Erreurs de calcul des ordinateurs

Jocelyne Erhel

équipe FLUMINANCE, Inria Rennes et IRMAR, France

INSA, Mai 2016

Plan

- Introduction
- Arithmétique flottante
- Norme IEEE-754
- Phénomènes d'absorption et cancellation
- Algorithmes précis

Petit bestiaire des nombres

Ensembles de nombres

Nombres entiers naturels et relatifs
Nombres rationnels, fractions et nombres décimaux
Nombres réels, rationnels et irrationnels
Nombres complexes

Nombres décimaux

Ecriture en base 10: 10 chiffres $0, 1, \ldots, 9$
Décomposition avec les puissances de 10

Nombres binaires

Ecriture en base 2: 2 chiffres $0, 1$

Il y a 10 catégories de personnes, celles qui connaissent les nombres binaires et les autres.

Calculs en base 2 et nombres en base 10

Lu sur https://support.microsoft.com/fr-fr/kb/214118
Pour réduire les effets éventuels de l'inexactitude du stockage des nombres à virgule flottante, utilisez la fonction Round() pour arrondir les nombres au nombre de décimales requis par votre calcul.

Vous pouvez souvent empêcher les erreurs d'arrondi d'affecter votre travail grâce à l'option *Calcul avec la précision au format affiché*. Cette option force chaque nombre de la feuille de calcul à prendre la précision affichée.
Remarque L'utilisation de l'option *Calcul avec la précision au format affiché* peut avoir des effets de calculs cumulatifs susceptibles de rendre vos données de plus en plus inexactes au fil du temps. N'utilisez cette option que si vous êtes certain que la précision affichée permettra à vos données de rester exactes.

Arithmétique flottante

Les nombres flottants en écriture décimale

Ecriture normalisée

$x = 123,456$ peut s'écrire $123456\ 10^{-3}$ ou $0,123456\ 10^{3}$ ou ...
L'écriture normalisée a un chiffre non nul avant la virgule: $1,23456\ 10^{2}$
$1,23456$ est la mantisse et 2 est l'exposant

Décomposition décimale

La base arithmétique est 10
Les chiffres sont $0, 1, 2, ..., 9$
$1,23456 = 1\ 10^{0} + 2\ 10^{-1} + 3\ 10^{-2} + 4\ 10^{-3} + 5\ 10^{-4} + 6\ 10^{-5}$

Précision

Par exemple, format flottant avec 5 chiffres après la virgule
Le nombre qui précède $1,23456$ est $1,23455$ et le nombre qui suit est $1,23457$
la différence entre deux nombres consécutifs est 10^{-5}
C'est la précision du format flottant

Les nombres flottants dans une base quelconque

Le format flottant est défini par

- la base b
- le nombre de chiffres de la mantisse p
- la plage d'exposants $E_{min} \ldots E_{max}$

Un nombre flottant est défini par

$$x = (-1)^s b^e a_0.a_1 a_2 \ldots a_p$$

avec $a_0 \neq 0$ (écriture normalisée)

La précision du format flottant est $\epsilon = b^{-p}$

Quelques propriétés du système flottant

L'ensemble des flottants est symétrique par rapport à 0
Le nombre de nombres flottants est fini

Il faut arrondir les nombres réels

Le plus grand nombre flottant est $x_{max} = b^{E_{max}} m_{max}$
Le plus petit nombre flottant > 0 est $u = b^{E_{min}}$

Il faut prévoir les dépassements vers $\pm\infty$ et vers 0

Arrondis - encadrement d'un réel

Exemples

format : $b = 10$ et $p = 3$ donc $\epsilon = 10^{-3}$
$x = 1.23456$ est encadré par $x_1 = 1.234$ et $x_2 = 1.235$
et $|x_2 - x_1| = 10^{-3} = \epsilon \leq |x|\epsilon$
le nombre flottant le plus proche est x_2

$x = -765.4321 = -10^2 \times 7.654321$ est encadré par $x_1 = -10^2 \times 7.655$ et $x_2 = -10^2 \times 7.654$
et $|x_2 - x_1| = 10^2 \times 10^{-3} \leq |x|\epsilon$
le nombre flottant le plus proche est x_2

Cas général

Soit x un nombre réel (ni trop petit ni trop grand)
x est encadré par 2 nombres flottants consécutifs x_1 et x_2

$$x_1 \leq x \leq x_2 \text{ et } |x_2 - x_1| \leq \epsilon|x|$$

On peut arrondir x soit vers x_1 soit vers x_2

Modes d'arrondi - propriétés

Propriétés d'un arrondi

Soit fl une fonction des réels (ni trop petits ni trop grands) vers les flottants
Pour que fl soit un arrondi, il doit avoir les propriétés suivantes:
monotonie : x, y réels, $x \leq y \Rightarrow \ fl(x) \leq fl(y)$
projection : x flottant $\Rightarrow fl(x) = x$
précision : $\frac{|fl(x)-x|}{|x|} \leq \epsilon$

L'erreur relative d'un arrondi est la précision machine

4 modes d'arrondi

$fl(x)$ est l'un des flottants x_1 ou x_2 qui encadrent x
Arrondi au plus près: précision en $\epsilon/2$
Arrondis vers $-\infty$, et vers $+\infty$: bornes sûres et arithmétique d'intervalles
Arrondi vers 0 ou troncature

Opérations arithmétiques

Exemples

Format : $b = 10,\ p = 3$

$x = 1.234$ et $y = 8.103$ alors $x \times y = 10.528488$

$r = 10 \times 8.532$ et $s = 5.276$ alors $r - s = 80.044$

Il faut arrondir le résultat d'une opération

$fl(x \times y) = 10 \times 1.053$

$fl(r - s) = 10 \times 8.004$

Opération correcte

Le résultat de l'opération entre deux nombres flottants
est l'arrondi du résultat exact
lorsqu'il n'est ni trop petit ni trop grand

Propriétés des opérations arithmétiques

Propriétés conservées

0 est élément neutre de l'addition
1 est élément neutre de la multiplication
x nombre flottant, $x - x = 0$
l'addition et la multiplication sont commutatives

Propriétés NON conservées

L'addition n'est pas associative
La multiplication n'est pas associative
La multiplication n'est pas distributive par rapport à l'addition
Il existe x nombre flottant, $x \times (1/x) \neq 1$

Erreur relative d'une opération

Sans overflow et sans underflow:
$fl(x + y) = (x + y)(1 + \alpha)$ avec $|\alpha| \leq \epsilon/2$
$fl(x \times y) = (x \times y)(1 + \beta)$ avec $|\beta| \leq \epsilon/2$
$fl(\sqrt{x}) = (\sqrt{x})(1 + \gamma)$ avec $|\gamma| \leq \epsilon/2$

Dans une boulangerie française le 1 janvier 2002

Sans un seul Euro en poche, mais le porte monnaie plein de Francs, un client entre chez sa boulangère préférée pour lui présenter ses meilleurs voeux et acheter une baguette bien croustillante.

1 Euro = 6,5596 Francs et 1 Franc = 0,1524 Euros

C'est combien la baguette maintenant ?
4,30 Francs comme d'habitude

Oui mais à partir d'aujourd'hui c'est en Euros.
Ah, j'oubliais (un petit coup d'EuroCalculette) : ça fait 0,66 Euros.

0,66 Euros (un petit coup d'EuroCalculette)
mais ça fait 4,33 Francs, ma baguette a augmenté de 3 centimes !

Dans une boulangerie française le 1 janvier 2002

Sans un seul Euro en poche, mais le porte monnaie plein de Francs, un client entre chez sa boulangère préférée pour lui présenter ses meilleurs voeux et acheter une baguette bien croustillante.
1 Euro = 6,5596 Francs et 1 Franc = 0,1524 Euros

C'est combien la baguette maintenant ?
4,30 Francs comme d'habitude

Oui mais à partir d'aujourd'hui c'est en Euros.
Ah, j'oubliais (un petit coup d'EuroCalculette) : ça fait 0,66 Euros.

0,66 Euros (un petit coup d'EuroCalculette)
mais ça fait 4,33 Francs, ma baguette a augmenté de 3 centimes !

Dans une boulangerie française le 1 janvier 2002

Sans un seul Euro en poche, mais le porte monnaie plein de Francs, un client entre chez sa boulangère préférée pour lui présenter ses meilleurs voeux et acheter une baguette bien croustillante.
1 Euro = 6,5596 Francs et 1 Franc = 0,1524 Euros

C'est combien la baguette maintenant ?
4,30 Francs comme d'habitude

Oui mais à partir d'aujourd'hui c'est en Euros.
Ah, j'oubliais (un petit coup d'EuroCalculette) : ça fait 0,66 Euros.

0,66 Euros (un petit coup d'EuroCalculette)
mais ça fait 4,33 Francs, ma baguette a augmenté de 3 centimes !

Dans une boulangerie française le 1 janvier 2002

Sans un seul Euro en poche, mais le porte monnaie plein de Francs, un client entre chez sa boulangère préférée pour lui présenter ses meilleurs voeux et acheter une baguette bien croustillante.
1 Euro = 6,5596 Francs et 1 Franc = 0,1524 Euros

C'est combien la baguette maintenant ?
4,30 Francs comme d'habitude

Oui mais à partir d'aujourd'hui c'est en Euros.
Ah, j'oubliais (un petit coup d'EuroCalculette) : ça fait 0,66 Euros.

0,66 Euros (un petit coup d'EuroCalculette)
mais ça fait 4,33 Francs, ma baguette a augmenté de 3 centimes !

Dans une boulangerie française le 1 janvier 2002 (suite)

Je vous donne une pièce de 5 Francs et vous me rendez la monnaie en Euros.
Bien, 5 Francs ça fait (EuroCalculette) 0,76 Euros.
Moins 0,66 Euros la baguette, je vous rends 10 centimes d'Euro.

Chouette, ma première pièce en Euro.
Alors (EuroCalculette) 0,10 Euros, cela fait 0,66 Francs, c'est marrant, comme le prix de la baguette en Euros.

Donc je vous ai donné 5 Francs, vous me rendez 0,66 Francs, ça met la baguette à 4,34 Francs ! Le prix a beaucoup augmenté !

Dans une boulangerie française le 1 janvier 2002 (suite)

Je vous donne une pièce de 5 Francs et vous me rendez la monnaie en Euros.
Bien, 5 Francs ça fait (EuroCalculette) 0,76 Euros.
Moins 0,66 Euros la baguette, je vous rends 10 centimes d'Euro.

Chouette, ma première pièce en Euro.
Alors (EuroCalculette) 0,10 Euros, cela fait 0,66 Francs, c'est marrant, comme le prix de la baguette en Euros.

Donc je vous ai donné 5 Francs, vous me rendez 0,66 Francs, ça met la baguette à 4,34 Francs ! Le prix a beaucoup augmenté !

Dans une boulangerie française le 1 janvier 2002 (suite)

Je vous donne une pièce de 5 Francs et vous me rendez la monnaie en Euros.
Bien, 5 Francs ça fait (EuroCalculette) 0,76 Euros.
Moins 0,66 Euros la baguette, je vous rends 10 centimes d'Euro.

Chouette, ma première pièce en Euro.
Alors (EuroCalculette) 0,10 Euros, cela fait 0,66 Francs, c'est marrant, comme le prix de la baguette en Euros.

Donc je vous ai donné 5 Francs, vous me rendez 0,66 Francs, ça met la baguette à 4,34 Francs ! Le prix a beaucoup augmenté !

Dans une boulangerie française le 2 janvier 2002

Le même client décide le lendemain d'acheter deux baguettes, toujours dans la même boulangerie.

Bonjour, je voudrais deux baguettes. C'est combien en Euros ?
Voyons 4,30 x 2 = 8,60 Francs, soit (EuroCalculette) 1,31 Euros.
Deux baguettes coûtent moins cher que 2 fois une baguette (2 x 0,66), c'est transcendant ce truc !

Je vous donne une pièce de 10 Francs, et comme hier, vous me rendez la monnaie en Euros.
Donc (Eurocalculette) 1,52 Euros moins 1,31 Euros, je vous rends 0,21 Euros.

Voyons voir, (Eurocalculette) 0,21 Euros, cela fait 1,38 Francs, j'ai donc payé (10-1,38)/2 = 8,62/2= 4,31 Francs la baguette.
C'est moins cher qu'hier mais plus qu'en 2001 !

Dans une boulangerie française le 2 janvier 2002

Le même client décide le lendemain d'acheter deux baguettes, toujours dans la même boulangerie.

Bonjour, je voudrais deux baguettes. C'est combien en Euros ?
Voyons 4,30 x 2 = 8,60 Francs, soit (EuroCalculette) 1,31 Euros.
Deux baguettes coûtent moins cher que 2 fois une baguette (2 x 0,66), c'est transcendant ce truc !

Je vous donne une pièce de 10 Francs, et comme hier, vous me rendez la monnaie en Euros.
Donc (Eurocalculette) 1,52 Euros moins 1,31 Euros, je vous rends 0,21 Euros.

Voyons voir, (Eurocalculette) 0,21 Euros, cela fait 1,38 Francs, j'ai donc payé (10-1,38)/2 = 8,62/2= 4,31 Francs la baguette.
C'est moins cher qu'hier mais plus qu'en 2001 !

Dans une boulangerie française le 2 janvier 2002

Le même client décide le lendemain d'acheter deux baguettes, toujours dans la même boulangerie.

Bonjour, je voudrais deux baguettes. C'est combien en Euros ?
Voyons 4,30 x 2 = 8,60 Francs, soit (EuroCalculette) 1,31 Euros.
Deux baguettes coûtent moins cher que 2 fois une baguette (2 x 0,66), c'est transcendant ce truc !

Je vous donne une pièce de 10 Francs, et comme hier, vous me rendez la monnaie en Euros.
Donc (Eurocalculette) 1,52 Euros moins 1,31 Euros, je vous rends 0,21 Euros.

Voyons voir, (Eurocalculette) 0,21 Euros, cela fait 1,38 Francs, j'ai donc payé (10-1,38)/2 = 8,62/2= 4,31 Francs la baguette.
C'est moins cher qu'hier mais plus qu'en 2001 !

Norme IEEE-754

Norme IEEE-754 - définition

- Norme en 1985: fin d'une belle pagaille
- Arithmétique en base 2
- Plusieurs formats de flottants binaires et de flottants décimaux
- Gestion des quatre arrondis
- Opérations correctes: addition, soustraction, multiplication, division, racine carrée
- Conversions correctes: binaire/décimal
- Gestion des exceptions: dépassements et opérations illicites
- Révision en 2008: des nouveautés
- Spécification des fonctions élémentaires
- Opération correcte: "Fuse Multiply-Add: FMA" $xy + z$

Formats de la norme IEEE-754

Codage des formats binaires - base $b = 2$

Format	nombre de bits	signe	exposant	mantisse
simple précision	32	1	8	23
double précision	64	1	11	52
quadruple précision	128	1	15	112

1er chiffre implicite (toujours 1 en binaire)
exposant biaisé pour avoir un entier positif

Caractéristiques des formats binaires

Format	p	E_{min}	E_{max}	ϵ	u	x_{max}
simple précision	23	-126	127	10^{-7}	10^{-38}	10^{+38}
double précision	52	-1022	1023	10^{-16}	10^{-308}	10^{+308}
quadruple précision	112	-16382	16383			

Conversion dans la norme IEEE-754

Formats décimaux: base 10

Format	p	E_{min}	E_{max}
simple précision	6	-95	96
double précision	15	-383	384
quadruple précision	33	-6143	6144

Exceptions

Nombres spéciaux

- Exposant nul (tous les bits à 0) et mantisse nulle pour ± 0
- Exposant nul et mantisse non nulle pour les nombres dénormalisés
- Exposant maximal (tous les bits à 1) et mantisse nulle pour $\pm\infty$
- Exposant maximal et mantisse non nulle pour *NaN*

Underflow, Overflow, et Invalid

Type	Exemple	Résultat flottant
underflow	$u/2$	± 0 (*)
overflow	x_{max}^2	$\pm\infty$
invalid	$0/0$	*NaN*

(*): underflow graduel vers 0 grâce aux nombres dénormalisés

Calculs avec les nombres spéciaux

Les exceptions génèrent des nombres spéciaux et le calcul continue.

Calculs avec ∞ et avec 0

- $1/0 = \infty$
- $2 + \infty = \infty$
- $3 \times \infty = \infty$
- $1/\infty = 0$

Calculs avec *NaN*

- $0/0 = NaN$
- $\infty - \infty = NaN$
- $0 \times \infty = NaN$
- $3 + NaN = NaN$
- toute comparaison avec un *NaN* retourne faux sauf
- si $x = NaN$ alors $x \neq x$ retourne vrai

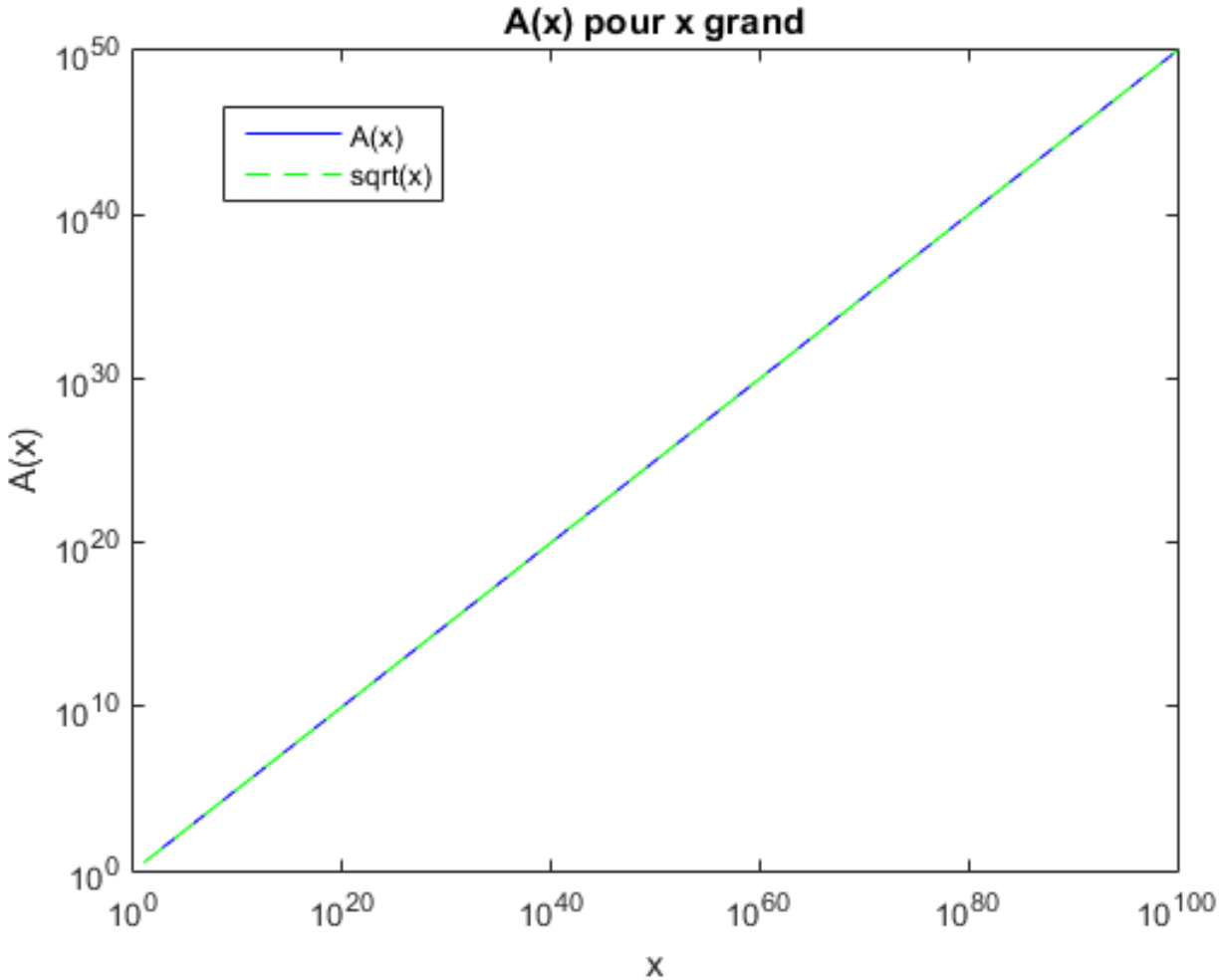
Limite mathématique d'une fonction

Exemple dû à W. Kahan

$$A(x) = \frac{x^2}{\sqrt{x^3 + 1}}$$

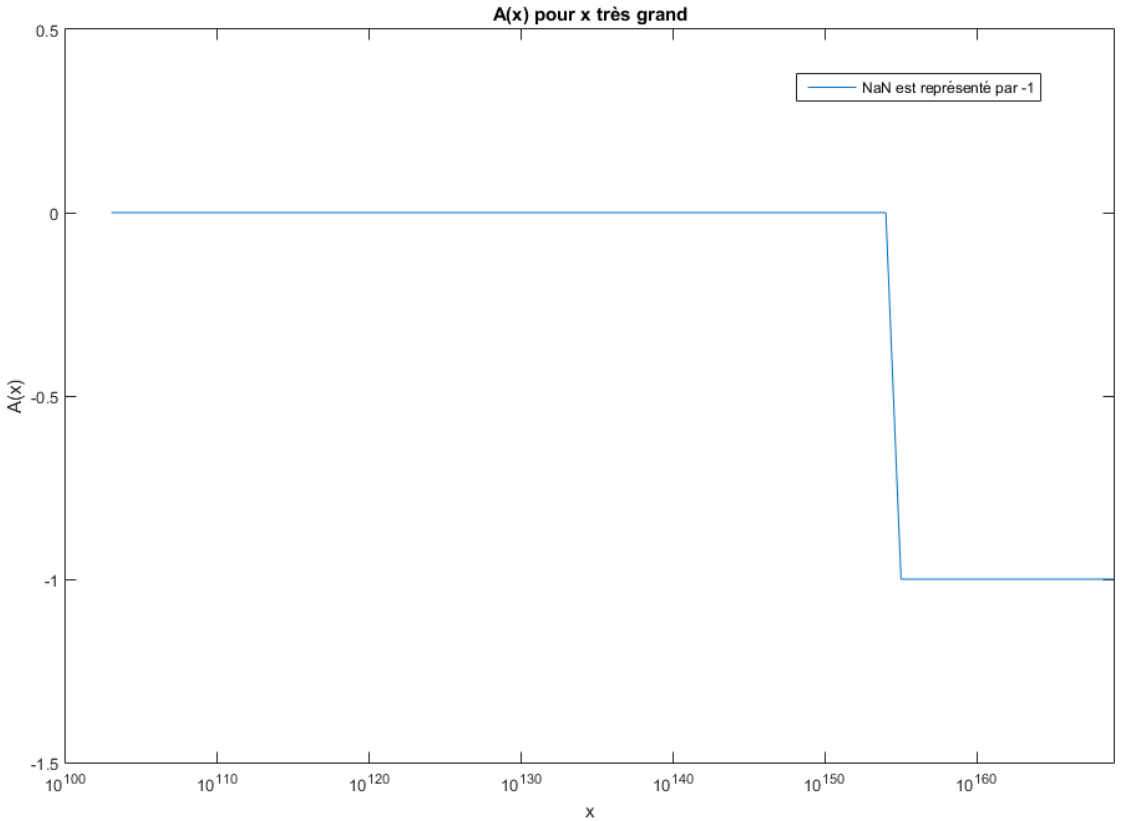
Mathématiquement $A(x)$ est équivalent à $\sqrt{x}$ quand x tend vers $+\infty$
Peut-on le vérifier numériquement ?

Limite numérique d'une fonction



Numériquement, on vérifie que $A(x)$ est équivalent à $\sqrt{x}$ (jusqu'à $x = 10^{100}$)

Limite numérique d'une fonction (suite)



Numériquement, on vérifie que $A(x) = 0$
(de $x = 10^{103}$ à $x = 10^{154}$)
Mais on vérife aussi que $A(x) = NaN$ pour $x \geq 10^{155}$

Intérêt de la norme IEEE-754

- Même résultat d'une machine à l'autre (en principe)
- Preuve de stabilité numérique des algorithmes
- Construction d'algorithmes précis
- Arithmétique d'intervalles avec arrondis vers $\pm\infty$
- Recommandation pour les fonctions élémentaires usuelles

Produit de n nombres flottants

$P = x_1 \times x_2 \ldots \times x_n$ avec $x_1, x_2, \ldots, x_n$ nombres flottants
Pas d'overflow, underflow, invalid

Algorithme de calcul

$$\begin{cases} P_1 = x_1 \\ P_i = P_{i-1} \times x_i, \ i = 2, \ldots, n \\ P = P_n \end{cases}$$

Erreurs d'arrondis

$fl(P_i) = fl(P_{i-1})\, x_i(1+\beta_i)$ avec $|\beta_i| \leq 1/2\epsilon$
$fl(P_n) = P(1+\beta_2)\ldots(1+\beta_n)$
$\frac{|fl(P)-P|}{|P|} = |(1+\beta_2)\ldots(1+\beta_n) - 1|$
$(1-\epsilon/2)^{n-1} - 1 \leq \frac{|fl(P)-P|}{|P|} \leq (1+\epsilon/2)^{n-1} - 1$
Au premier ordre en ϵ, on obtient: $\frac{|fl(P)-P|}{|P|} \simeq \frac{n-1}{2}\epsilon$

L'erreur relative d'un produit de n nombres est environ $n-1$ fois l'erreur relative de l'arrondi au plus près.

Absorption et cancellation

Phénomène d'absorption

Exemple

en base 10, avec 3 chiffres de mantisse après la virgule
$fl(10^5 + 10) = fl(10^5(1 + 10^{-4})) = 10^5$

Absorption

Dès que y est beaucoup plus petit que x,
alors $x + y$ est arrondi à x.

Un petit nombre est absorbé par un grand nombre

Absorption et associativité

Calcul de $s = -x + x + y = y$

Avec x et y deux nombres flottants tels que y est absorbé par x.

Algorithme 1 : $s = (-x + x) + y$

```
z=-x+x
s=z+y
```

Algorithme 2 : $s = -x + (x + y)$

```
z=x+y
s= -x+z
```

L'algorithme 1 calcule juste, tandis que l'algorithme 2 calcule faux.

Deux algorithmes qui sont mathématiquement équivalents
ne sont pas numériquement équivalents.

Absorption et associativité

Calcul de $s = -x + x + y = y$

Avec x et y deux nombres flottants tels que y est absorbé par x.

Algorithme 1 : $s = (-x + x) + y$	Algorithme 2 : $s = -x + (x + y)$
z=-x+x	z=x+y
s=z+y	s= -x+z

L'algorithme 1 calcule juste, tandis que l'algorithme 2 calcule faux.

Deux algorithmes qui sont mathématiquement équivalents
ne sont pas numériquement équivalents.

Série harmonique sur ordinateur

Théorème vérifié sur ordinateur en calcul flottant

Si $u_n \geq 0$ et si $\lim_{n\to+\infty} u_n = 0$, alors la série de terme général u_n est convergente, ie $\lim_{n\to+\infty} \sum_{i=1}^{n} u_i < \infty$

Théorème appris en mathématiques

$\lim_{n\to+\infty} \sum_{i=1}^{n} \frac{1}{i} = \infty$

Quel est le théorème le plus sûr ?

Série harmonique (suite)

Algorithme 1 de calcul

$$\begin{cases} S_1 = u_1 \\ S_i = S_{i-1} + u_i, \ i = 2, \ldots, n \\ S = S_n \end{cases}$$

Pour n assez grand $n \geq N$, u_n est absorbé par S_n donc $\forall n \geq N, \ S_n = S_N$

Exemple $u_n = 1/n$

Cancellation catastrophique

Exemple base $b = 10$ précision $p = 2$

$x = 1,2751$ et $y = 1,2749$ et $s = x - y = 2\ 10^{-4}$
$fl(x) = 1,28$ et $fl(y) = 1,27$ donc $fl(s) = fl(x) - fl(y) = 2\ 10^{-2}$
$\frac{|fl(s)-s|}{|s|} = \frac{10^{-2}-10^{-4}}{10^{-4}} \simeq 100$

Cancellation

Deux nombres réels x et y arrondis ; $s = x - y$ avec s petit par rapport à x et à y
$fl(s) = fl(x) - fl(y)$
$fl(s) = x(1+\alpha) - y(1+\beta) = x - y + x\,\alpha - y\,\beta$ avec $|\alpha| \leq \epsilon/2$ et $|\beta| \leq \epsilon/2$
$\frac{|s-(x-y)|}{|s|} \leq \epsilon/2 \frac{(|x|+|y|)}{|s|}$
L'erreur relative est grande si $|s|$ est petit par rapport à $|x| + |y|$.
La soustraction est exacte mais fait ressortir les erreurs sur les opérandes.

La cancellation amplifie les erreurs de calcul

Produit de deux nombres complexes

Deux nombres complexes $z_1 = x_1 + i\ y_1$ et $z_2 = x_2 + i\ y_2$ avec x_1, y_1, x_2, y_2 flottants

$z_1 \times z_2 = (x_1 \times x_2 - y_1 \times y_2) + i\ (y_1 \times x_2 + x_1 \times y_2)$

Soit $D = x_1 \times x_2 - y_1 \times y_2$

Algorithme 1

$$\begin{cases} P = x_1 \times x_2 \\ Q = y_1 \times y_2 \\ D = P - Q \end{cases}$$

$$\begin{cases} fl(P) = x_1 \times x_2(1 + \beta_1) \\ fl(Q) = y_1 \times y_2(1 + \beta_2) \\ fl(D) = (fl(P) - fl(Q))(1 + \beta_3) \end{cases}$$

$fl(D) = (D + x_1 \times x_2\beta_1 - y_1 \times y_2\beta_2)(1 + \beta_3)$

$|fl(D) - D| \leq \epsilon/2(|D| + |x_1||x_2| + |y_1||y_2|) + \epsilon^2/4(|x_1||x_2| + |y_1||y_2|)$

L'erreur relative est grande si $|D|$ est petit par rapport à $|x_1||x_2| + |y_1||y_2|$.

Il y a un risque de cancellation et de calcul faux.

Produit de deux nombres complexes (suite)

$D = x_1 \times x_2 - y_1 \times y_2$

Algorithme 2

$$\begin{cases} Q = y_1 \times y_2 \\ D = x_1 \times x_2 - Q(\text{FMA}) \end{cases}$$

$$\begin{cases} fl(Q) = y_1 \times y_2(1+\beta_1) \\ fl(D) = (x_1 \times x_2 - fl(P))(1+\beta_2) \end{cases}$$

$fl(D) = (D - y_1 \times y_2\beta_1)(1+\beta_2)$
$|fl(D) - D| \leq \epsilon/2(|D| + |y_1||y_2|) + \epsilon^2/4|y_1||y_2|$
L'erreur relative est grande si $|D|$ est petit par rapport à $|y_1||y_2|$.

Il y a un risque de cancellation et de calcul faux.

Algorithmes de calcul précis

Voir les travaux de Jean-Michel Müller

Calul de l'erreur sur la multiplication

Produit de deux nombres flottants

$P = x \times y$ avec x et y nombres flottants, sans overflow, underflow, invalid
Le résidu vaut $r = P - fl(P)$
On montre que le résidu est un nombre flottant. Donc $r = fl(r)$

Algorithme de calcul du résidu: une multiplication et un FMA

$$\begin{cases} P = x \times y \\ r = x \times y - P \end{cases}$$

Le calcul donne le résidu exact.

Algorithme TwoMultFMA: $(P, r) = TwoMultFMA(x, y)$

Retour vers les produits de nombres complexes

$D = x_1 \times x_2 - y_1 \times y_2$

Algorithme 3

$$\begin{cases} Q = y_1 \times y_2 \\ r = Q - y_1 \times y_2 (\text{FMA}) \\ P = x_1 \times x_2 - Q (\text{FMA}) \\ D = P + r \end{cases}$$

On montre que $|D - fl(D)| \leq \epsilon |D|$

En compensant l'erreur sur la multiplication, on réduit l'erreur globale.

Calcul de l'erreur sur l'addition

Somme de deux nombres flottants

$S = x + y$ avec x et y nombres flottants, sans overflow, underflow, invalid

Le résidu vaut $r = S - fl(S)$

On montre que le résidu est un nombre flottant. Donc $r = fl(r)$

Algorithme de calcul du résidu: trois additions

$$\begin{cases} S = x + y \\ z = S - x \\ r = y - z \end{cases}$$

Le calcul donne le résidu exact.

Algorithme Fast2Sum: $(S, r) = Fast2Sum(x, y)$

Série harmonique (saison 2)

Calcul de la somme de n nombres

$S = \sum_{i=1}^{n} x_i$ avec x_i nombre flottant

Algorithme 2

$$\begin{cases} S_1 = x_1 \\ e_1 = 0 \\ \text{For} i = 2, n \\ y_i = x_i + e_{i-1} \\ (S_i, e_i) = Fast2Sum(S_{i-1}, y_i) \\ \text{EndFor} \\ S = S_n \end{cases}$$

L'erreur sur l'addition précédente est ajoutée à l'opérande x_i. Il y a compensation des erreurs.

L'algorithme 2 est plus précis que l'algorithme 1.

Série harmonique (saison 2 - suite)

Calcul de la somme de n nombres

$S = \sum_{i=1}^{n} x_i$ avec x_i nombre flottant

Algorithme 3

$$\begin{cases} S_1 = x_1 \\ r = 0 \\ Fori = 2, n \\ (S_i, r_i) = Fast2Sum(S_{i-1}, x_i) \\ r = r + r_i \\ EndFor \\ S = S_n + r \end{cases}$$

Les erreurs des additions sont cumulées et ajoutées au résultat final.

L'algorithme 3 est plus précis que l'algorithme 2.

Si $\epsilon\ n/2 < 1$ et sans overflow

$$|fl(S) - S| \leq \epsilon|S|/2 + \gamma^2 \sum_{i=1}^{n} |x_i|$$

avec $\gamma = \frac{(n-1)\epsilon}{2-(n-1)\epsilon}$

Conclusion

- Les algorithmes ne sont pas numériquement équivalents.
- Algorithmes précis avec des compensations d'erreurs.
- Bornes d'erreur sur le résultat du calcul.