Algorithme de Hansen pour l'optimisation globale

Nathalie Revol
Labo. ANO, Univ. Lille
et projet INRIA Arénaire, LIP, ENS-Lyon
Nathalie.Revol@ens-lyon.fr

Calcul Ensembliste, 01-02-2002

Problèmes d'erreurs d'arrondi
Scud et Patriot

cf. `http ://www.fas.org/spp/starwars/gao/im92026.htm`

The facts :

On February 25, 1991, a Patriot missile defense system operating at Dhahran, Saudi Arabia, during Operation Desert Storm failed to track and intercept an incoming Scud. This Scud subsequently hit an Army barracks killing 28 Americans.

Explanation :

software problem in the system's weapons control computer. This problem led to an inaccurate tracking calculation that became worse the longer the system operated. At the time of the incident, the battery had been operating continuously for over 100 hours.

Problèmes d'erreurs d'arrondi
Vancouver Stock Exchange

cf. page Web de Pete Stewart

The facts :

- 1982 : the Vancouver stock exchange introduced an index with nominal value 1000.000
- after each transaction, it was recomputed and truncated to the third place to the right of the decimal point
- after 22 months the index was 524.881
- the correct value was 1098.811

Explanation :

all rounding errors are in the same direction, downward ("truncation").

Problèmes d'erreurs d'arrondi
limites théoriques

Accuracy and Stability of Numerical Algorithms, N. Higham, 1996

Étude théorique de la stabilité d'une factorisation de matrice pleine (par exemple la factorisation LU ou QR) :
la stabilité est assurée si

$$n^3 u < 1$$

avec 1^+ le plus petit flottant (représentable en machine) > 1
et $u = 1^+ - 1$,
autrement dit OK si $n < 2 \times 10^5$ en IEEE double précision

Cette taille est quasiment atteinte désormais. . .
Remarque : de plus, le calcul est effectué en parallèle
(**Trading off parallelism and numerical stability, J.W. Demmel, 1993**).

Étude de la qualité numérique

- **“manuelle”**

CEA : simulation d’accidents nucléaires
Aérospatiale : ? ?

- **contrôle des erreurs**

Cadna : arithmétique stochastique
localisation des problèmes

- **correction des erreurs**

CENA : calcul des erreurs d’arrondi et correction
estimation au premier ordre + borne (algos linéaires)

Étude de la qualité numérique

- **arithmétique par intervalles**

principe : tout nombre est remplacé par un intervalle le contenant
ex : $\pi \in [3.14159; 3.14160]$, donnée $\in$ donnée mesurée $+[-\tau, \tau]$

$\oplus$ calcul certifié ou garanti
$\oplus$ estimation des erreurs d'arrondi (?)
$\oplus$ informations globales (cf. plus loin)

$\ominus$ **sur**-encadrements

Arithmétique par intervalles : informations globales

image d'une fonction (continue) sur tout un ensemble

surencadrement obtenu en englobant cet ensemble dans un intervalle I et en évaluant une expression de cette fonction sur I.

utilisation en optimisation globale :

si $F(I) > \bar{f}$ alors I ne peut pas contenir le minimum global.

théorème de Brouwer :

si $F(I) \subset I$ alors f admet un point fixe sur I

si inclusion stricte ($+$ d'autres conditions) alors unicité du point fixe.

Plan de l'exposé

- **Arithmétique par intervalles**
  - principe
  - évaluation d'une expression
- **Algorithme de Hansen**
  - résolution de systèmes linéaires
  - Newton par intervalles
  - algorithme de Hansen
- **Parallélisation de l'algorithme de Hansen**
  - méthodologie
  - résultats expérimentaux
- **Arithmétique par intervalles en précision multiple**
  - motivation
  - Newton par intervalles en précision arbitraire

Principes
de l'arithmétique par intervalles

- définition
- évaluation d'une fonction

Arithmétique par intervalles : définition (Moore 1966, Kulisch 1983. . .)

Nombres remplacés par des intervalles.

π remplacé par $[3.14159, 3.14160]$

Vecteurs remplacés par des vecteurs d'intervalles.

Matrices remplacées par des matrices d'intervalles.

$$\begin{pmatrix} [0,2] & [1,2] & [-5,-4] \\ [-1,1] & [2,3] & [1,5] \\ [-2,0] & [-2,1] & [-2,-1] \end{pmatrix}$$

Intérêt : incertitudes sur les données (de mesure. . .) prises en compte.

Arithmétique par intervalles : opérations

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

$[-3, 2] * [-3, 2] = [-6, 9]$ 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 : remarques

Remarque : avec une implémentation machine, utiliser des arrondis dirigés (cf. norme IEEE 754)

$$\sqrt{[2,3]} = [\triangledown\sqrt{2}, \triangle\sqrt{3}]$$

Avantage : tout résultat est garanti (à la différence des résultats flottants) le résultat du calcul appartient à l'intervalle résultat.

Problème : les opérations ont perdu certaines de leurs propriétés
$-$ n'est pas l'opposé de $+$:

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

$$[3,10] - [5,7] = [-4,5] \neq [-2,3]$$

(comme en flottant. . .)

Arithmétique par intervalles : évaluation d'une fonction

Soient $f : \mathbb{R} \rightarrow \mathbb{R}$ et I un intervalle.

Idéal : déterminer $F(I)$ le plus petit intervalle contenant $f(I)$.

Problème : f polynôme en plusieurs variables à coefficients rationnels
$f(I)$ cherché à ε près
problème NP-dur **(théorème de Gaganov)**.

Objectif plus réaliste : éviter d'obtenir des intervalles résultats trop larges.

Arithmétique par intervalles : évaluation d'une fonction

Soient $f : \mathbb{R} \to \mathbb{R}$ et I un intervalle.

Idéal : déterminer $F(I)$ le plus petit intervalle contenant $f(I)$.

Exemple : $f : x \mapsto x^2 - 2x + 1$, $I = [-1, 3]$

$I^2 - 2 * I + 1 = [-1, 3]^2 - 2[-1, 3] + 1 = [0, 9] + [-6, 2] + 1 = [-5, 12]$

et en écrivant $x^2 - 2 * x + 1 = x * (x - 2) + 1$

$$I * (I - 2) + 1 = [-1, 3] * [-3, 1] + 1 = [-9, 3] + 1 = [-8, 4]$$

alors que $F(I) = f(I) = [0, 4]$.

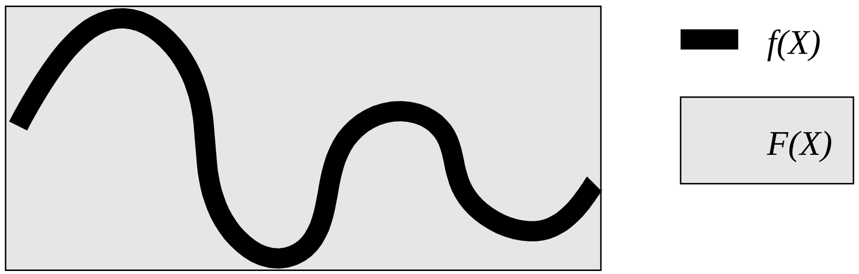
Arithmétique par intervalles : origines des problèmes

Dépendance des variables

$$I * I = \{x * y \,/\, x \in I,\ y \in I\} \text{ alors que } I^2 = \{x^2 \,/\, x \in I\}$$

dépendance entre le premier x et le second perdue.
décorrélation des variables.

Effet enveloppant



Plan de l'exposé

- **Arithmétique par intervalles**
  - principe
  - évaluation d'une expression
- **Algorithme de Hansen**
  - résolution de systèmes linéaires
  - Newton par intervalles
  - algorithme de Hansen
- **Parallélisation de l'algorithme de Hansen**
  - méthodologie
  - résultats expérimentaux
- **Arithmétique par intervalles en précision multiple**
  - motivation
  - Newton par intervalles en précision arbitraire

Résoudre un système linéaire $Ax = b$: éliminer l'élimination de Gauss

$$\begin{pmatrix} a_{1,1} & \dots & \dots \\ a_{2,1} & \dots & \dots \\ \vdots & & \vdots \end{pmatrix} \rightarrow \begin{pmatrix} a_{1,1} & \dots & \dots \\ 0 & \dots & \dots \\ 0 & \vdots & \vdots \end{pmatrix}$$

Élimination de Gauss : principe CL des lignes 1 et 2 pour éliminer $a_{2,1}$:

$$\begin{array}{rcl} l_2 & \leftarrow & l_2 - \frac{a_{2,1}}{a_{1,1}} l_1 \\ \text{c'est-à-dire } a_{2,1} & \leftarrow & a_{2,1} - a_{2,1} * \frac{a_{1,1}}{a_{1,1}} \end{array}$$

En intervalles

$$\begin{array}{rcl} a_{2,1} - a_{2,1} * \frac{a_{1,1}}{a_{1,1}} & \ni & 0 \\ & \neq & 0 \end{array}$$

Factorisation QR, méthodes de Krylov : notion d'orthogonalité inexistante si X et Y pavés dans $I\!R^n$, $X^t.Y \neq 0$.

Résoudre un système linéaire $Ax = b$: préconditionnement + itération contractante

Position du problème :
système linéaire $Ax = b$
on cherche un (sur-)ensemble de

$$\{x \;/\; \exists \mathcal{A} \in A,\ \exists \lfloor \in b, \mathcal{A}x = \lfloor\}$$

Méthode itérative contractante :
Jacobi, Gauss-Seidel (plus rapide) ou autre splitting
si $\rho(Z) < 1$

Préconditionnement pour la rendre contractante :
pré-multiplication du système $Ax = b$ par l'inverse de mid$(A)^{-1}$
(méthode de Hansen-Sengupta)

Algorithme de Newton par intervalles
Pourquoi une itération spécifique ?

Formule ponctuelle :

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)}$$

Équivalent intervalle :

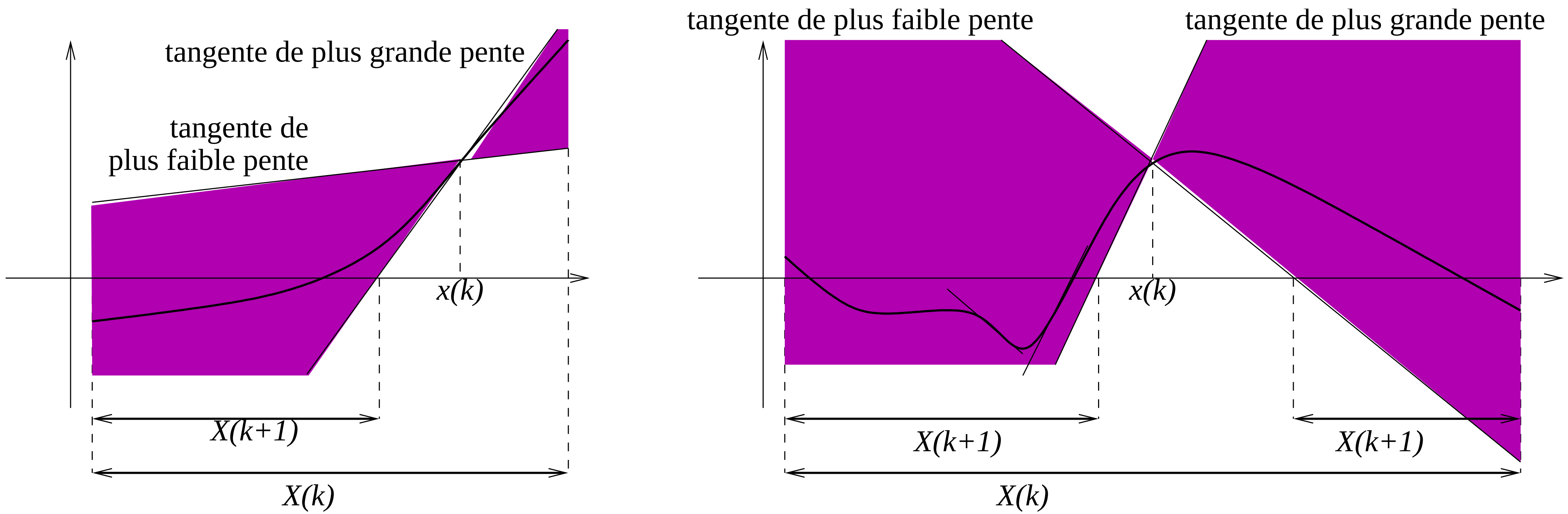
$$X_{k+1} = X_k - \frac{f(X_k)}{f'(X_k)}$$

$$w(X_{k+1}) = w(X_k) + w\left(\frac{f(X_k)}{f'(X_k)}\right) > w(X_k)$$

divergence !

Algorithme de Newton par intervalles
principe d'une itération

(Hansen & Greenberg 83, Baker Kearfott 95-97, Mayer 95, van Hentenryck et al. 97)



Algorithme de Newton par intervalles
propriétés

Prouver l'existence et l'unicité d'un zéro

OK s'il n'y a pas de trou et si le nouvel itéré (avant $\bigcap$) est contenu dans l'intérieur du précédent.

Idée : s'il y a une réduction stricte, l'itération est strictement contractante $\Rightarrow$ le théorème du point fixe s'applique.

Prouver l'existence d'un zéro

possible à l'aide du théorème des valeurs intermédiaires :

OK si $f(\inf(X))$ et $f(\sup(X))$ sont de signes opposés. (Théorème de Miranda en dimension supérieure).

OK grâce au théorème de Brouwer : s'il y a une réduction (même non stricte), l'itération est contractante $\Rightarrow$ le théorème du point fixe s'applique.

Optimiser une fonction continue

Problème

$f\ :\ \mathbb{R}^n \longrightarrow \mathbb{R}$

trouver x^* et f^* tels que

$$f^* = f(x^*) = \min_x f(x)$$

Hypothèses

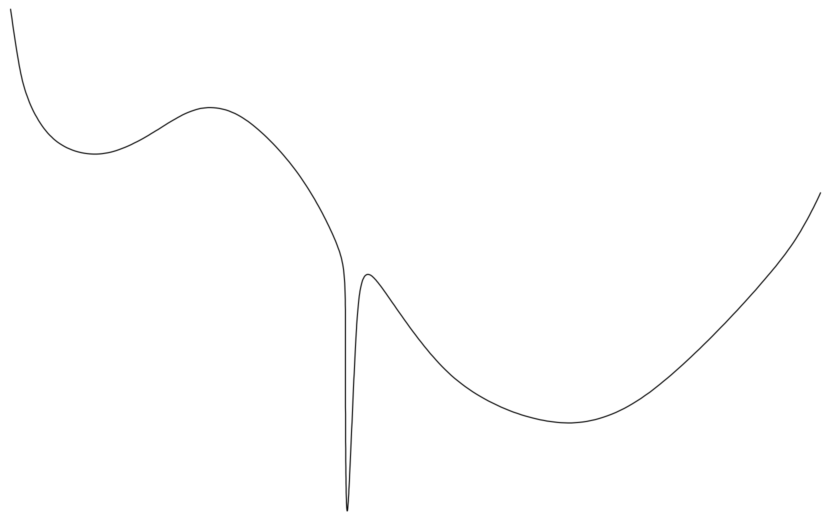
- recherche dans un pavé X_0
- $X^* \in$ intérieur (X_0), pas sur le bord
- f de classe $\mathcal{C}^2$

Pb. avec l'optimisation numérique en virgule flottante

(Dennis and Schnabel, 1983)

Méthodes numériques usuelles : optimum local
sauf si des propriétés fortes sont vérifiées par la fonction à optimiser (convexité. . .)

Recherches probabilistes : évitent l'optimum local le plus proche
toujours aucune garantie sur l'optimum trouvé : global ou non ?

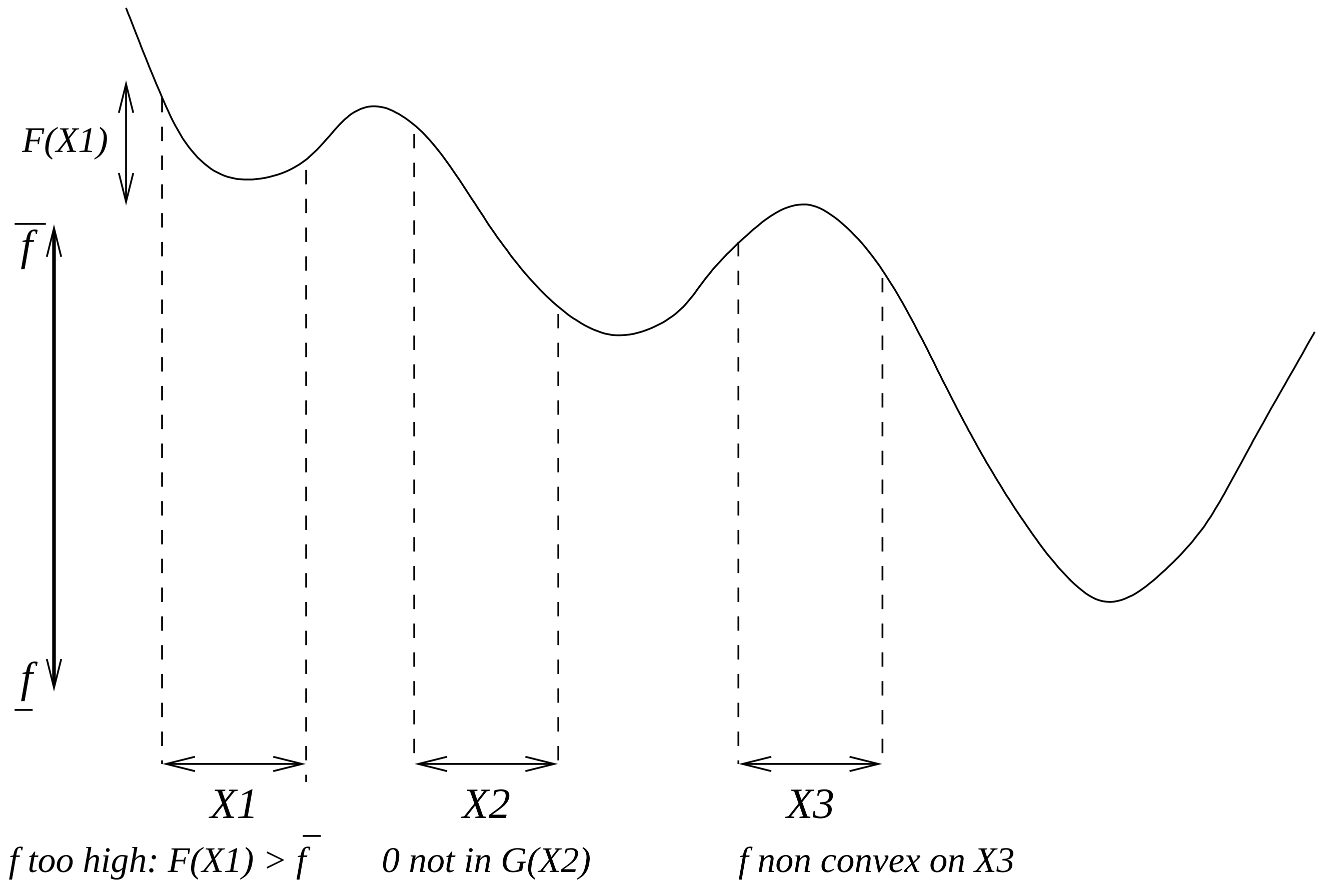


Optimiser une fonction continue méthode de base : Branch-and-Bound

$\bar{f}$ majorant courant de f^*
X pavé à examiner

tant qu'il reste un pavé X à examiner
    si $F(X) > \bar{f}$ alors
        rejeter X
    sinon
        couper X en 2 : X_1 et X_2
        examiner X_1 et X_2

Principe des différentes procédures



Optimiser une fonction continue
Branch-and-Prune

On réduit X avant de le couper en 2.

Utiliser le gradient

rejeter X si $\mathrm{Grad} f(X) \not\ni 0$

appliquer Newton sur $\mathrm{Grad} f$ (1 itération ou plusieurs)

Utiliser la Hessienne

en un minimum la fonction est convexe, $i.e.$ la Hessienne est SPD

OK si la Hessienne est à diagonale strictement positive.

Résoudre partiellement $F(X) \leq \bar{f}$

Taylor-Lagrange au 1er ordre : trouver $Y \subset X$ tel que

$F(Y) \subset f(x) + \mathrm{Grad} f(X)(Y - x) \leq \bar{f}$

résolution d'inéquations linéaires

Optimiser une fonction continue
algorithme de Hansen Hansen 1992

$\mathcal{L}$ = 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

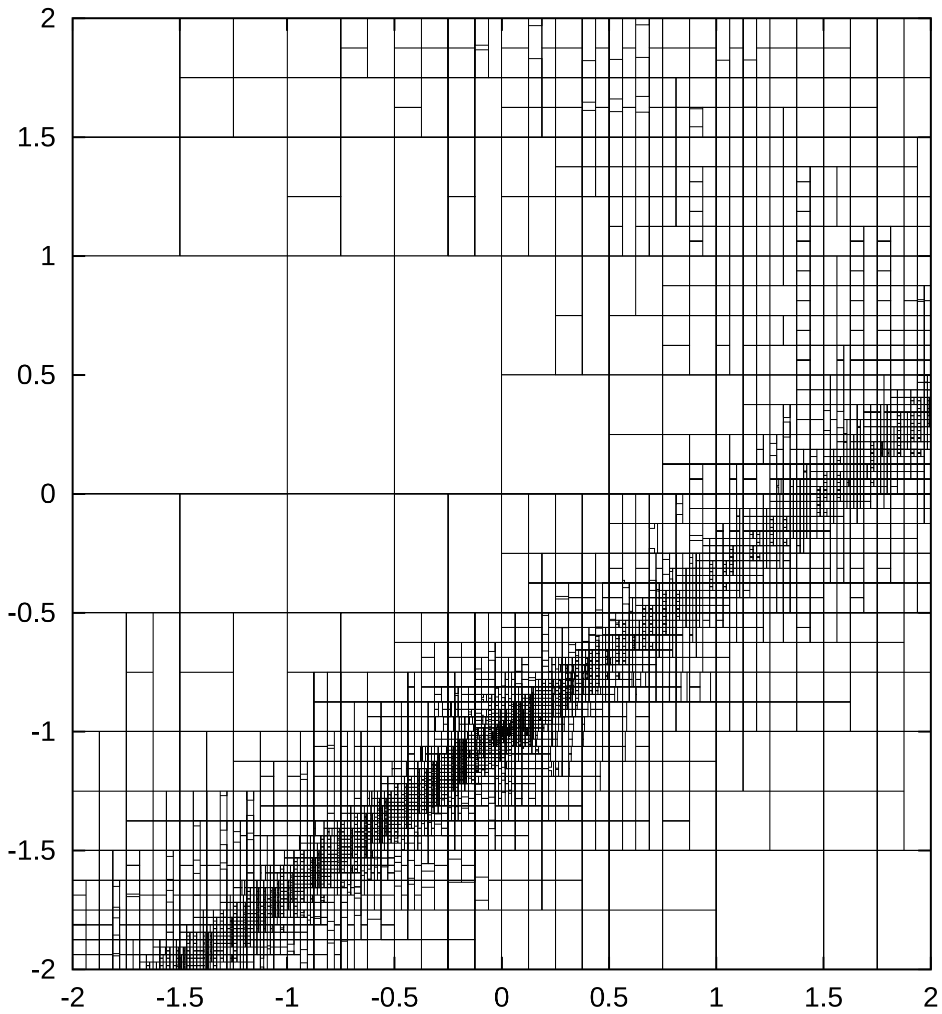
réduire X

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 $\mathcal{L}$


2
1.5
1
0.5
0
-0.5
-1
-1.5
-2
-2
-1.5
-1
-0.5
0
0.5
1
1.5
2

Plan de l'exposé

- **Arithmétique par intervalles**
  - principe
  - évaluation d'une expression
- **Algorithme de Hansen**
  - résolution de systèmes linéaires
  - Newton par intervalles
  - algorithme de Hansen
- **Parallélisation de l'algorithme de Hansen**
  - méthodologie
  - résultats expérimentaux
- **Arithmétique par intervalles en précision multiple**
  - motivation
  - Newton par intervalles en précision arbitraire

Parallelization of continuous verified global optimization

Nathalie Revol — Projet INRIA Arénaire, LIP, ENS-Lyon and Lab. ANO, Univ. Lille

Yves Denneulin — Projet INRIA Apache, ID-IMAG, Grenoble

Jean-François Méhaut — Lab. TRIVIA, Univ. Antilles-Guyane, Pointe-à-Pitre

Benoît Planquelle — LIFL, Lille

Parallel Processing and Applied Mathematics, 10-09-2001, Poland

Parallelization of Hansen's algorithm

(Berner 1996, Wiethoff 1997, Hu and Baker Kearfott 2001 + Grama and Kumar 1999)

Distinctive feature : Branch-and-Prune algorithm

Deserves parallelization since its complexity is exponential (worst cases)

Irregular algorithm : precedence graph is not predictible, thus scheduling is costly (Gautier, Roch and Villard 1995)
$\Rightarrow$ difficult to parallelize

Principle : begin the handling of a sub-box as soon as it is available
$\Rightarrow$ high level of speculation (Denneulin 1998)

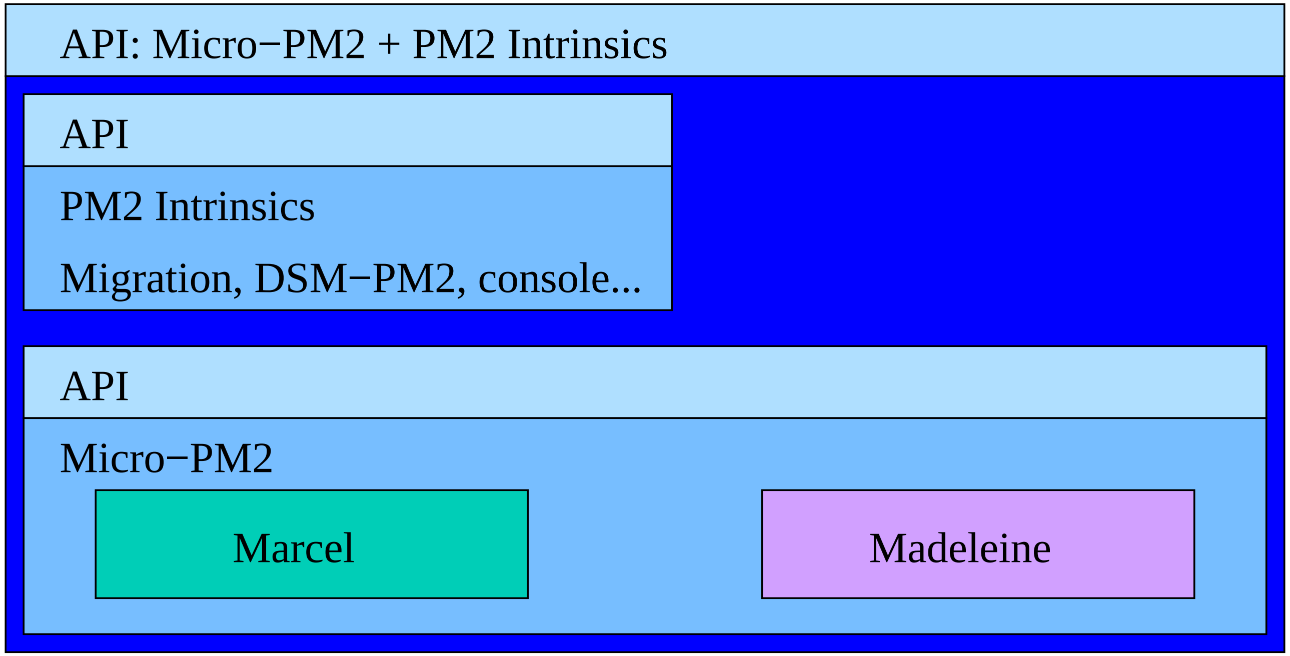
Parallel programming environment PM2

(Denneulin, Méhaut and Namyst 1997, Aumage, Antoniu, Bougé, Danjean and Namyst 2001)

Programming paradigm : multithread

Thread = lightweight computing task, several threads per processor.

PM2 structure



Scheduling and load balancing

(Blumofe and Leiserson 1998, Denneulin 1998. . .)

Distinct from the application : separate layer between PM2 and the application.

Strategy : customizable.

Priorities : the most promising sub-boxes have a high priority.

Requirement : not only scheduling but also balancing of the load among the processors $\Rightarrow$ priorities globally respected.

Problems

six-hump-camel-back

$X = [-2, 2]^2$

$$\min_{x \in X} 4x_1^2 - 2.1x_1^4 + 1/3x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$$

Goldstein-Price

$X = [-2, 2]^2$

$$\min_{x \in X} \left[1 + (x_1 + x_2 + 1)^2 \times (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)\right] \times \left[30 + (2x_1 - 3x_2)^2 \times (18\right.$$

Hartman 6

$X = [0, 1]^6$

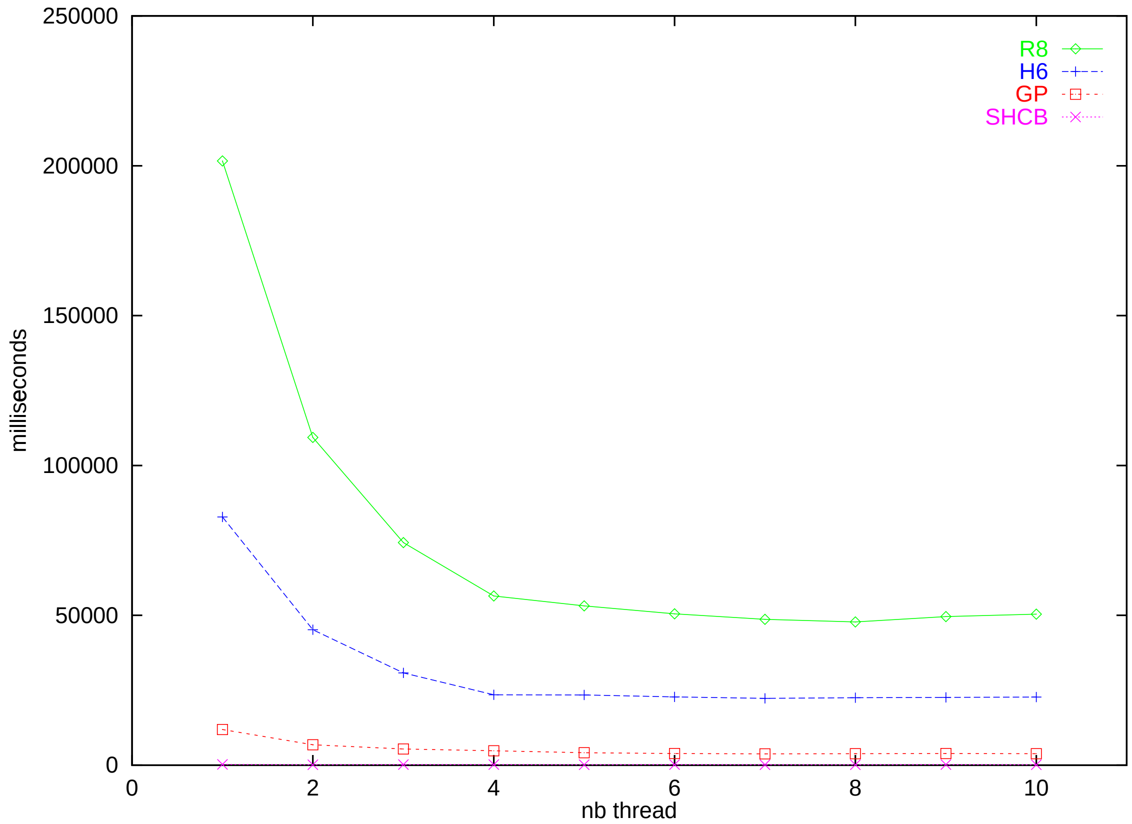
$$\min_{x \in X} - \sum_{i=1}^{4} c_i \exp\left(-\sum_{j=1}^{6} A_{ij}(x_j - P_{ij})^2\right)$$

Ratz 8

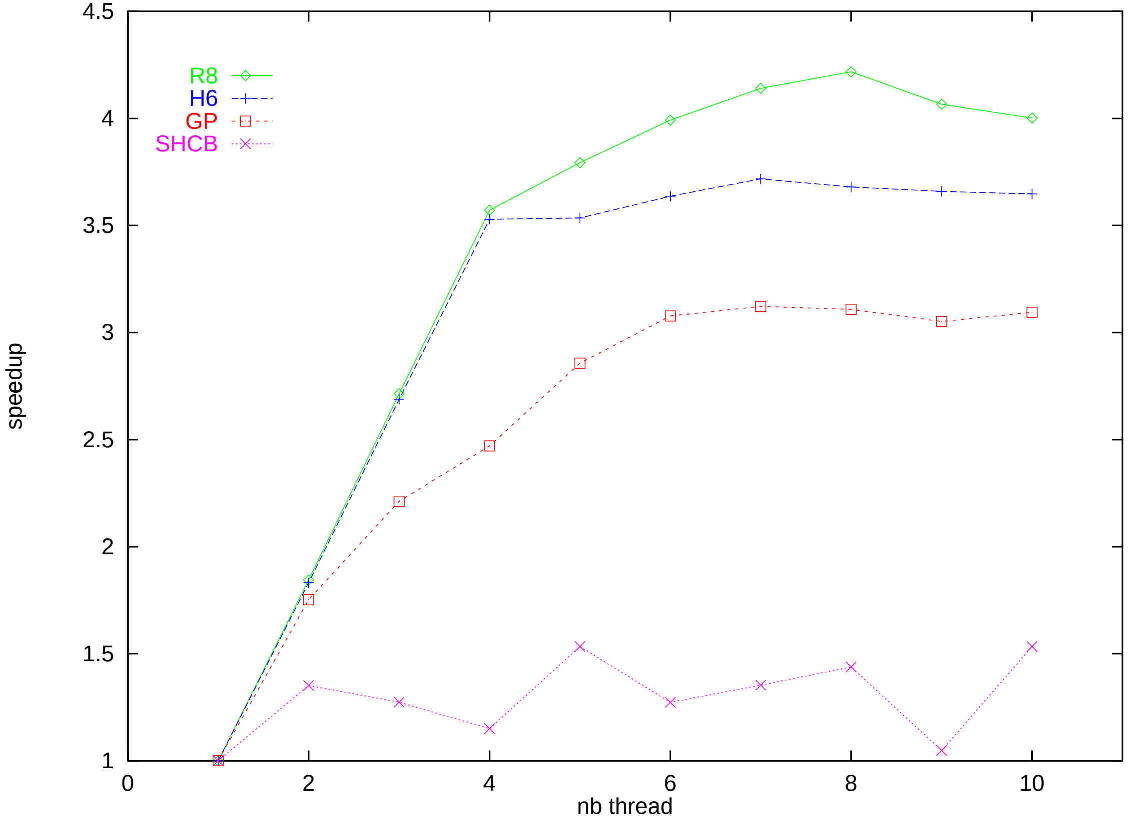
$X = [-10, 10]^9$

$$\min_{x \in X} \left[\sin^2(\pi \frac{x_1 + 3}{4}) + \sum_{i=1}^{8} \left(\frac{x_i - 1}{4}\right)^2 \times \left(1 + 10 \sin^2(\frac{x_{i+1} + 3}{4})\right)\right]$$

Execution times (Knüppel 1994, Csallner, Csendes and Markót 2001)



Speed-ups



Plan de l'exposé

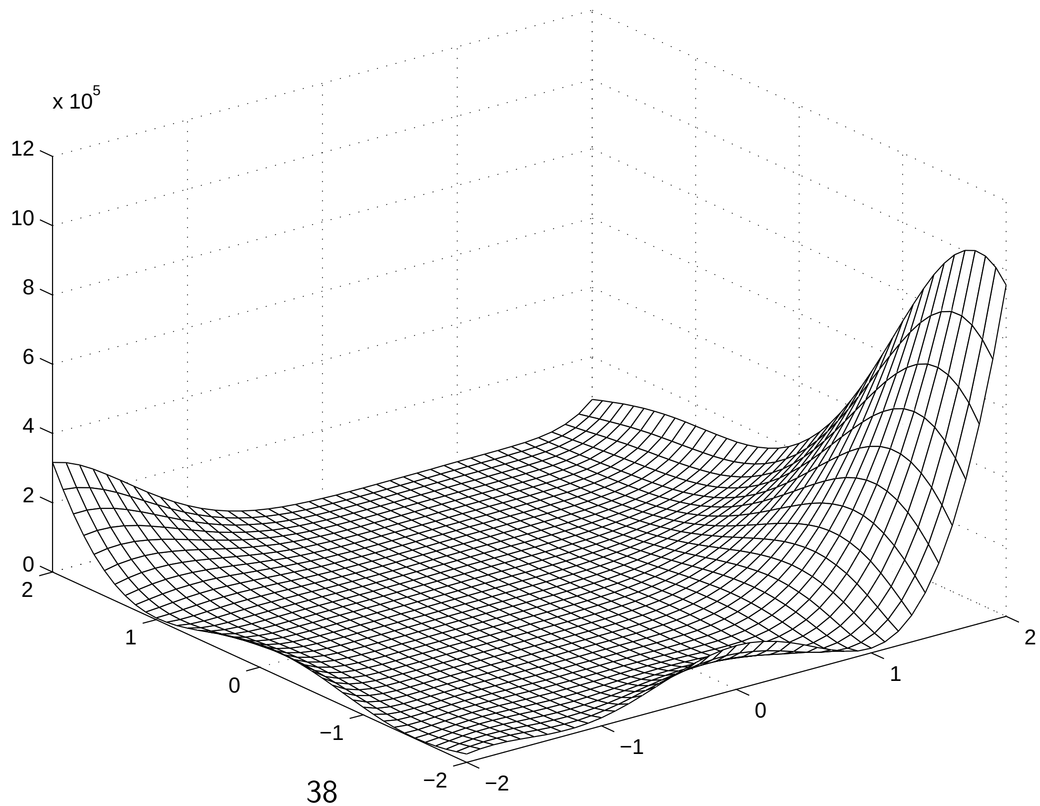
- **Arithmétique par intervalles**
  - principe
  - évaluation d'une expression
- **Algorithme de Hansen**
  - résolution de systèmes linéaires
  - Newton par intervalles
  - algorithme de Hansen
- **Parallélisation de l'algorithme de Hansen**
  - méthodologie
  - résultats expérimentaux
- **Arithmétique par intervalles en précision multiple**
  - motivation
  - Newton par intervalles en précision arbitraire

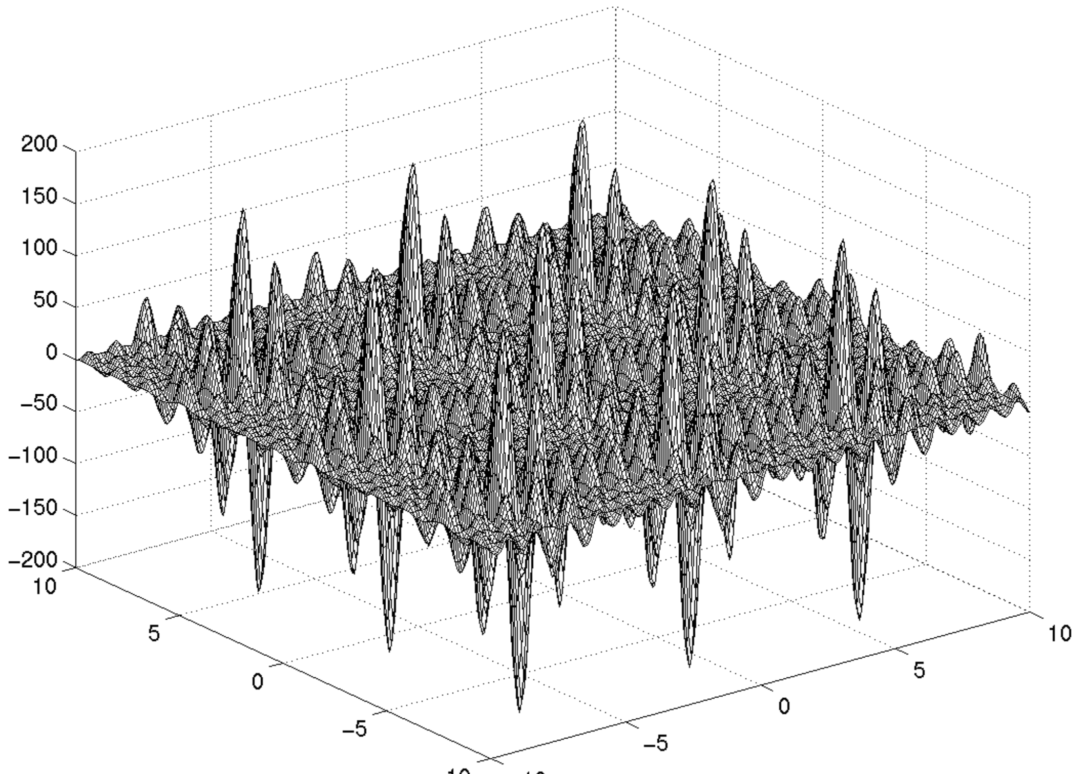
Arithmétique par intervalles : quand a-t-on besoin de plus de précision ?

Application visée : optimisation globale d'une fonction continue.

Fonction "vallée très plate" : sans MP, pavage de tout le fond de la vallée par de tout petits pavés.
Avec MP, on peut distinguer les points les plus élevés des autres et les rejeter.

Fonction "boîte à œufs" : avec beaucoup de minima locaux avec des valeurs de fonction très proches, indiscernables avec la précision machine.


x 10^5
12
10
8
6
4
2
0
2
1
0
−1
−2
−2
−1
0
1
2


200
150
100
50
0
−50
−100
−150
−200
10
5
0
−5
−10
−10
−5
0
5
10

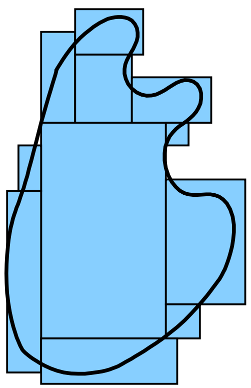
Arithmétique par intervalles : solution “naïve”

Couper les intervalles en deux :

si $X = X_1 \bigcup X_2$, alors $F(X_1) \bigcup F(X_2) \subset F(X)$:

Exemple : avec $F : x \mapsto x^2 - 2x + 1$,
$X = [-1, 3]$ et $X_1 = [-1, 1]$, $X_2 = [1, 3]$,
$F(X) = [-5, 12]$ contient strictement l'union de
$F(X_1) = [-1, 4]$ et de $F(X_2) = [-4, 8]$.

Utiliser des pavages :



Arithmétique par intervalles : solution $=$ Taylor

Si $f\ : I\!R^n \rightarrow I\!R$, de classe $\mathcal{C}^2$.

Développement de Taylor-Lagrange au 1er ordre :
$\forall x,\ \forall y,\ \exists \xi \in [x, y]\ /\ f(y) = f(x) + \text{grad} f(\xi).(y - x)$
ce qui s'interprète en intervalles :
$\forall X,\ \forall x \in X,\ \ f(X) \subset f(x) + \text{grad} f(X).(X - x)$.
Cette formule fournit un encadrement dont la largeur est ($\simeq$) proportionnelle à la largeur de X et surtout

$$\text{dist}(f(X), F(X)) \leq \mathcal{O}(w(X)^2) \text{ : approximation quadratique.}$$

Remarque : OK si X est petit, sinon une évaluation directe est meilleure.

Morale : pour pouvoir raffiner la précision des résultats, il faut pouvoir raffiner la précision sur les entrées. . . à volonté.

Arithmétique par intervalles et précision multiple

arithmétique multi-précision

$\oplus$ plus de précision (!)

$\ominus$ aucune garantie sur les résultats des calculs

arithmétique par intervalles multi-précision

$\oplus$ calcul certifié ou garanti

$\oplus$ et précis

MPFI

bibliothèque d'arithmétique par intervalles MP
co-développée avec F. Rouillier, en C, basée sur MPFR
disponible à `http://www.ens-lyon.fr/~nrevol`, rubrique `Softwares`

Arithmétique par intervalles et précision multiple

(Keiper 1993, Connell and Corless 1993, Geulig and Krämer 1999)

Certaines solutions ne font pas l'affaire. . .

Mathematica

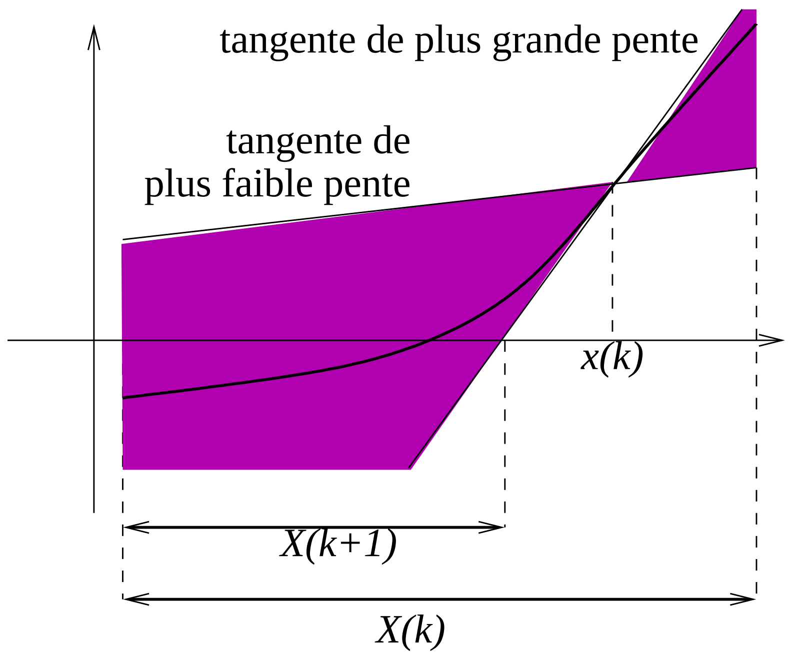
hypothèse : erreur ≤ 1 ulp sur les fonctions élémentaires
(ce qui n'est pas vérifié par l'implantation).

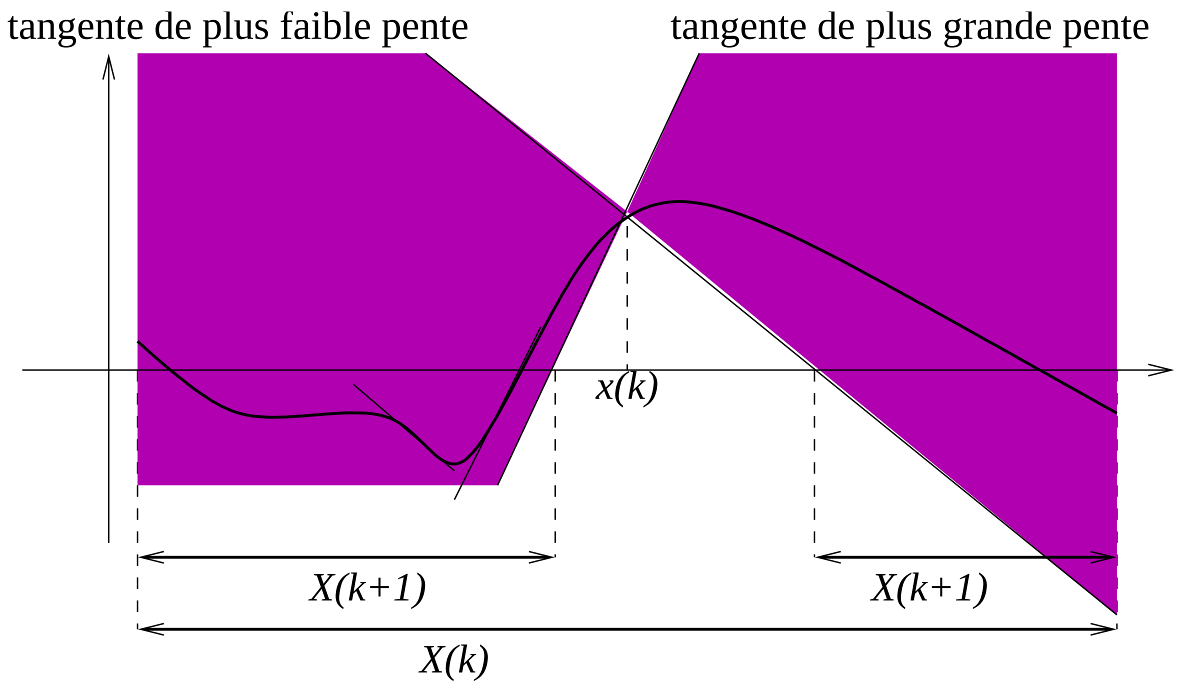
Maple

hypothèse : erreur ≤ 0.6 ulp sur les fonctions élémentaires
problèmes avec les arrondis dirigés :
$\bigtriangledown(1 - 9.10^{-3}) = 1.$

Algorithme de Newton par intervalles
principe d'une itération

(Hansen & Greenberg 83, Baker Kearfott 95-97, Mayer 95, van Hentenryck et al. 97)





Algorithme de Newton par intervalles MP

Input : F, F', X_0 // X_0 initial search interval

Initialization : $\mathcal{L} = \{X_0\}$, $\alpha = 0.75$ //any value in $]0.5, 1[$ is suitable

Loop : while $\mathcal{L} \neq \emptyset$

Suppress $(X, \mathcal{L})$

Increase the working precision if needed

$x :=$ mid(X)

$(X_1, X_2) := \left(x - \frac{F(\{x\})}{F'(X)}\right) \bigcap X$ // X_1 and X_2 can be empty

if $w(X_1) > \alpha w(X)$ or $w(X_2) > \alpha w(X)$ then $(X_1, X_2) :=$ bisect(X)

if $X_1 \neq \emptyset$ and $F(X_1) \ni 0$ then

if $w(X_1)/|\text{mid}(X_1)| \leq \varepsilon_X$ **and** $w(F(X_1)) \leq \varepsilon_Y$ then Insert X_1 in Res

else Insert X_1 in $\mathcal{L}$

same handling of X_2

Output : Res, a list of intervals that may contain the roots.

Algorithme de Newton par intervalles test d'arrêt et preuve de terminaison

(Baker Kearfott and Walster 2000)

Test d'arrêt : RRA and ARA

RRA (Relative Root Accuracy) : $w(X_1)/|x_1| \leq \varepsilon_X$
ε_X choisi par l'utilisateur

ARA (Absolute Residual Accuracy) : $w(F(X_1)) \leq \varepsilon_Y$
ε_Y choisi par l'utilisateur

Preuve de terminaison

$$\text{Un pas de l'algorithme } = \begin{cases} \text{1 pas de Newton} \\ \text{OU} \\ \text{1 pas de dichotomie} \end{cases}$$

Cela garantit que $w(X_{k+1}) \leq \alpha w(X_k)$
et donc le critère *Relative Root Accuracy* finira par être satisfait.

Under mild assumptions sur la fonction (expression courante pour désigner une expression lipschitzienne au sens intervalle), on a $w(F(X)) \leq K_F w(X)$, ce qui garantit que le critère *Absolute Residual Accuracy* finira par être satisfait également.

Encore vérifié si on prend en compte les erreurs d'arrondis, avec une précision suffisante.

Algorithme de Newton par intervalles adaptation automatique de la précision de calcul

Premier besoin : pouvoir couper l'intervalle d'entrée X en deux
$\Rightarrow$ augmenter la précision de calcul quand $w(X) \simeq 1$ "ulp".

Second besoin : pouvoir raffiner l'évaluation de la fonction $F(X)$
$\Rightarrow$ augmenter la précision de calcul quand $w(F(X)) \leq w(F(X_1))$.

Comment augmenter la précision : elle est doublée puisque le nombre de chiffres corrects est *grosso modo* doublé à chaque itération (Newton).

Algorithme de Newton par intervalles
la méthode idéale pour valider MPFI !

Méthode particulièrement adaptée au calcul par intervalles :
calculs ponctuels conservés tant que possible
itération contractante.

Pourquoi peut-on éviter de recommencer tout le calcul ?
à chaque fois que la précision est augmentée : la méthode de Newton est auto-correctrice.

Algorithme de Newton par intervalles
polynômes de Chebychev

Polynômes de Chebychev : $C_n(\cos(\theta)) = \cos(n\theta)$.
Difficiles à évaluer avec une bonne précision : même s'ils prennent leurs valeurs dans $[-1, 1]$, leurs coefficients sont très grands en valeur absolue.

Résultats : racines calculées très précisément pour des degrés jusqu'à 40, avec preuve d'existence de toutes les racines et d'unicité de beaucoup.

Algorithme de Newton par intervalles polynôme de Wilkinson $\prod_{i=1}^{20}(X-i)$.

Avec assez de précision pour pouvoir représenter exactement les coefficients : racines trouvées avec une précision de 5.10^{-2} et existence prouvée (mais pas unicité).
Beaucoup d'intervalles non éliminés : $[0.96, 1.02]$ et $[1.62, 20.984]$.
Résultats similaires (mais temps de calcul doublé) avec pour intervalles initiaux $[-10, 40]$ et $[-100, 400]$.

Avec assez de précision et une pertubation $[-2^{-19}, 2^{-19}]$ sur le coefficient de X^{19} : racines (avec preuve d'existence mais pas d'unicité) : 1 ± 4.10^{-2}, 2 ± 5.10^{-2}, 3 ± 4.10^{-2}, 4 ± 4.10^{-2}, 5 ± 4.10^{-2}, 6 ± 5.10^{-2}, 7 ± 6.10^{-2} et $[7.91, 22.11]$.
Beaucoup d'intervalles non éliminés : $[0.96, 22.64]$.

Polynôme

Langlois & Revol 2001

Polynôme

$$P(x) = 1.47x^3 + 1.19x^2 - 1.83x + 0.45$$

qui est la forme développée de

$$\frac{3 \times 7 \times 7}{100} \left(x + \frac{5}{3} \right) \times \left(x - \frac{3}{7} \right)^2 .$$

But : étudier le comportement de différentes versions de Newton autour de la racine double 3/7.

Racine mal conditionnée

Sensibilité aux perturbations d'une racine multiple

Si x^* est racine multiple (de multiplicité m) du polynôme

$$p(x) = a_n x^n + \cdots + a_1 x + a_0,$$

alors une perturbation δ du coefficient a_i perturbe la racine x^* en $x^*(\delta)$:

$$x^*(\delta) - x^* = \mathcal{O}(\delta^{1/m}) \simeq \delta^{1/m} \left(- \frac{m!\, a_i {x^*}^i}{p^{(m)}(x^*)} \right)^{1/m}.$$

Meilleure borne possible si la méthode de calcul des racines évalue p.
Autrement dit, en précision fixée $\geq \delta$, comme un polynôme et son perturbé sont équivalents, x^* et $x^*(\delta)$ sont indiscernables.

Interval Newton algorithm experiments : double root

Pol : $7^2 \times 3/100 \times (x - 5/3) \times (x - 3/7)^2$.
In IEEE single prec. (24 bits). Studied root : $3/7 \simeq 0.428\,571\,428$.

Classical Newton with IEEE arithmetic : $x^* = 0.428\,5\underline{61\,18}$

Digits validated by CADNA : $x^* = 0.428\,\underline{7}$

Root corrected by CENA : $x^*_- = 0.428\,\underline{495\,94}$ and $x^*_+ = 0.428\,\underline{646\,92}$

Interval Newton in fixed prec. : $\simeq -5/3$, $[0.428\,\underline{476\,72}, 0.428\,\underline{528\,19}]$, $[0.428\,6\underline{17\,47}, 0.428\,6\underline{76\,46}]$

Interval Newton in MP : existence and uniqueness of 3 single roots.

Interval Newton in MP and outward rounded (interval) coefficients : no answer.

Expérimentations : 1e série

En IEEE simple prec. (24 bits).
Racine double étudiée : $3/7 \simeq 0.428\,571\,428$.
Erreur théorique prédite : $(2^{-24})^{1/2} \simeq (5.96 \times 10^{-8})^{1/2} \simeq 2.4 \times 10^{-4}$.

Newton en IEEE : $x^* = 0.428\,5\underline{61\,18}$
soit 4 chiffres décimaux corrects : OK.

Chiffres validés par CADNA : $x^* = 0.428\,\underline{7}$
soit 4 chiffres décimaux corrects : OK.

Expérimentations : 1e série

Chiffres validés par CADNA : $x^* = 0.428\,\underline{7}$

Racine avec correction par CENA : (des évaluations du pol.)

en partant de $0.4280\,0000$: $x^*_- = 0.428\,\underline{495\,94}$

en partant de $0.4289\,9999$: $x^*_+ = 0.428\,\underline{646\,92}$

Newton par intervalles en préc. fixée :

dans $[-2, 2]$, 3 racines prouvées (existence)

$\simeq -5/3$, $[0.428\,\underline{476\,72}, 0.428\,\underline{528\,19}]$, $[0.428\,6\underline{17\,47}, 0.428\,6\underline{76\,46}]$

Newton par intervalles en MP :

existence et unicité de 3 racines simples. (Confirmé à l'aide de Maple).

Explication :

conversion décimale $\Rightarrow$ binaire des coeff. du polynôme équivalent à une perturbation $< (2^{-24})^{1/2} \simeq (5.96 \times 10^{-8})^{1/2}$.

Expérimentations : 2e série

En IEEE simple prec. (24 bits),
avec arrondis "corrects" des coeff. du polynôme.
Racine double étudiée : $3/7 \simeq 0.428\,571\,428$.

Chiffres validés par CADNA : $x^* = 0.428\,\underline{7}$
soit 4 chiffres décimaux corrects : OK.

Newton par intervalles en préc. quelconque
et coeff. arrondis vers l'extérieur : $\simeq -5/3$ et $[0.428\underline{3\,8624}, 0.428\underline{7\,4685}]$.
Explication : autour de 3/7
avec des coeff. arrondis vers $+\infty$, 0 racine réelle
avec des coeff. arrondis au plus près, 2 racines simples
polynôme initial : 1 racine double
Pas de conclusion commune possible.

Conclusion

• résultats satisfaisants avec l'arithmétique par intervalles, si les algorithmes sont correctement conçus ;

• MP et arithmétique par intervalles sont coûteux. . . pas trop ! (cf. Berz)

Travail en cours et à venir

• **Tester et améliorer la bibliothèque :** MPFI and MPFI++

• **Développer des algorithmes numériques** utilisant l'arithmétique par intervalles MP : algèbre linéaire $\Rightarrow$ Newton à plusieurs variables $\Rightarrow$ optimisation globale

• **Applications : automatique et robotique** estimation de paramètres et contrôle robuste