Reliable and accurate solutions of linear and nonlinear systems

Nathalie Revol

Projet INRIA Arénaire, LIP, ENS-Lyon
and Lab. ANO, Univ. Lille
Nathalie.Revol@ens-lyon.fr

SIAM Conference on Optimization

Toronto, Ontario, Canada, 20-22 May, 2002

Reliable computing: Interval Arithmetic

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

Numbers are replaced by intervals.

Ex: π replaced by $[3.14159, 3.14160]$

Advantages: every result is guaranteed.

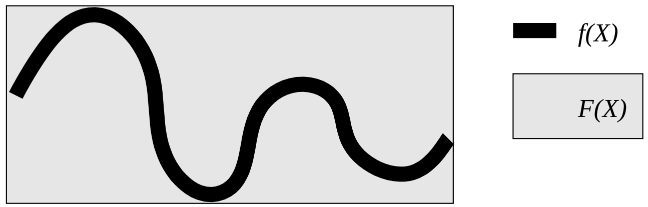
Data known up to measurement errors are representable.

Drawbacks: overestimation of the results.

Variable dependency

$$
\begin{aligned}
I * I &= \{x * y \,;\, x \in I,\ y \in I\} \\
&\neq I^2 = \{x^2 \,;\, x \in I\}
\end{aligned}
$$

Wrapping effect



Hansen's algorithm for global optimization

$\mathcal{L}$ = list of boxes to process $:= \{X_0\}$

while $\mathcal{L} \neq \emptyset$ loop

  suppress X from $\mathcal{L}$

  reject X ?

    yes if $F(X) > \bar{f}$

    yes if Grad$F(X) \not\ni 0$

    yes if $HF(X)$ has its diag. non > 0

  reduce X

    Newton applied with the gradient

    solve $Y \subset X$ such that $F(Y) \leq \bar{f}$

  bisect Y into Y_1 and Y_2 if Y is not a result

    insert Y_1 and Y_2 in $\mathcal{L}$

When is extra precision needed?

Target application: global optimization of a continuous function.
Difficulties with "very flat valley" function and with "egg-box" function.

Solution: bisect intervals
if $X = X_1 \bigcup X_2$, then $F(X_1) \bigcup F(X_2) \subset F(X)$
and the left interval is usually tighter than the right one.
Theorem: Under mild assumptions on f and with a direct evaluation,

$$\text{dist}(f(X), F(X)) \leq \mathcal{O}(w(X))$$

Moral: in order to refine the accuracy on the results, the precision on the inputs must be refined. . . as much as required.

Accurate computing: arbitrary precision

(Knuth, Schönhage & Strassen, Brent 1976 . . .)

Representable numbers are floating-point numbers with arbitrary precision, chosen by the user.

MPFI: Multiple Precision Floating-point Interval arithmetic library.

- **what:** a C library, based upon MPFR (**www.mpfr.org**), a multiple precision floating-point library, providing directed exact rounding, based upon GMP (**www.swox.se/gmp**) for portability and efficiency.
- **why:** Maple or Mathematica packages are incorrect.
- **who:** N. Revol and F. Rouillier.
- **how:** sources and documentation available from **www.ens-lyon.fr/~nrevol**

Reliable and accurate computing: transforming usual algorihms

Usual approach: (cf. **iRRAM**)

if accuracy of result is not sufficient then
    increase computing precision
    restart whole computation

Goal: avoid restarting whole computation.

if accuracy of result is not sufficient then
    increase computing precision
    go on

Stopping criterion: arbitrary accuracy can be required both on results and on residuals.

Agenda

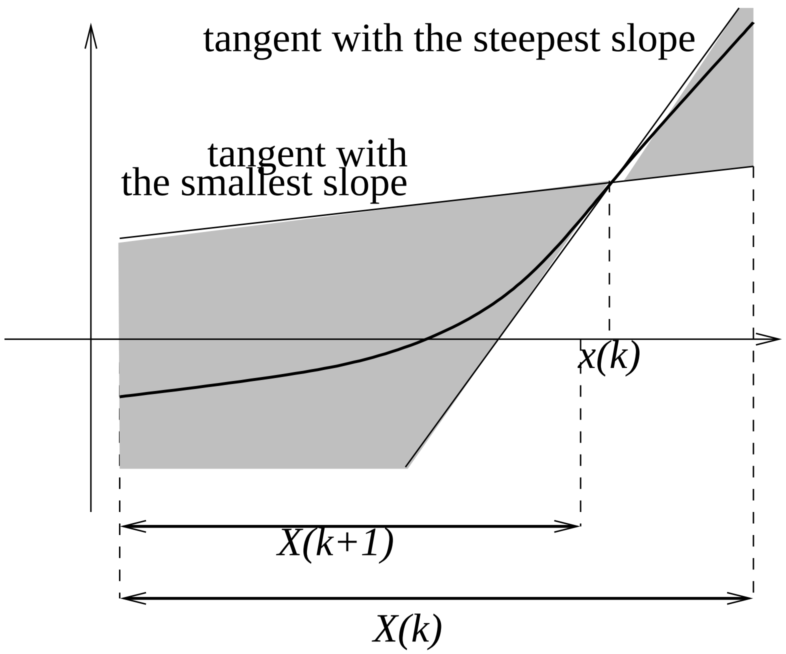
- **Reliable and accurate computations**
  - reliability: interval arithmetic
  - accuracy: arbitrary precision
  - MPFI: multiple precision interval arithmetic library

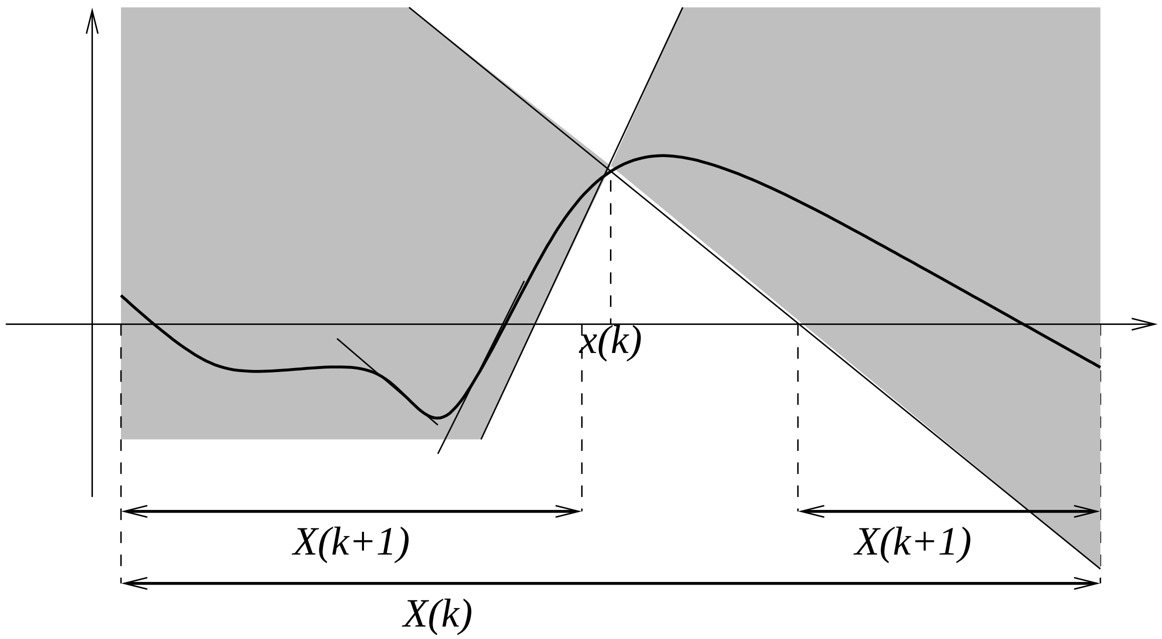
- **Solving nonlinear equations**
  - interval Newton algorithm
  - some properties
  - experimental results

- **Solving linear systems**
  - Hansen & Sengupta's algorithm
  - adaptation to multiple precision

Interval Newton algorithm
principle of one iteration

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





The result will be a list of intervals.

Interval Newton algorithm

Input: F, F', X_0 $\qquad$ // X_0 initial search interval

Initialization: $\mathcal{L} = \{X_0\}$, $\alpha = 0.75$ $\qquad$ //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)

$(X_1, X_2) := \left(x - \frac{F(\{x\})}{F'(X)}\right) \bigcap X$ $\qquad$ // 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.

Interval Newton algorithm stopping criterion and termination proof

(Baker Kearfott and Walster 2000)

Stopping criterion

Relative Root Accuracy: $w(X_1)/|x_1| \leq \varepsilon_X$

Absolute Residual Accuracy: $w(F(X_1)) \leq \varepsilon_Y$

(ε_X and ε_Y given by the user)

Termination proof

$$\text{One step of the algorithm} = \begin{cases} \text{1 step of Newton} \\ \text{OR} \\ \text{1 step of dichotomy} \end{cases}$$

This ensures that $w(X_{k+1}) \leq \alpha w(X_k)$.

Interval Newton algorithm
automatic adaptation of the working precision

First need: being able to bisect the current interval X
$\Rightarrow$ increase the working precision when $w(X) = 1$ "ulp".

Second need: being able to refine the function evaluation $F(X)$
$\Rightarrow$ increase the working precision when $w(F(X)) \leq w(F(X_1))$.

How to increase the working precision : many possible choices, for us it is doubled since the number of correct digits is roughly doubled at each iteration (Newton).

Interval Newton algorithm experiments: Chebychev polynomials

Chebychev polynomials: $C_n(\cos(\theta)) = \cos(n\theta)$.
They are difficult to evaluate with a good precision: even if their values belong to $[-1, 1]$, their coefficients are very large.

Results: very precise roots for degrees up to 40, with proof of existence and uniqueness.

Interval Newton algorithm experiments: Wilkinson polynomial $\prod_{i=1}^{20}(X-i)$.

With enough precision to be able to store exactly the coefficients: roots found with a precision 5.10^{-2} and with a proof of existence (but not uniqueness).
A lot of intervals are not eliminated: $[0.96, 1.02]$ et $[1.62, 20.984]$.
Similar behaviour (but the computing time is twice as much) with the starting intervals $[-10, 40]$ and $[-100, 400]$.

With enough precision and a perturbation $\pm 2^{-19}$ on the coefficient of X^{19}: roots (with proof of existence but not uniqueness):
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} and $[7.91, 22.11]$.
A lot of intervals are not eliminated: $[0.96, 22.64]$.

Agenda

- **Reliable and accurate computations**
  - reliability: interval arithmetic
  - accuracy: arbitrary precision
  - MPFI: multiple precision interval arithmetic library
- **Solving nonlinear equations**
  - interval Newton algorithm
  - some properties
  - experimental results
- **Solving linear systems**
  - Hansen & Sengupta's algorithm
  - adaptation to multiple precision

Solving linear systems
Hansen & Sengupta's algorithm

Linear system: $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ with $\boldsymbol{A}$ and $\boldsymbol{b}$ given.
Problem: compute an enclosure of
Hull $(\Sigma_{\exists\exists}(\boldsymbol{A}, \boldsymbol{b})) =$ Hull $(\{\boldsymbol{x} \ : \ \exists A \in \boldsymbol{A}, \ \exists b \in \boldsymbol{b}, Ax = b\})$.

Hansen & Sengupta's algorithm
compute C an approximation of mid$(\boldsymbol{A})^{-1}$
apply Gauss-Seidel to $C\boldsymbol{A}\boldsymbol{x} = C\boldsymbol{b}$ until convergence.

Idea:
$C\boldsymbol{A}$ contains the identity matrix, the iteration matrix has a spectral radius close to 1 (and even < 1?) and this iteration is a contraction.

Hansen & Sengupta's algorithm
automatic adaptation of the working precision

First need: being able to improve the current interval $\boldsymbol{x}$
$\Rightarrow$ increase the working precision when $w(\boldsymbol{x}) = 1$ "ulp".

Second need: being able to refine the Gauss-Seidel iteration evaluation.
Notation: $[\boldsymbol{x_1}, \boldsymbol{x_2}] =$ bisect $(\boldsymbol{x})$
one iteration applied to $\boldsymbol{x}$ (resp. to $\boldsymbol{x_1}$, resp. to $\boldsymbol{x_2}$) gives $\boldsymbol{y}$ (resp. $\boldsymbol{y_1}$, (resp. $\boldsymbol{y_2}$)
increase the working precision if $w(\boldsymbol{y_1}) \geq w(\boldsymbol{y})$ or $w(\boldsymbol{y_2}) \geq w(\boldsymbol{y})$.

How to increase the working precision : many possible choices, for us it is doubled.

Conclusion and future work

Conclusion

• **Reliable and accurate computing is possible,** thanks to arbitrary precision interval arithmetic.

• **Algorithms need to be adapted:** automatic adaptation of the precision, more stringent stopping criteria.

Future work

• **Testing and improving the library:** MPFI and MPFI++ (interface à la Profil/BIAS).

• **Development of numerical algorithms** with MP interval arithmetic: multivariate Newton $\Rightarrow$ global optim. $\Rightarrow$ constrained global optim.

• **Applications: automatics and robotics**
parameters estimation and robust control.