITÉ DE NANTES

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left| \frac{x - \widehat{x}}{x} \right| \leqslant \left| \frac{x - \widehat{x}}{m \times 2^E} \right|$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left| \frac{x - \widehat{x}}{x} \right| \leqslant \left| \frac{x - \widehat{x}}{m \times 2^E} \right| \leqslant \frac{1}{2} \frac{2^{E-n}}{m \times 2^E}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2} \frac{2^{E-n}}{m \times 2^E} = \frac{1}{2} \frac{2^{-n}}{m}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2}\frac{2^{E-n}}{m \times 2^E} = \frac{1}{2}\frac{2^{-n}}{m} = \frac{1}{2}\frac{\varepsilon_M}{m}$$

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2}\frac{2^{E-n}}{m \times 2^E} = \frac{1}{2}\frac{2^{-n}}{m} = \frac{1}{2}\frac{\varepsilon_M}{m}$$

1. Si $\widehat{x}$ VF alors l'**erreur relative** est majorée $\left|\frac{x-\widehat{x}}{x}\right| \leqslant \frac{1}{2}\varepsilon_M$

UNIVERSITÉ DE NANTES

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2}\frac{2^{E-n}}{m \times 2^E} = \frac{1}{2}\frac{2^{-n}}{m} = \frac{1}{2}\frac{\varepsilon_M}{m}$$

1. Si $\widehat{x}$ VF alors l'**erreur relative** est majorée $\left|\frac{x-\widehat{x}}{x}\right| \leqslant \frac{1}{2}\varepsilon_M$
2. Si $\widehat{x}$ est sous-normal, alors l'**erreur absolue** est majorée

UNIVERSITÉ DE NANTES

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2}\frac{2^{E-n}}{m \times 2^E} = \frac{1}{2}\frac{2^{-n}}{m} = \frac{1}{2}\frac{\varepsilon_M}{m}$$

1. Si $\widehat{x}$ VF alors l'**erreur relative** est majorée $\left|\frac{x-\widehat{x}}{x}\right| \leqslant \frac{1}{2}\varepsilon_M$
2. Si $\widehat{x}$ est sous-normal, alors l'**erreur absolue** est majorée

UNIVERSITÉ DE NANTES

Précision machine

$$M \times 2^{E-n} \leqslant x < (M+1) \times 2^{E-n} = M \times 2^{E-n} + 2^{E-n}$$

$$|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E-n}$$

$$\left|\frac{x - \widehat{x}}{x}\right| \leqslant \left|\frac{x - \widehat{x}}{m \times 2^E}\right| \leqslant \frac{1}{2}\frac{2^{E-n}}{m \times 2^E} = \frac{1}{2}\frac{2^{-n}}{m} = \frac{1}{2}\frac{\varepsilon_M}{m}$$

1. Si $\widehat{x}$ VF alors l'**erreur relative** est majorée $\left|\frac{x-\widehat{x}}{x}\right| \leqslant \frac{1}{2}\varepsilon_M$
2. Si $\widehat{x}$ est sous-normal, alors l'**erreur absolue** est majorée $|x - \widehat{x}| \leqslant \frac{1}{2} 2^{E_{\mathbf{min}}-1-n}$

UNIVERSITÉ DE NANTES

NE FAITES PAS DES TESTS D'ÉGALITÉ MAIS DES TESTS D'APPARTENANCE À DES INTERVALLES DE LARGEUR L'ε MACHINE !

Précision machine

$$\widehat{x} = x(1 + \eta_1) + \eta_2$$

avec :

- si $\widehat{x}$ est normal $|\eta_1| \leqslant \frac{1}{2}\varepsilon_M$ et $\eta_2 = 0$

Précision machine

$$\widehat{x} = x(1 + \eta_1) + \eta_2$$

avec :

- si $\widehat{x}$ est normal $|\eta_1| \leqslant \frac{1}{2}\varepsilon_M$ et $\eta_2 = 0$
- si $\widehat{x}$ est sous-normal $\eta_1 = 0$ et $|\eta_2| \leqslant \frac{1}{2}2^{E_{\mathbf{min}}-1-n}$

Précision machine

$$\widehat{x} = x(1 + \eta_1) + \eta_2$$

avec :

- si $\widehat{x}$ est normal $|\eta_1| \leqslant \frac{1}{2}\varepsilon_M$ et $\eta_2 = 0$
- si $\widehat{x}$ est sous-normal $\eta_1 = 0$ et $|\eta_2| \leqslant \frac{1}{2}2^{E_{\mathbf{min}}-1-n}$

Précision machine

$$\widehat{x} = x(1 + \eta_1) + \eta_2$$

avec :

- si $\widehat{x}$ est normal $|\eta_1| \leqslant \frac{1}{2}\varepsilon_M$ et $\eta_2 = 0$
- si $\widehat{x}$ est sous-normal $\eta_1 = 0$ et $|\eta_2| \leqslant \frac{1}{2}2^{E_{\mathbf{min}}-1-n}$

Mouais...

UNIVERSITÉ DE NANTES

Élimination

$$\widehat{a} = a(1 + \eta_a),$$

Élimination

$$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b),$$

Élimination

$$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b), x = a - b$$

Élimination

$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b), x = a - b$ et $\widehat{x} = \widehat{a - b}$

Élimination

$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b), x = a - b$ et $\widehat{x} = \widehat{\widehat{a} - \widehat{b}} = \widehat{a} - \widehat{b}$.

Élimination

$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b), x = a - b$ et $\widehat{x} = \widehat{\widehat{a} - \widehat{b}} = \widehat{a} - \widehat{b}$.

$$\left|\frac{x - \widehat{x}}{x}\right| = \left|\frac{-a\eta_a + b\eta_b}{a - b}\right|$$

Élimination

$\widehat{a} = a(1+\eta_a), \widehat{b} = b(1+\eta_b), x = a - b$ et $\widehat{x} = \widehat{\widehat{a} - \widehat{b}} = \widehat{a} - \widehat{b}$.

$$\left|\frac{x - \widehat{x}}{x}\right| = \left|\frac{-a\eta_a + b\eta_b}{a-b}\right| \leqslant \max(|\eta_a|, |\eta_b|)\frac{|a| + |b|}{|a-b|}$$

Catastrophe

```
def deriv(f,h,x):
    return (f(x+h) - f(x)) / h
```

Catastrophe

```
def deriv(f,h,x):
    return (f(x+h) - f(x)) / h
```

```
def derivn(f,x,h,n):
    if n == 0:
        return f(x)
    else:
        return derivn(lambda x: deriv(f,h,x),x,h,n-1)
```

Catastrophe

```
In [1]: [derivn(lambda x: x**4,1,1e-7,k) for k in range(5)]
Out[1]:
[1,
 4.000000601855902,
 12.012613126444194,
 -222044.6049250313,
 2220446049250.313]
```

Catastrophe

```
In [1]: [derivn(lambda x: x**4,1,1e-7,k) for k in range(5)]
Out[1]:
[1,
 4.000000601855902,
 12.012613126444194,
 -222044.6049250313,
 2220446049250.313]
```

```
In [2]: [derivn(lambda x: x**4,1,1e-8,k) for k in range(5)]
Out[2]: [1, 4.000000042303498, 11.102230246251565, 0.0,
         ↪   2.220446049250313e+16]
```

Catastrophe

```
In [3]: [derivn(lambda x: x**4,1,1e-9,k) for k in range(5)]
Out[3]: [1, 4.000000330961484, 0.0, 0.0, 0.0]
```

Catastrophe

```
In [3]: [derivn(lambda x: x**4,1,1e-9,k) for k in range(5)]
Out[3]: [1, 4.000000330961484, 0.0, 0.0, 0.0]
```

```
In [4]: [derivn(lambda x: x**4,1,1e-3,k) for k in range(6)]
Out[4]:
[1,
 4.006004000999486,
 12.024014000244776,
 24.03599941303014,
 24.001245435556484,
 -2.4424906541753444]
```

UNIVERSITÉ DE NANTES

Catastrophe

```
In [5]: [derivn(lambda x: x**4,1,2**-10,k) for k in range(6)]
Out[5]: [1, 4.005863190628588, 12.02345085144043, 24.03515625, 24.0,
         ↪  0.0]
```

Catastrophe

```
In [5]: [derivn(lambda x: x**4,1,2**-10,k) for k in range(6)]
Out[5]: [1, 4.005863190628588, 12.02345085144043, 24.03515625, 24.0,
         ↪   0.0]
```

```
In [6]: [derivn(lambda x: x**4,1,1.0/1026.,k) for k in range(6)]
Out[6]:
[1,
 4.005851753982072,
 12.023405112200917,
 24.035087928668187,
 23.999573957547014,
 0.755876563500351]
```

Catastrophe

```
In [7]: [derivn(lambda x: x**4,1,2**-20,k) for k in range(6)]
Out[7]: [1, 4.000005722045898, 12.0, 0.0, 268435456.0,
         ↪   -844424930131968.0]
```

Catastrophe

```
In [7]: [derivn(exp,0,2**-5,k) for k in range(10)]
Out[7]:
[1.0,
 1.0157890399712883,
 1.0318273737254913,
 1.0481189373895177,
 1.0646677284967154,
 1.081477828323841,
 1.0985534191131592,
 1.1158447265625,
 1.1376953125,
 0.953125]
```

Catastrophe

```
In [8]: [derivn(exp,0,2**-20,k) for k in range(10)]
Out[8]: [1.0, 1.0000004768371582, 1.0, 0.0, 0.0, 0.0, 0.0, 0.0, 0.0,
         ↪   0.0]
```

TROP DE PRÉCISION TUE LA PRÉCISION !

TROP DE PRÉCISION TUE LA PRÉCISION ! DIVISEZ PAR DES PUISSANCES DE 2...

TROP DE PRÉCISION TUE LA PRÉCISION !
DIVISEZ PAR DES PUISSANCES DE 2...PAS DES PUISSANCES DE 10 !

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ *avec* $a \neq 0$*...*
Les racines, si elles existent, sont données par une formule bien connue dépendant de a*,* b *et* c *:*

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ *avec* $a \neq 0$*...*
Les racines, si elles existent, sont données par une formule bien connue dépendant de a*,* b *et* c *:*

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

1. *Que se passe-t-il lorsque* $b^2 \gg |4ac|$ *? En quoi la formule* $\frac{-\left(b+\text{signe}(b)\sqrt{\Delta}\right)}{2a}$ *peut aider ?*

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ avec $a \neq 0$...
Les racines, si elles existent, sont données par une formule bien connue dépendant de a, b et c :

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

1. *Que se passe-t-il lorsque $b^2 \gg |4ac|$? En quoi la formule $\frac{-\left(b+\text{signe}(b)\sqrt{\Delta}\right)}{2a}$ peut aider ?*
2. *Que se passe-t-il lorsque $b^2 \approx 4ac$? Peut-on y remédier ? Que peut-on dire de Δ par rapport à b^2 ? Y a-t-il élimination catastrophique ?*

UNIVERSITÉ DE NANTES

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ *avec* $a \neq 0$*...*
Les racines, si elles existent, sont données par une formule bien connue dépendant de a*,* b *et* c *:*

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

1. *Que se passe-t-il lorsque* $b^2 \gg |4ac|$ *? En quoi la formule* $\frac{-\left(b+\text{signe}(b)\sqrt{\Delta}\right)}{2a}$ *peut aider ?*
2. *Que se passe-t-il lorsque* $b^2 \approx 4ac$ *? Peut-on y remédier ? Que peut-on dire de* Δ *par rapport à* b^2 *? Y a-t-il élimination catastrophique ?*
3. *Que se passe-t-il dans le cas de l'équation* $10^{200}x^2 - 3 \times 10^{200}x + 2 \times 10^{200} = 0$ *?*

UNIVERSITÉ DE NANTES

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ *avec* $a \neq 0$*...*
Les racines, si elles existent, sont données par une formule bien connue dépendant de a*,* b *et* c *:*

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

1. *Que se passe-t-il lorsque* $b^2 \gg |4ac|$ *? En quoi la formule* $\frac{-\left(b+\text{signe}(b)\sqrt{\Delta}\right)}{2a}$ *peut aider ?*
2. *Que se passe-t-il lorsque* $b^2 \approx 4ac$ *? Peut-on y remédier ? Que peut-on dire de* Δ *par rapport à* b^2 *? Y a-t-il élimination catastrophique ?*
3. *Que se passe-t-il dans le cas de l'équation* $10^{200}x^2 - 3 \times 10^{200}x + 2 \times 10^{200} = 0$ *?*

UNIVERSITÉ DE NANTES

Exercice 3

Vous savez résoudre une équation du type $ax^2 + bx + c = 0$ *avec* $a \neq 0$*...*
Les racines, si elles existent, sont données par une formule bien connue dépendant de a*,* b *et* c *:*

$$r = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a} = \frac{-b \pm \sqrt{\Delta}}{2a} \quad \text{avec} \quad \Delta = b^2 - 4ac$$

Où peuvent se cacher d'éventuelles annulations catastrophiques ? Étudiez ces cas avec attention, voyez si vous pouvez éviter les éliminations catastrophiques en réécrivant les formules un peu dans l'esprit de la « levée d'indétermination ».

1. *Que se passe-t-il lorsque* $b^2 \gg |4ac|$ *? En quoi la formule* $\frac{-\left(b+\text{signe}(b)\sqrt{\Delta}\right)}{2a}$ *peut aider ?*
2. *Que se passe-t-il lorsque* $b^2 \approx 4ac$ *? Peut-on y remédier ? Que peut-on dire de* Δ *par rapport à* b^2 *? Y a-t-il élimination catastrophique ?*
3. *Que se passe-t-il dans le cas de l'équation* $10^{200}x^2 - 3 \times 10^{200}x + 2 \times 10^{200} = 0$ *?*

UNIVERSITÉ DE NANTES

Élimination

```
def expn(n):
    return (1. + 1./n)**n
```

Élimination

```
In [1]: [(exp(1) - expn(10.0**k))
        ↪   for k in range(20)]
Out[1]:
[0.7182818284590451,
 0.12453936835904278,
 0.01346799903751661,
 0.0013578962234515046,
 0.0001359016341195 84,
 1.359126674760347e-05,
 1.359363291708604e-06,
 1.3432696333026684e-07,
 3.011168736577474e-08,
```

```
-2.2355251516614771e-07,
-2.2477574246337895e-07,
-2.248980650598753e-07,
-0.00024166757819266138,
0.002171794372144209,
0.0021717943720220845,
-0.31675337809021675,
1.718281828459045,
1.718281828459045,
1.718281828459045,
1.718281828459045]
```

Élimination

```
In [2]: [(exp(1) -
         ↪   expn(2.0**(3*k))) for
         ↪   k in range(20)]
Out[2]:
[0.7182818284590451,
 0.1524973145086972,
 0.020936875893946105,
 0.0026498282900537795,
 0.0003317472693793455,
 4.147652875108321e-05,
 5.1846929989274315e-06,
 6.480886076687398e-07,
 8.101110671177025e-08,
```

```
1.0126388616527038e-08,
1.2657985770658797e-09,
1.582245445774788e-10,
1.977795704988239e-11,
2.4722446312352986e-12,
3.090860900556436e-13,
3.863576125695545e-14,
4.884981308350689e-15,
4.440892098500626e-16,
1.718281828459045,
1.718281828459045]
```

```
In [89]: [2.0**53 + k for k in range(10)]
Out[89]:
[9007199254740992.0,
 9007199254740992.0,
 9007199254740994.0,
 9007199254740996.0,
 9007199254740996.0,
 9007199254740996.0,
 9007199254740998.0,
 9007199254741000.0,
 9007199254741000.0,
 9007199254741000.0]
```

```
In [91]: [2.0**55 + k for k in range(10)]
Out[91]:
[3.602879701896397e+16,
 3.602879701896397e+16,
 3.602879701896397e+16,
 3.602879701896397e+16,
 3.602879701896397e+16,
 3.6028797018963976e+16,
 3.6028797018963976e+16,
 3.6028797018963976e+16,
 3.6028797018963976e+16,
 3.6028797018963976e+16]
```

UNIVERSITÉ DE NANTES

Attention !

Pour éviter de déduire des lemmes suivants des théorèmes totalement faux, n'oubliez pas que DANS CE QUI SUIT X ET Y SONT DES NOMBRES À VIRGULE FLOTTANTE !

Lemme 2 (Majoration de l'erreur d'une somme)

Posons $x \oplus y = x + y + \mathsf{err}(x \oplus y)$. *Alors, s'il n'y a pas de dépassement de capacité,*

$$|\mathsf{err}(x \oplus y)| \leqslant \min(|x|, |y|)$$

On a bien sûr un résultat analogue pour la différence

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$

- $x \oplus y = x + y + \mathsf{err}(x \oplus y)$
- $x = x + y + (-y)$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$
- de même $|\mathrm{err}(x \oplus y)| \leqslant |x|$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$
- de même $|\mathrm{err}(x \oplus y)| \leqslant |x|$

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$
- de même $|\mathrm{err}(x \oplus y)| \leqslant |x|$

Faites un dessin...

- $x \oplus y = x + y + \mathrm{err}(x \oplus y)$
- $x = x + y + (-y)$
- x est un VF situé à la distance $|y|$ de $x + y$
- $x \oplus y$ est le VF le plus proche de $x + y$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$
- de même $|\mathrm{err}(x \oplus y)| \leqslant |x|$

Faites un dessin...

À noter

De l'importance de disposer avec la IEEE 754 de la **meilleure approximation** !

UNIVERSITÉ DE NANTES

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathsf{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathsf{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathsf{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

  x $\boxed{x_1}$ $\boxed{x_2}$

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

x $\boxed{x_1}$ $\boxed{x_2}$

$+y$ $\boxed{y_1}$ $\boxed{y_2}$

UNIVERSITÉ DE NANTES

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

$$
\begin{array}{lcccc}
x & \boxed{x_1} & \boxed{x_2} & & \\
+y & & & \boxed{y_1} & \boxed{y_2} \\
= & \boxed{x_1} & \boxed{x_2 + y_1} & \mathrm{err}(x \oplus y) &
\end{array}
$$

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

  x $\boxed{x_1}$ $\boxed{x_2}$

  $+y$ $\boxed{y_1}$ $\boxed{y_2}$

  $=$ $\boxed{x_1}$ $\boxed{x_2 + y_1}$ $\mathrm{err}(x \oplus y)$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$

UNIVERSITÉ DE NANTES

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

  x $\boxed{x_1}$ $\boxed{x_2}$

  $+y$ $\boxed{y_1}$ $\boxed{y_2}$

  $=$ $\boxed{x_1}$ $\boxed{x_2 + y_1}$ $\mathrm{err}(x \oplus y)$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$

UNIVERSITÉ DE NANTES

Corollaire 3 (Møller, Knuth, Dekker)

L'erreur commise $|\mathrm{err}(x \oplus y)|$ *peut être exprimée exactement sur* $p = ulp(y)$ *bits.*

- $|x| \geqslant |y|$
- x et y sont des VF : le plus petit bit significatif de $\mathrm{err}(x \oplus y)$ est au moins de magnitude celle de $\mathrm{ulp}(y)$

  x $\boxed{x_1}$ $\boxed{x_2}$

  $+y$ $\boxed{y_1}$ $\boxed{y_2}$

  $=$ $\boxed{x_1}$ $\boxed{x_2 + y_1}$ $\mathrm{err}(x \oplus y)$
- $|\mathrm{err}(x \oplus y)| \leqslant |y|$donc la mantisse entière de $\mathrm{err}(x \oplus y)$ a une longueur inférieure à p bits

UNIVERSITÉ DE NANTES

Lemme 4

Supposons que $|x + y| \leqslant \min(|x|, |y|)$*, alors* $x \oplus y = x + y$*.*
On obtient un résultat analogue pour la soustraction.

Lemme 4

Supposons que $|x + y| \leqslant \min(|x|, |y|)$*, alors* $x \oplus y = x + y$*.*
On obtient un résultat analogue pour la soustraction.

1. Supposons, sans perdre de généralité, que $|x| \geqslant |y|$

Lemme 4

Supposons que $|x+y| \leqslant \min(|x|,|y|)$*, alors* $x \oplus y = x + y$*.*
On obtient un résultat analogue pour la soustraction.

1. Supposons, sans perdre de généralité, que $|x| \geqslant |y|$
2. Le plus petit bit significatif de $x+y$ est au moins de magnitude celle de $\mathsf{ulp}(y)$

Lemme 4

Supposons que $|x+y| \leqslant \min(|x|,|y|)$*, alors* $x \oplus y = x + y$*.*
On obtient un résultat analogue pour la soustraction.

1. Supposons, sans perdre de généralité, que $|x| \geqslant |y|$
2. Le plus petit bit significatif de $x+y$ est au moins de magnitude celle de $\mathrm{ulp}(y)$
3. $|x+y| \leqslant |y|$ donc la mantisse entière de $x+y$ a une longueur inférieure à p bits

DANGER

Nous avons démontré nos lemmes en considérant des VF x et y.
Est-ce que le résultat suivant contredit notre dernier lemme ?

```
In [86]: 1 - 0.9
Out[86]: 0.09999999999999998
```

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

1. $\boldsymbol{x < y}$: alors $x < y \leqslant 2x$ donc $0 < y - x \leqslant x \leqslant y$;

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

1. $\boldsymbol{x < y}$: alors $x < y \leqslant 2x$ donc $0 < y - x \leqslant x \leqslant y$;
2. $\boldsymbol{x \geqslant y}$: alors $\frac{x}{2} \leqslant y \leqslant x$ donc $-\frac{x}{2} \leqslant y - x \leqslant 0$ et par suite $0 \leqslant x - y \leqslant \frac{x}{2} \leqslant y \leqslant x$.

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

1. $\boldsymbol{x < y}$: alors $x < y \leqslant 2x$ donc $0 < y - x \leqslant x \leqslant y$;
2. $\boldsymbol{x \geqslant y}$: alors $\frac{x}{2} \leqslant y \leqslant x$ donc $-\frac{x}{2} \leqslant y - x \leqslant 0$ et par suite $0 \leqslant x - y \leqslant \frac{x}{2} \leqslant y \leqslant x$.

Lemme 5 (Lemme de Sterbenz (1973))

Soit $(x, y) \in \mathbb{V}^2$ *vérifiant* $\frac{x}{2} \leqslant y \leqslant 2x$ *alors*

$$x \ominus y = x - y$$

La différence de deux VF suffisamment proches est donc exacte.

1. $\boldsymbol{x < y}$: alors $x < y \leqslant 2x$ donc $0 < y - x \leqslant x \leqslant y$;
2. $\boldsymbol{x \geqslant y}$: alors $\frac{x}{2} \leqslant y \leqslant x$ donc $-\frac{x}{2} \leqslant y - x \leqslant 0$ et par suite $0 \leqslant x - y \leqslant \frac{x}{2} \leqslant y \leqslant x$.

- Concrètement, à quoi correspond cette condition $\frac{x}{2} \leqslant y \leqslant 2x$?

- Concrètement, à quoi correspond cette condition $\frac{x}{2} \leqslant y \leqslant 2x$?
- Demandez-vous ce que l'on peut dire de l'écart maximum entre les exposants de x et y.

```
In [9]: 2.**53
Out[9]: 9007199254740992.0

In [10]: 2.**53 + 1
Out[10]: 9007199254740992.0

In [11]: 2.**53 + 2
Out[11]: 9007199254740994.0

In [12]: 2.**53 + 3
Out[12]: 9007199254740996.0

In [13]: 2.**53 + 4
Out[13]: 9007199254740996.0

In [14]: 2.**53 + 5
Out[14]: 9007199254740996.0
```

UNIVERSITÉ DE NANTES

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

```
0 10000110100 0000000000000000000000000000000000000000...00000
```

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

0 10000110100 0000000000000000000000000000000000000000...00000

$$1 = (-1)^0 \times 1,000...000 \times 2^0$$

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

```
0 10000110100 0000000000000000000000000000000000000...00000
```

$$1 = (-1)^0 \times 1,000...000 \times 2^0$$

```
0 01111111111 0000000000000000000000000000000000000...00000
```

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

0 10000110100 0000000000000000000000000000000000000000...00000

$$1 = (-1)^0 \times 1,000...000 \times 2^0$$

0 01111111111 0000000000000000000000000000000000000000...00000

$$1 = 0,\underbrace{000...000}_{52 bits}\ 1 \times 2^{53}$$

$$2^{53} = (-1)^0 \times 1,000...000 \times 2^{53}$$

```
0 10000110100 0000000000000000000000000000000000000...00000
```

$$1 = (-1)^0 \times 1,000...000 \times 2^0$$

```
0 01111111111 0000000000000000000000000000000000000...00000
```

$$1 = 0,\underbrace{000...000}_{52 bits}\ 1 \times 2^{53}$$

```
1,0000000000000000000000000000000000000000000000000000|
0,0000000000000000000000000000000000000000000000000000|1
----------------------------------------------------------
1,0000000000000000000000000000000000000000000000000000|1
```

Unit in the Last Place

Unit in the Last Place

Mauvaise nouvelle : une addition entre deux nombres flottants peut donc, dans certains cas, créer une erreur.

Unit in the Last Place

Mauvaise nouvelle : une addition entre deux nombres flottants peut donc, dans certains cas, créer une erreur.

Bonne nouvelle : on peut récupérer cette erreur.

Théorème 6 (Fast2Sum - Dekker & Kahan)

On considère deux VF x et y tels que $|x| \geqslant |y|$ et l'algorithme suivant :

```
1  s  ← x ⊕ y
2  y_v ← s ⊖ x
3  d  ← y ⊖ y_v
4  Retourner (s, d)
```

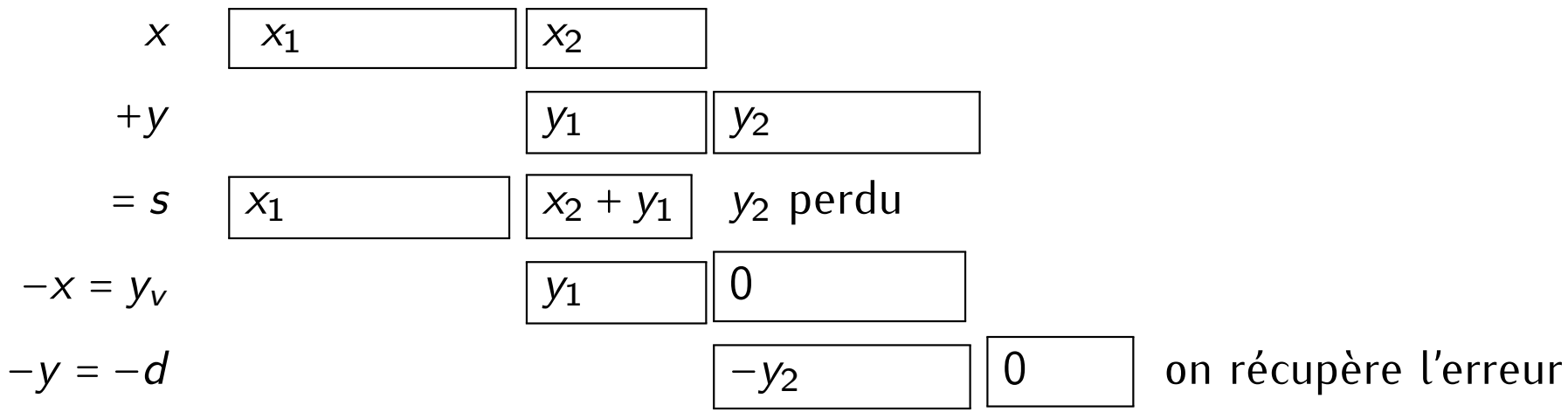
Alors $x + y = s + d$ avec $s = x \oplus y$ et $d = err(x \oplus y)$. De plus s et d ne se chevauchent pas.

UNIVERSITÉ DE NANTES

- **si x et y sont de même signe OU si $|y| \leqslant \frac{|x|}{2}$** : alors $\frac{x}{2} \leqslant s \leqslant 2x$ et on peut appliquer le lemme de Sterbenz ;

- **si x et y sont de même signe OU si $|y| \leqslant \frac{|x|}{2}$** : alors $\frac{x}{2} \leqslant s \leqslant 2x$ et on peut appliquer le lemme de Sterbenz ;
- **sinon** : on a x et y de signes opposés ET $y > \frac{|x|}{2}$ alors si y est négatif $x/2 < -y < x$ et sinon $-x/2 < y < -x$. Dans les deux cas, d'après le lemme de Sterbenz, s est calculée exactement et alors $y_v = y$.

Somme compensée



```
def fast2sum(x,y):
    if abs(x) >= abs(y):
        s  = x + y
        yv = s - x
        d  = y - yv
        return (s,d)
    else:
        return fast2sum(y,x)
```

```
In [100]: fast2sum(2.0**54,1.0)
Out[100]: (1.8014398509481984e+16, 1.0)
```

```
In [100]: fast2sum(2.0**54,1.0)
Out[100]: (1.8014398509481984e+16, 1.0)
```

```
In [101]: fast2sum(2.0**54,2.0)
Out[101]: (1.8014398509481984e+16, 2.0)
```

```
In [100]: fast2sum(2.0**54,1.0)
Out[100]: (1.8014398509481984e+16, 1.0)
```

```
In [101]: fast2sum(2.0**54,2.0)
Out[101]: (1.8014398509481984e+16, 2.0)
```

```
In [102]: fast2sum(2.0**54,3.0)
Out[102]: (1.8014398509481988e+16, -1.0)
```

```
In [100]: fast2sum(2.0**54,1.0)
Out[100]: (1.8014398509481984e+16, 1.0)
```

```
In [101]: fast2sum(2.0**54,2.0)
Out[101]: (1.8014398509481984e+16, 2.0)
```

```
In [102]: fast2sum(2.0**54,3.0)
Out[102]: (1.8014398509481988e+16, -1.0)
```

```
In [103]: fast2sum(2.0**54,4.0)
Out[103]: (1.8014398509481988e+16, 0.0)
```

```
In [100]: fast2sum(2.0**54,1.0)
Out[100]: (1.8014398509481984e+16, 1.0)
```

```
In [101]: fast2sum(2.0**54,2.0)
Out[101]: (1.8014398509481984e+16, 2.0)
```

```
In [102]: fast2sum(2.0**54,3.0)
Out[102]: (1.8014398509481988e+16, -1.0)
```

```
In [103]: fast2sum(2.0**54,4.0)
Out[103]: (1.8014398509481988e+16, 0.0)
```

```
In [104]: fast2sum(2.0**54,5.0)
Out[104]: (1.8014398509481988e+16, 1.0)
```

Théorème 7 (2Sum - Knuth)

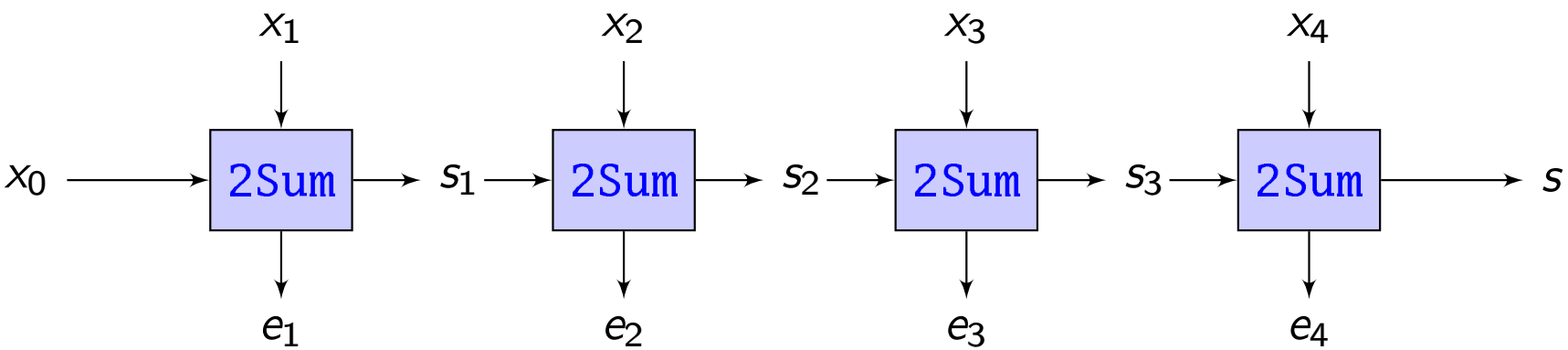
On considère deux VF x et y et l'algorithme suivant :

```
1  s  ← x ⊕ y
2  y_v ← s ⊖ x
3  x_v ← s ⊖ y_v
4  y_a ← y ⊖ y_v
5  x_a ← x ⊖ x_v
6  d  ← x_a ⊕ y_a
7  Retourner (s, d)
```

Alors $x + y = s + d$ avec $s = x \oplus y$ et $d = err(x \oplus y)$. De plus s et d ne se chevauchent pas.

UNIVERSITÉ DE NANTES

Somme compensée d'un nombre quelconque de flottants



Somme compensée d'un nombre quelconque de flottants

```
def sommeKahan(liste):
    (s,c) = (0.,0.)
    for x in liste:
        (s,c) = fast2sum(s, x + c)
    return s
```

Somme compensée d'un nombre quelconque de flottants

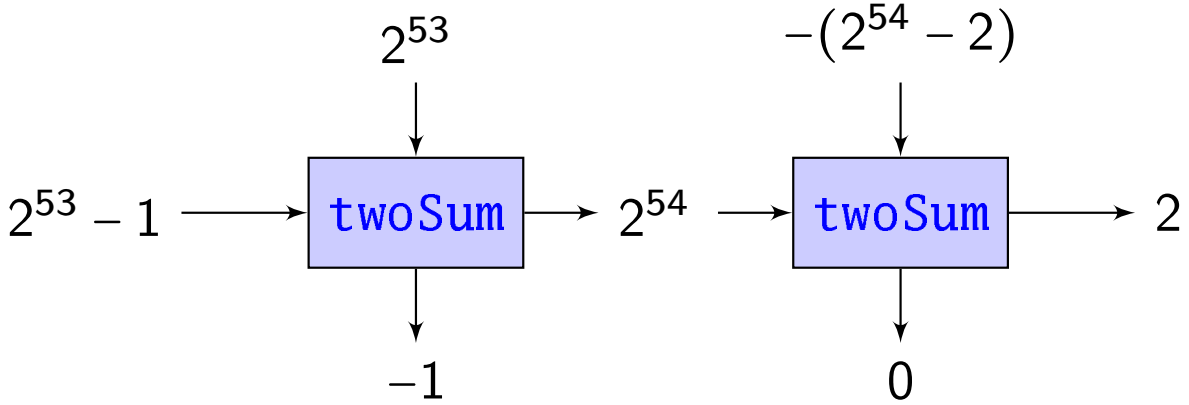
```
def sommeKahan(liste):
    (s,c) = (0.,0.)
    for x in liste:
        (s,c) = fast2sum(s, x + c)
    return s
```

```
def sommePichat(liste):
    s,e = 0.,0.
    for x in liste:
        (s,c) = fast2sum(s, x)
        e += c
    return s + e
```

UNIVERSITÉ DE NANTES

$x_0 = 2^{53} - 1$, $x_1 = 2^{53}$ et $x_2 = -(2^{54} - 2)$

$x_0 = 2^{53} - 1$, $x_1 = 2^{53}$ et $x_2 = -(2^{54} - 2)$



Exercice 4

Expliquez les résultats trouvés. Que va faire l'algorithme de somme compensée ? Et celui de somme en cascade ?

```
In [105]: sommePichat([1. / n for n in range(1,100001)])
Out[105]: 12.090146129863427

In [106]: sommeKahan([1. / n for n in range(1,100001)])
Out[106]: 12.090146129863427

In [107]: sum([1. / n for n in range(1,100001)])
Out[107]: 12.090146129863335

In [108]: sum([1. / n for n in range(100000,0,-1)])
Out[108]: 12.090146129863408
```

```
In [105]: sommePichat([1. / n for n in range(1,100001)])
Out[105]: 12.090146129863427

In [106]: sommeKahan([1. / n for n in range(1,100001)])
Out[106]: 12.090146129863427

In [107]: sum([1. / n for n in range(1,100001)])
Out[107]: 12.090146129863335

In [108]: sum([1. / n for n in range(100000,0,-1)])
Out[108]: 12.090146129863408
```

```
sage: sum(1. / n, n , 1 , 100000).n(64)
12.090146129863427
```

Banque chaotique

Exemple dû à Jean-Michel Muller.

Banque chaotique

Exemple dû à Jean-Michel Muller.

M. X a récemment été à sa banque (Chaotic Bank Society), pour connaître les nouvelles offres proposées aux meilleurs clients. Son banquier lui propose l'offre suivante : « vous déposez tout d'abord e - 1 euros sur votre compte (où e = 2.7182818... est la base du logarithme népérien). La première année, nous prenons 1 euro sur votre compte de frais de gestion. Par contre, la deuxième année est plus intéressante pour vous, car nous multiplions votre capital restant par 2 et prenons 1 euro de frais de gestion. La troisième année est encore plus intéressante, car nous multiplions votre capital par 3 et prenons 1 euro de frais de gestion. Et ainsi de suite : la n-ième année, nous multiplions votre capital par n et prenons 1 euro de frais de gestion. Intéressant, non ? » Pour prendre sa décision, M. X décide de demander l'aide d'un informaticien et se retourne vers vous.

UNIVERSITÉ DE NANTES

```
def chaotic(n):
    cpt = exp(1) - 1
    for i in range(1,n + 1):
        cpt = i*cpt - 1
    return cpt
```

```
def chaotic(n):
    cpt = exp(1) - 1
    for i in range(1,n + 1):
        cpt = i*cpt - 1
    return cpt
```

$$
\begin{aligned}
c_n &= n! \times \left(c_0 - 1 - \frac{1}{2!} - \frac{1}{3!} - \cdots - \frac{1}{n!} \right) \\
&= n! \times \left(c_0 - (e-1) + \frac{1}{(n+1)!} + \frac{1}{(n+2)!} + \cdots \right)
\end{aligned}
$$

```
def chaotic(n):
    cpt = exp(1) - 1
    for i in range(1,n + 1):
        cpt = i*cpt - 1
    return cpt
```

$$
\begin{aligned}
c_n &= n! \times \left(c_0 - 1 - \frac{1}{2!} - \frac{1}{3!} - \cdots - \frac{1}{n!} \right) \\
&= n! \times \left(c_0 - (e-1) + \frac{1}{(n+1)!} + \frac{1}{(n+2)!} + \cdots \right)
\end{aligned}
$$

Mais ceci est une autre histoire qui sera comptée l'an prochain...

UNIVERSITÉ DE NANTES