Compter sur ordinateur : plus vite, plus loin, plus sûr

Nathalie Revol

INRIA, LIP, ENS-Lyon

Nathalie.Revol@ens-lyon.fr

Die, 19 octobre 2004

Algorithme. . .

suite d'instructions à effectuer pour accomplir une tâche

Pas d'implicite ni d'imprécision.
Syntaxe rigide quand on s'adresse à un ordinateur.

Algorithme. . .
recette du taboulé

Algorithme. . .
recette du taboulé (pour un humain)

1. Dans une terrine ajouter dans l'ordre en mélangeant à chaque fois : la semoule, les tomates coupées en dés avec leur jus, les oignons hachés, 3 cuillères de menthe hachée, 3 cuillères de persil haché, le jus des citrons, les olives et les raisins puis ajouter le reste des ingrédients : huile, sel et poivre.

2. Vérifier l'assaisonnement plusieurs fois. Laisser macérer 3 ou 4 heures (voire faire la veille).

3. En décoration quelques feuilles de menthe en bouquet une tomate en quartier et un petit oignon en quartier.

Recette du taboulé : traduction en langage algorithmique

1. Dans une terrine ajouter dans l'ordre en mélangeant à chaque fois : la semoule, les tomates coupées en dés avec leur jus, les oignons hachés, 3 cuillères de menthe hachée, 3 cuillères de persil haché, le jus des citrons, les olives et les raisins puis ajouter le reste des ingrédients huile , sel et poivre.
   ajouter 250g de semoule dans la terrine
   mélanger
   couper 500g de tomates en dés
   ajouter les 500g de tomates coupées en dés avec leur jus dans la terrine
   mélanger
   hacher 250g d'oignons
   ajouter les 250g d'oignons hachés dans la terrine
   mélanger

hacher 3 cuillères à soupe de menthe
ajouter les 3 cuillères à soupe de menthe hachée dans la terrine
mélanger
hacher 3 cuillères à soupe de persil
ajouter 3 cuillères à soupe de persil haché dans la terrine
mélanger
presser 3 citrons
ajouter le jus des 3 citrons dans la terrine
mélanger
ajouter 150g d'olives dans la terrine
mélanger
ajouter les 50g de raisins dans la terrine
mélanger
ajouter 8 cuillerées à soupe d'huile d'olive
ajouter 1/2 cuillerée à café de sel et 1/2 cuillerée à café de poivre
mélanger

2. Vérifier l'assaisonnement plusieurs fois. Laisser macérer 3 ou 4 heures (voire faire la veille).
   intraduisible !!!

3. En décoration quelques feuilles de menthe en bouquet une tomate en quartier et un petit oignon en quartier.
   poser au centre de la terrine quelques feuilles de menthe
   découper une tomate en quartier
   poser au centre de la terrine les quartiers de tomate
   découper un oignon (de 100g) en quartier
   poser au centre de la terrine les quartiers d'oignon

Et tout ceci en supposant que "peser", "hacher", "découper" etc ont déjà été précisément définis algorithmiquement !

Plan de l'exposé

- **Compter avec des entiers**
  - opérations arithmétiques (école primaire)
  - notation redondante
- **Compter en notation scientifique : les nombres flottants**
  - opération d'addition et arrondi
  - avantages et inconvénients
  - calculer le sinus ?
- **Compter sans se tromper !**
  - calculer avec des intervalles
  - avantages et inconvénients
  - arithmétique par intervalles et grande précision

Plan de l'exposé

- **Compter avec des entiers**
  - opérations arithmétiques (école primaire)
  - notation redondante
- **Compter en notation scientifique : les nombres flottants**
  - opération d'addition et arrondi
  - avantages et inconvénients
  - calculer le sinus ?
- **Compter sans se tromper !**
  - calculer avec des intervalles
  - avantages et inconvénients
  - arithmétique par intervalles et grande précision

Compter avec des entiers

Pour des humains ayant 10 doigts :

représentation des nombres entiers :

$$13 = 1 \times 10 + 3 \times 1 = 1 \times 10^1 + 3 \times 10^0$$

Pour des ordinateurs ayant 2 poings :

représentation des nombres entiers :

$$13_{10} = 1 \times 8 + 1 \times 4 + 0 \times 2 + 1 \times 1 = 1 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 1101_2$$

Compter avec des entiers : additionner

À la main :

$$\begin{array}{ccccc} & & ^{1} & ^{1} & \\ & 1 & 2 & 3 & 4 \\ + & & 5 & 6 & 7 \\ \hline & 1 & 8 & 0 & 1 \end{array}$$

Compter avec des entiers : additionner

Sur ordinateur : idem mais en base 2 :

$$
\begin{aligned}
1234_{10} &= 1_{\times 1024} + 0_{\times 512} + 0_{\times 256} + 1_{\times 128} + 1_{\times 64} + 0_{\times 32} + 1_{\times 16} + 0_{\times 8} + 0_{\times 4} + 1_{\times 2} + 0_{\times 1} \\
&= 10011010010_2 \\
567_{10} &= 1_{\times 512} + 0_{\times 256} + 0_{\times 128} + 0_{\times 64} + 1_{\times 32} + 1_{\times 16} + 0_{\times 8} + 1_{\times 4} + 1_{\times 2} + 1_{\times 1} \\
&= 11000110111_2
\end{aligned}
$$

$$
\begin{array}{cccccccccccc}
 & & & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & {\scriptstyle 1} & & {\scriptstyle 1} & {\scriptstyle 1} & & \\
 & 1 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 & 0 \\
+ & & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\
\hline
= & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 \\
 & {\scriptstyle \times 1024} & {\scriptstyle \times 512} & {\scriptstyle \times 256} & {\scriptstyle \times 128} & {\scriptstyle \times 64} & {\scriptstyle \times 32} & {\scriptstyle \times 16} & {\scriptstyle \times 8} & {\scriptstyle \times 4} & {\scriptstyle \times 2} & {\scriptstyle \times 1}
\end{array}
$$

Compter avec des entiers : problème sur ordinateur

$$\begin{array}{rcccc} & & {\scriptstyle 1} & {\scriptstyle 1} & \\ & 1 & 2 & 3 & 4 \\ + & & 5 & 6 & 7 \\ \hline & 1 & 8 & 0 & 1 \end{array}$$

Problème : la retenue !
Pourquoi ne pas commencer par la gauche, pour avoir tout de suite l'ordre de grandeur du résultat ?
Pourquoi ne pas additionner chacun deux chiffres simultanément ?

Solution : changer de système de numération !

Compter avec des entiers
Systèmes redondants de numération (Avizienis, 1961)

Principe : un même nombre peut avoir plusieurs représentations.
Avantage pour l'addition : on choisit toujours un chiffre qui permettra d'absorber la retenue (**absorber** au sens d'**empêcher de se propager**).

Comment : ne pas se limiter aux chiffres compris entre 0 et 9 :
- autoriser les chiffres supérieurs à 10 (par exemple de 0 à 19),
- autoriser les chiffres négatifs.

Exemple : en base 10 avec les chiffres de -6 à 6
1546 s'écrit 1546, $2(-5)46$ noté $2\bar{5}46$, $2(-4)(-6)6$ noté $2\bar{4}\bar{6}6$, $2\bar{4}5\bar{4}$, $16\bar{6}6$, $16\bar{5}\bar{4}\ldots$

Compter avec des entiers
Systèmes redondants de numération (Avizienis, 1961)

$$\begin{array}{rcccc} & 1 & 2 & 3 & 4 \\ + & & 5 & 6 & 7 \\ \hline & 1 & 7 & 9 & 11 \end{array}$$

soit $1790 + 11 = 1801.$

$$\begin{array}{rcccc} & 1 & 2 & 3 & 4 \\ + & & 5 & 6 & 7 \\ \hline & 2 & \bar{2} & 0 & 1 \end{array}$$

$2\bar{2}01 = 2000 - 200 + 00 + 1 = 1801.$

Systèmes redondants de numération

Avizienis, 1961

En base β, chiffres $\in \{-a, \cdots, +a\}$
avec a entier et $2a \geq \beta + 1$ et $a \leq \beta - 1$,
Le nombre x est représenté par $x_{n-1}x_{n-2}\cdots x_2 x_1 x_0$ si $x = \sum_{i=0}^{n-1} x_i \beta^i$.

Si $2a + 1 \geq \beta$ alors on peut représenter tous les entiers.
Si $2a + 1 > \beta$ alors un nombre peut avoir **plusieurs** représentations : le système est dit **redondant**.

Exemple : en base $\beta = 10$ avec $a = 6$, 1546 s'écrit 1546, $2(-5)46$ noté $2\bar{5}46$, $2(-4)(-6)6$ noté $2\bar{4}\bar{6}6$, $2\bar{4}\bar{5}\bar{4}$, $16\bar{6}6$, $16\bar{5}\bar{4}$...

Compter avec des entiers : multiplier

À la main :

$$
\begin{array}{cccccccc}
 & & & & 9 & 8 & 7 & 6 \\
\times & & & & 1 & 2 & 3 & 4 \\
\hline
 & & & 3 & 9 & 5 & 0 & 4 \\
 & & 2 & 9 & 6 & 2 & 8 & \\
 & 1 & 9 & 7 & 5 & 2 & & \\
 & 9 & 8 & 7 & 6 & & & \\
\hline
1 & 2 & 1 & 8 & 6 & 9 & 8 & 4
\end{array}
$$

Compter avec des entiers : multiplier

Sur ordinateur :

$$
\begin{array}{cccccccc}
 & & & & 9 & 8 & 7 & 6 \\
\times & & & & 1 & 2 & 3 & 4 \\
\hline
 & & & 3 & 9 & 5 & 0 & 4 \\
 & + & 2 & 9 & 6 & 2 & 8 & \\
\hline
 & & 3 & 3 & 5 & 7 & 8 & 4 \\
+ & 1 & 9 & 7 & 5 & 2 & & \\
\hline
 & 2 & 3 & 1 & 0 & 9 & 8 & 4 \\
+ & 9 & 8 & 7 & 6 & & & \\
\hline
1 & 2 & 1 & 8 & 6 & 9 & 8 & 4
\end{array}
$$

Compter avec des entiers : multiplier

Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
172	28

Compter avec des entiers : multiplier

Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
172	28
344	14

Compter avec des entiers : multiplier
Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
172	28
344	14
688	7

Compter avec des entiers : multiplier

Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
172	28
344	14
688	7
1376	3

Compter avec des entiers : multiplier
Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
172	28
344	14
688	7
1376	3
2752	1

Compter avec des entiers : multiplier
Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
~~172~~	~~28~~
~~344~~	~~14~~
688	7
1376	3
2752	1

Compter avec des entiers : multiplier

Multiplication des paysans russes

ou comment éviter d'apprendre ses tables

$$57 * 86 = 4902$$

86	57
~~172~~	~~28~~
~~344~~	~~14~~
688	7
1376	3
2752	1
4902	

Compter avec des entiers : diviser

Division

$$
\begin{array}{r|l}
990 & 252 \\ \cline{2-2}
-756 & \quad 3 \\ \cline{1-1}
234 &
\end{array}
\qquad\qquad
\begin{array}{r|l}
252 & 234 \\ \cline{2-2}
-234 & \quad 1 \\ \cline{1-1}
18 &
\end{array}
\qquad\qquad
\begin{array}{r|l}
234 & 18 \\ \cline{2-2}
-18 & \quad 13 \\ \cline{1-1}
54 & \\
-54 & \\ \cline{1-1}
0 &
\end{array}
$$

Compter avec des entiers : diviser le bug du Pentium en 1994

Algorithme utilisé : celui qu'on apprend à l'école.

Bug du Pentium : T. Nicely (Virginie, USA), en octobre 1004, a constaté que son ordinateur donnait un résultat erroné lors du calcul de $1.0/824633702441$: erreur à partir du 12e chiffre.
T. Coe a découvert le pire cas : $41195835.0/3145727.0$ donnait $1.333\underline{739068902}$ au lieu de $1.333\underline{820449136}$ (faux à partir du 5e chiffre).

Explication : pour aller plus vite, utilisation de notation redondante :
utilisation de chiffres supérieurs à la base pour les restes partiels
utilisation de chiffres négatifs pour le quotient
et erreur en mélangeant tout ça !

Plan de l'exposé

- **Compter avec des entiers**
  - opérations arithmétiques (école primaire)
  - notation redondante
- **Compter en notation scientifique : les nombres flottants**
  - opération d'addition et arrondi
  - avantages et inconvénients
  - calculer le sinus ?
- **Compter sans se tromper !**
  - calculer avec des intervalles
  - avantages et inconvénients
  - arithmétique par intervalles et grande précision

Compter en notation scientifique : les nombres flottants

Exemple : $\pi \simeq 3,14 \times 10^0 = 0,314 \times 10^1 = 0,031 \times 10^2 = 314 \times 10^{-2} \ldots$

Convention : on met un chiffre non nul avant la virgule : $\pi \simeq 3,14 \times 10^0$.

Représentation :
nombre de chiffres utilisés dans la représentation : constant, dans notre exemple, 3 chiffres décimaux.
Population de Die : 4668 habitants, représenté par $4,67 \times 10^3$.

Dénomination : notation scientifique dans la vie courante, nombre à virgule flottante ou par ellipse **nombre flottant**.

Compter en notation scientifique : additionner deux nombres flottants

Exemple : $4.67 \times 10^3 + 3,14 \times 10^0$?

$$
\begin{array}{rl}
 & 4,67 \quad \text{x } 10^3 \\
+ & 0,00314 \quad \text{x } 10^3 \\
\hline
 & 4,67314 \quad \text{x } 10^3 \\
\simeq & 4,67 \quad \text{x } 10^3
\end{array}
$$

Compter en notation scientifique : arrondir

Nombre de chiffres fixé $\longrightarrow$ arrondi obligatoire.

Comment arrondir ?

- vers le haut
- vers le bas
- vers 0
- en s'éloignant de 0
- au plus près : dans ce cas, comment arrondir 6,785 ?

Compter en notation scientifique : norme IEEE-754

Jusqu'en 1985 : anarchie totale dans le monde des processeurs.

En 1985 : adoption de la norme IEEE-754

Formats fixés :

simple précision : 24 bits de mantisse, 8 bits d'exposant, 1 bit de signe
$\longrightarrow$ 32 bits

signe	*exp.*	*mantisse*
1	8	24

double précision : 53 bits de mantisse, 11 bits d'exposant, 1 bit de signe
$\longrightarrow$ 64 bits

signe	*exp.*	*mantisse*
1	11	53

Compter en notation scientifique : norme IEEE-754

Jusqu'en 1985 : anarchie totale dans le monde des processeurs.

En 1985 : adoption de la norme IEEE-754

- formats fixés :
  simple précision et double précision
- arrondis possibles : vers $+\infty$, vers $-\infty$, vers 0 et au plus près (pair)
- opérations arithmétiques : $+$, $-$, $\times$, $/$ et $\sqrt{}$ doivent rendre l'arrondi du résultat exact (toujours possible avec 3 bits supplémentaires pendant les calculs).

Compter en notation scientifique

Avantages :

- représentation en ordre de grandeur
- opérations en temps constant (puisque nombre de chiffres constant)
- norme IEEE-754 : opérations le plus précises possible
- norme IEEE-754 : reproductibilité des calculs d'un ordinateur à l'autre
- norme IEEE-754 : calculs bien spécifiés, possibilité de prouver qu'un programme est correct.

Inconvénients :

- résultats toujours arrondis
- précision limitée

Arithmétique flottante : comptons jusqu'à 6 (Higham)

$$2 - 1$$

$$\left(\frac{1}{\cos(100\pi+\pi/4)}\right)^2$$

$$3.0 * (tan(atan(10000000.0))/10000000.0)$$

$$\left(\cdots(\sqrt{\cdots\sqrt{4}})^2\cdots\right)^2 \text{ (20 fois)}$$

$$5 \times \frac{(1+e^{-100})-1}{(1+e^{-100})-1}$$

$$\frac{\ln(e^{6000})}{1000}$$

Arithmétique flottante : comptons jusqu'à 6 (Higham)

$2-1$	1.00000000000000000
$\left(\frac{1}{\cos(100\pi+\pi/4)}\right)^2$	2.00000000000001110
$3.0 * (tan(atan(10000000.0))/10000000.0)$	3.00000000030727567
$\left(\cdots(\sqrt{\cdots\sqrt{4}})^2\cdots\right)^2$ (20 fois)	4.00000000006294343
$5 \times \frac{(1+e^{-100})-1}{(1+e^{-100})-1}$	NaN
$\frac{\ln(e^{6000})}{1000}$	$+\infty$

Compter en notation scientifique encore de la recherche à faire

Opérateurs

division : algorithmes différents selon les processeurs

opérations : basse consommation

étudier les propriétés que l'on peut établir sur les opérations

Compter en notation scientifique encore de la recherche à faire

Fonctions élémentaires

fonctions élémentaires : les calculer vite et bien

système	$sin(10^{22})$ (Ng)
valeur exacte	-0.852200849767. . .
HP 48 GX	-0.852200849767
HP 700	0.0
IBM 3090/600S-VF AIX 370	0.0
Matlab 4.2c.1 Sparc	-0.8522
Matlab 4.2c.1 MacIntosh	0.8740
SG Indy	0.87402806
Sharp EL5806	-0.090748172
DEC Station 3100	NaN

Compter en notation scientifique encore de la recherche à faire

Fonctions élémentaires :

sinus, cosinus hyperbolique, arc-tangente, exponentielle, logarithme. . .

- les faire entrer dans la norme IEEE-754
- les calculer correctement

Flottants en précision multiple : plus de chiffres

Principe :

utiliser des représentations de longueur fixée
($\Rightarrow$ le produit de 2 nombres a la même longueur que les multiplicandes)
mais plus longues que celles des flottants machine.

Complexité :

$\leq$ celle du calcul sur des entiers de même longueur.

Avantage :

plus de précision.

Inconvénient :

toujours aucune garantie sur les résultats.

Plan de l'exposé

- **Compter avec des entiers**
  - opérations arithmétiques (école primaire)
  - notation redondante
- **Compter en notation scientifique : les nombres flottants**
  - opération d'addition et arrondi
  - avantages et inconvénients
  - calculer le sinus ?
- **Compter sans se tromper !**
  - calculer avec des intervalles
  - avantages et inconvénients
  - arithmétique par intervalles et grande précision

Compter sans se tromper arithmétique par intervalles

(Moore 1966, Kulisch 1983, Neumaier 1990, Rump 1994, Alefeld and Mayer 2000. . .)

Principe

Nombres remplacés par des intervalles.
π remplacé par $[3.14159, 3.14160]$

Contenu de mon porte-monnaie : entre 10 Euros et 20 Euros, $\in [10, 20]$ Euros.

Calculs

Compter sans se tromper arithmétique par intervalles

$$[10, 20] + [5, 10] = [15, 30]$$
$$[-2, 3] + [5, 7] = [3, 10]$$

$[-3, 2] * [-3, 2] = [-6, 9]$ est différent de $[-3, 2]^2 = [0, 9]$

$$[-3, 2]/[0.5, 1] = [-6, 4]$$

$$X \diamond Y = \{x \diamond y \;/\; x \in X,\; y \in Y\}$$

$$\exp[-2, 3] = [\exp(-2), \exp(3)]$$

car $\exp$ est une fonction croissante.

$$\sin[\pi/3, \pi] = [0, 1]$$

attention, $\sin$ n'est pas monotone.

Arithmétique par intervalles : avantages

Prise en compte des incertitudes

données expérimentales, résultats de mesures physiques, valeurs non exactement représentables.

Calcul garanti

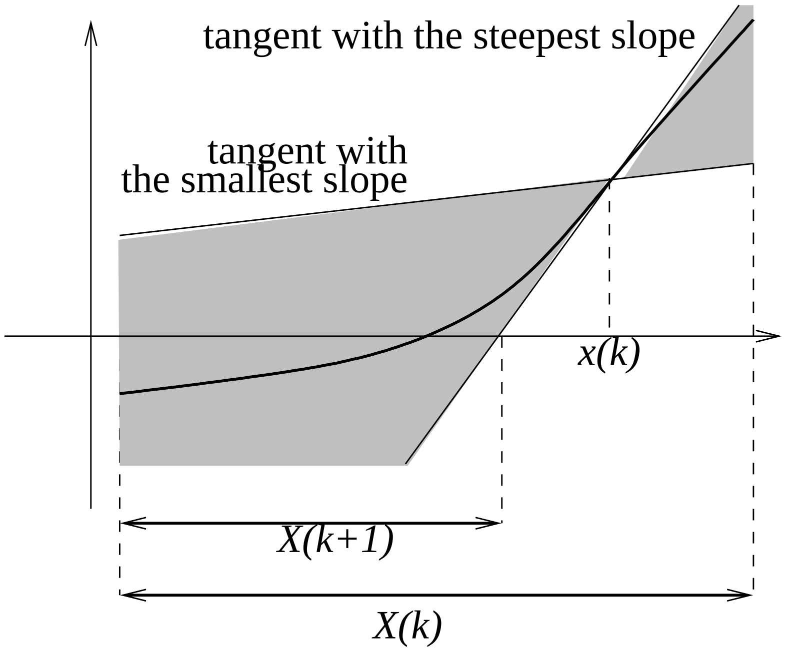
le résultat cherché appartient à l'intervalle calculé.

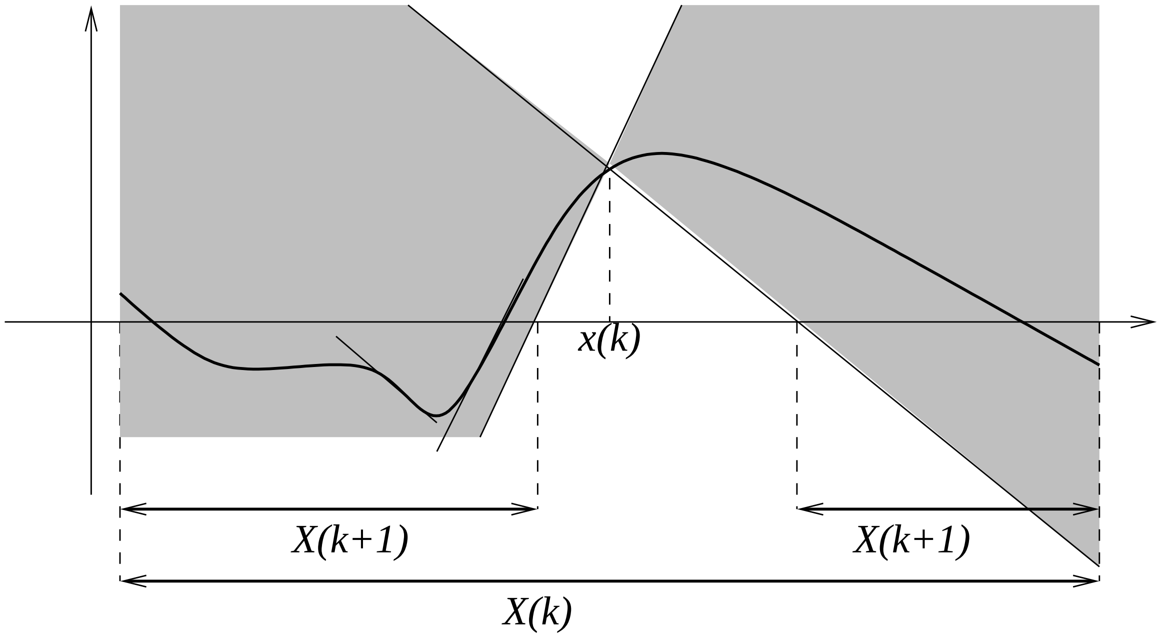
Information globale

on sait encadrer l'image d'une fonction sur tout un intervalle.

Arithmétique par intervalles : algorithme de Newton par intervalles

(Hansen & Greenberg 1983, Mayer 1995, van Hentenryck et al. 1997. . .)





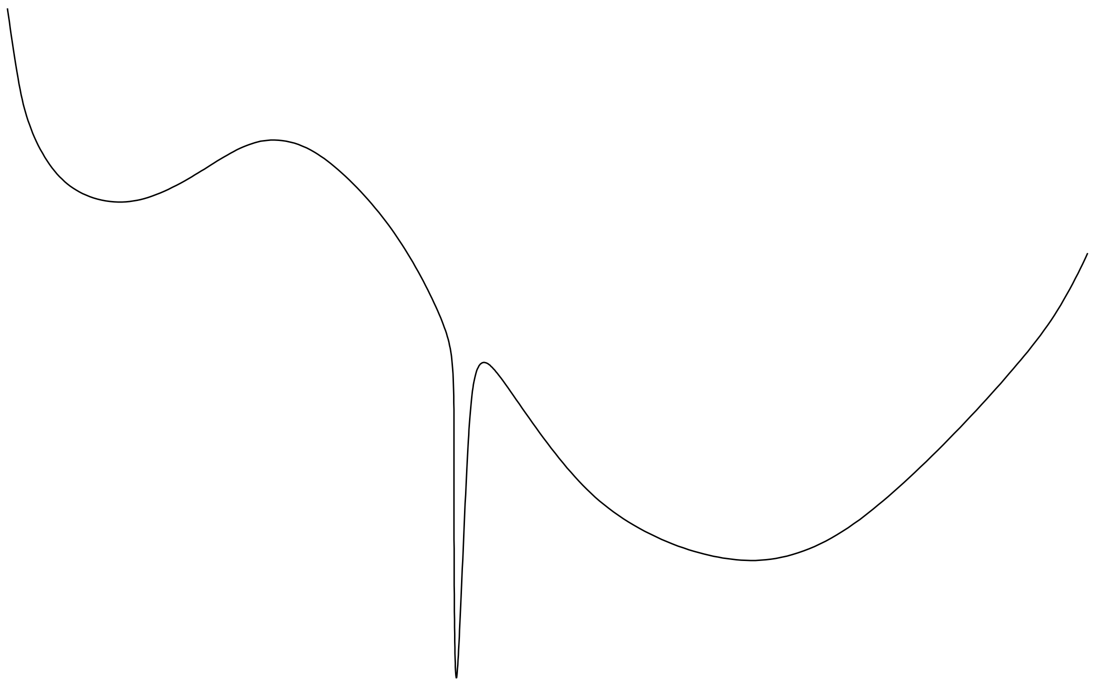
The result will be a list of intervals.

Théorème de Brouwer : si $f(I) \subset I$ alors f admet un point fixe dans I.

Arithmétique par intervalles : exemple d'algorithme

Algorithme de Hansen par intervalles

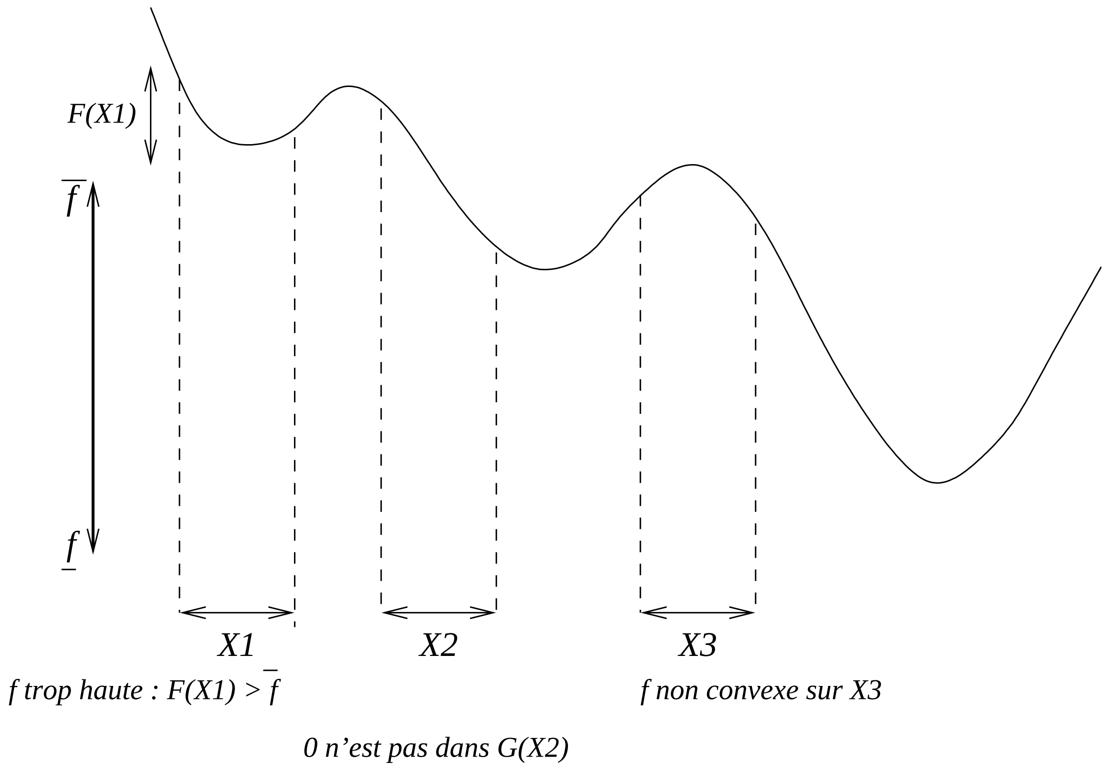
(Hansen 1992)



Arithmétique par intervalles : exemple d'algorithme

Algorithme de Hansen par intervalles

(Hansen 1992)



Arithmétique par intervalles : algorithme de Hansen

$\mathcal{L}$ = liste des pavés en attente := $\{X_0\}$

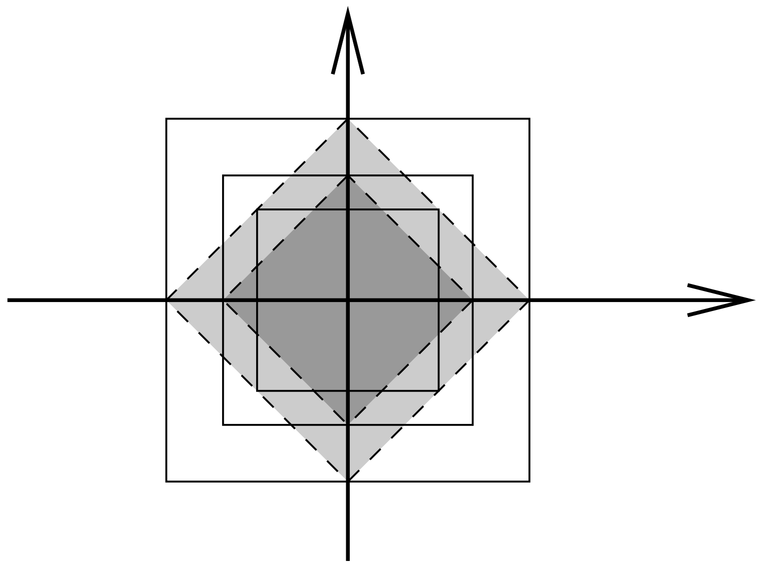
tant que $\mathcal{L} \neq \emptyset$ faire
- sortir X de $\mathcal{L}$
- **rejeter** X **?**
  - oui si $F(X) > \bar{f}$
  - oui si Grad$F(X) \not\ni 0$
  - oui si $HF(X)$ à diagonale non > 0
- **réduire** X
  - Newton sur le gradient
  - résoudre $Y \subset X$ tel que $F(Y) \leq \bar{f}$
- **couper** Y **en 2** : Y_1 et Y_2
  - ranger Y_1 et Y_2 dans $\mathcal{L}$

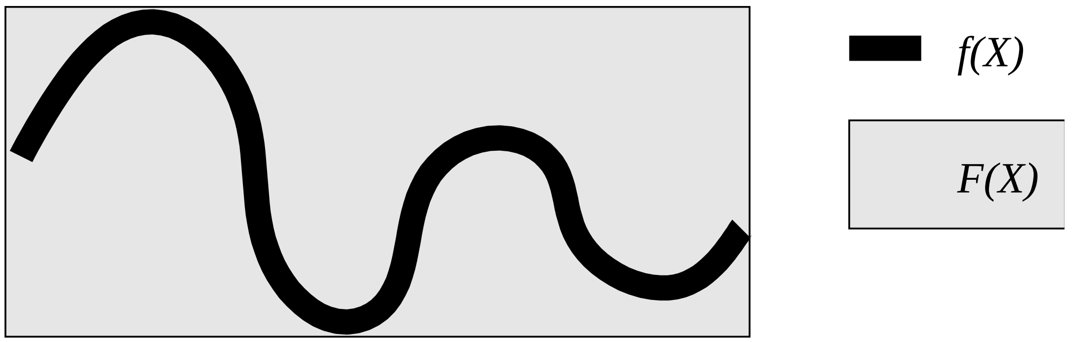
Arithmétique par intervalles : problèmes

Décorrélation des variables ou *variable dependency*

$$I - I = \{x - y \,;\, x \in I,\, y \in I\} \supset \{0\} = \{x - x \,;\, x \in I\}$$

Effet enveloppant ou *wrapping effect*





Arithmétique par intervalles
plus de précision

Idée :

calculer par intervalles pour conserver la garantie sur les résultats
utiliser des flottants en précision arbitraire pour gagner en précision.

Exemple : π

10 bits	$[3.140,$	$3.145]$
24 bits	$[3.1415925,$	$3.1415928]$
53 bits	$[3.141592653589793,$	$3.141592653589794]$
100 bits	$[3.141592653589793238462643383279,$	$3.1415926535897932384626433832 83]$

Arithmétique par intervalles
plus de précision

Aspirateur autonome

- par intervalles : ne casse rien mais passe loin du vase en porcelaine de Chine

- en précision arbitraire : passe près du vase en porcelaine de Chine et le casse peut-être

- par intervalles en précision arbitraire : passe près du vase en porcelaine de Chine et ne le casse pas !

Conclusion

On sait calculer

comme on a appris à l'école
les opérations arithmétiques exactes et flottantes

Pour calculer plus vite :

changer de système de numération
oublier les algorithmes qu'on a appris à l'école
et ne pas se tromper !

Pour calculer plus fiable

avoir une arithmétique bien spécifiée
avoir une arithmétique par intervalles

Conclusion et perspectives

Pour calculer plus vite, plus fiable : encore de la recherche à mener

pour calculer plus vite, avec moins de surface de silicium

pour consommer moins

pour bien spécifier les calculs

pour respecter les spécifications

pour calculer de façon garantie.

Références

Vulgarisation

- *Histoire d'algorithmes - Du caillou à la puce*, J.-L. Chabert et al., Belin 1994
- *Logique, informatique et paradoxes*, J.-P. Delahaye, Belin 1995
- *Histoire des codes secrets*, S. Singh, Le livre de poche 2001

Spécialisés

- *Qualité des calculs sur ordinateurs - Vers des arithmétiques plus fiables ?*, coordonné par M. Daumas et J.-M. Muller, Masson 1997
- *Modern computer algebra*, J. von zur Gathen and J. Gerhard, Cambridge University Press 1999
- *Arithmétique des ordinateurs*, J.-M. Muller, Masson 1989

Addition sans propagation de retenue

En base β avec des chiffres compris entre $-a$ et a , calcul de

$$s_n \cdots s_0 = x_{n-1} \cdots x_0 \; + \; y_{n-1} \cdots y_0$$

1. Calculer pour $i = 0 \cdots n-1$

$$\begin{array}{lcl} x_i + y_i & & \\ t_{i+1} & = & \left\{ \begin{array}{rlc} -1 & \text{si} & x_i + y_i \leq -a \\ 0 & \text{si} & -a+1 \leq x_i + y_i \leq a-1 \\ 1 & \text{si} & x_i + y_i \geq a \end{array} \right. \\ w_i & = & x_i + y_i - \beta t_{i+1} \end{array}$$

2. Calculer pour $i = 0 \cdots n$:

$$s_i = w_i + t_i \text{ avec } w_n = t_0 = 0.$$

Exemple : $\beta = 10$, $a = 6$, calcul de $x_i + y_i$

i		4	3	2	1	0
x_i		1	$\bar{2}$	5	3	$\bar{4}$
y_i		3	5	1	$\bar{5}$	$\bar{7}$
$x_i + y_i$		4	3	6	$\bar{2}$	$\bar{1}\bar{1}$
t_{i+1}						
w_i						
s_i						

Addition sans propagation de retenue

En base 10 avec des chiffres compris entre -6 et 6 , calcul de

$$s_n \cdots s_0 = x_{n-1} \cdots x_0 \ + \ y_{n-1} \cdots y_0$$

1. Calculer pour $i = 0 \cdots n-1$

$$\begin{array}{rcl} t_{i+1} & = & \left\{ \begin{array}{rlc} -1 & \text{si} & x_i + y_i \leq -a \\ 0 & \text{si} & -a+1 \leq x_i + y_i \leq a-1 \\ 1 & \text{si} & x_i + y_i \geq a \end{array} \right. \\ w_i & = & x_i + y_i - \beta t_{i+1} \end{array}$$

2. Calculer pour $i = 0 \cdots n$:

$$s_i = w_i + t_i \text{ avec } w_n = t_0 = 0.$$

Exemple : $\beta = 10$, $a = 6$, calcul de t_{i+1}

i		4	3	2	1	0
x_i		1	$\bar{2}$	5	3	$\bar{4}$
y_i		3	5	1	$\bar{5}$	$\bar{7}$
$x_i + y_i$		4	3	6	$\bar{2}$	$\bar{1}\bar{1}$
t_{i+1}	0	0	1	0	$\bar{1}$	
w_i						
s_i						

Addition sans propagation de retenue

En base β avec des chiffres compris entre $-a$ et a , calcul de

$$s_n \cdots s_0 = x_{n-1} \cdots x_0 \; + \; y_{n-1} \cdots y_0$$

1. Calculer pour $i = 0 \cdots n-1$

$$\begin{aligned} t_{i+1} &= \left\{ \begin{array}{rcc} -1 & \text{si} & x_i + y_i \le -a \\ 0 & \text{si} & -a+1 \le x_i + y_i \le a-1 \\ 1 & \text{si} & x_i + y_i \ge a \end{array} \right. \\ w_i &= x_i + y_i - \beta t_{i+1} \end{aligned}$$

2. Calculer pour $i = 0 \cdots n$:

$$s_i = w_i + t_i \text{ avec } w_n = t_0 = 0.$$

Exemple : $\beta = 10$, $a = 6$, calcul de w_i

i		4	3	2	1	0
x_i		1	$\bar{2}$	5	3	$\bar{4}$
y_i		3	5	1	$\bar{5}$	$\bar{7}$
$x_i + y_i$		4	3	6	$\bar{2}$	$\bar{1}\bar{1}$
t_{i+1}	0	0	1	0	$\bar{1}$	
w_i		4	3	$\bar{4}$	$\bar{2}$	$\bar{1}$
s_i						

Addition sans propagation de retenue

En base β avec des chiffres compris entre $-a$ et a , calcul de

$$s_n \cdots s_0 = x_{n-1} \cdots x_0 \; + \; y_{n-1} \cdots y_0$$

1. Calculer pour $i = 0 \cdots n-1$

$$\begin{array}{rcl} t_{i+1} & = & \begin{cases} -1 & \text{si} \quad x_i + y_i \leq -a \\ 0 & \text{si} \quad -a+1 \leq x_i + y_i \leq a-1 \\ 1 & \text{si} \quad x_i + y_i \geq a \end{cases} \\ w_i & = & x_i + y_i - \beta t_{i+1} \end{array}$$

2. Calculer pour $i = 0 \cdots n$:

$$s_i = w_i + t_i \text{ avec } w_n = t_0 = 0.$$

Exemple : $\beta = 10$, $a = 6$

i		4	3	2	1	0
x_i		1	$\bar{2}$	5	3	$\bar{4}$
y_i		3	5	1	$\bar{5}$	$\bar{7}$
$x_i + y_i$		4	3	6	$\bar{2}$	$\bar{1}\bar{1}$
t_{i+1}	0	0	1	0	$\bar{1}$	
w_i		4	3	$\bar{4}$	$\bar{2}$	$\bar{1}$
s_i		4	4	$\bar{4}$	$\bar{3}$	$\bar{1}$

Écriture usuelle : $x = 1\bar{2}53\bar{4} = 8526$, $y = 351\bar{5}\bar{7} = 35043$.
Somme $= 44\bar{4}\bar{2}\bar{1} = 43569$.

Addition dans un système redondant de numération

Les $t_i \simeq$ retenues de l'addition usuelle
MAIS pas de dépendance entre t_i et t_{i+1}
$\Rightarrow$ le calcul de chaque chiffre peut se faire en parallèle.
Complexité : $\mathcal{O}(1)$ si on a assez de ressources de calcul.

Remarque : si $\beta = 2$, les conditions $2a \geq b + 1$ et $a \leq b - 1$ ne peuvent pas être satisfaites simultanément.
Autres algorithmes, avec les chiffres $\bar{1}$, 0 et 1 pour la base 2.