MPFI: a library for arbitrary precision interval arithmetic

Nathalie Revol Arénaire (CNRS/ENSL/INRIA), LIP, ENS-Lyon and Lab. ANO, Univ. Lille
Nathalie.Revol@ens-lyon.fr

Fabrice Rouillier Spaces, CNRS, LIP6, Univ. Paris 6, LORIA-INRIA
Fabrice.Rouillier@loria.fr

SCAN 2002, Paris, France, 24-27 September

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 $[3.14159, 3.14160]$

Computations: can be done with interval operands.

Advantages:

- Every result is guaranteed (including rounding errors).
- Data known up to measurement errors are representable.
- Global information can be reached ("computing in the large").

Drawbacks: overestimation of the results.

Hansen's algorithm for global optimization

Hansen 1992

$\mathcal{L}$ = list of boxes to process $:= \{X_0\}$ $\quad$ F: function to minimize

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 $\mathcal{L}$

Solving linear systems
preconditioned Gauss-Seidel algorithm

Hansen & Sengupta 1981

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 $(\{x \, : \, \exists A \in \boldsymbol{A}, \, \exists b \in \boldsymbol{b}, Ax = b\})$.

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 (hope):

$C\boldsymbol{A}$ contains the identity matrix, is diagonally dominant, thus the iteration matrix has a spectral radius < 1 and this iteration is a **contraction**.

Agenda

- **Introduction: when is more precision wanted?**
  - global optimization (Hansen's algorithm)
  - linear system solving (Hansen & Sengupta's algorithm)

- **MPFI: a library for arbitrary precision interval arithmetic**
  - general view of MPFI
  - MPFI in details
  - other libraries

- **Applications**
  - robot kinematics
  - isolation of real roots of a polynomial
  - approximation of zeros of a function

MPFI Multiple Precision Floating-point Interval arithmetic library
general view

- **what:** C library for arbitrary precision interval arithmetic
- **why:** to compute reliably and accurately
- **where:** freely available, including source code and documentation (last release: April 2002)
- **who:** N. Revol and F. Rouillier
- **how:** based on MPFR for arbitrary precision floating-point arithmetic (based on GMP: efficient and portable)

MPFI: in details

- **Requirements:** directed roundings are required, exact is better. Even for elementary functions! (provided by MPFR)

- **Interval:** connected closed subset of $I\!R$

- **Operation:** $op(\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n) \supset \{op(x_1, \ldots, x_n) \ : \ x_i \in \boldsymbol{X}_i\}$

- **Undefined operation:** NaN if $op(x_1, \ldots, x_n)$ is undefined (no *a priori* intersection with the domain)

- **Precision:** intervals are represented by their endpoints, each endpoint carries its own precision.

MPFI: functionalities

- arbitrary precision interval data type
- arithmetic operations: $+$, $-$, $\times$, $/$, $\sqrt{}$
- constants: π, $\log 2$, Euler constant
- elementary functions: exp, log, atan, cosh, asinh. . .
- IEEE-754 special values: ∞, signed zeros, NaN
- conversions to and from integer, double, exact naturals, exact integers, rational, “reals” (MPFR numbers)
- Input/Output (to be improved)

Performances

Development version

Gaussian elimination on a random M-matrix (300×300)

ARITHM PREC	double 53 bits	fi_lib 53 bits	MPFI 53 bits	MPFI 63 bits	MPFI 127 bits	MPFI 255 bits
TIME (s) RATIO	0.42 0.10	4.39 1.00	21.75 4.95	22.35 5.09	24.11 5.49	34.09 7.76

ARITHM PREC			MPFI 511 bits	MPFI 1023 bits	MPFI 2047 bits	MPFI 4095 bits
TIME (s) RATIO			51.61 11.75	112.30 25.58	285.57 65.05	817.05 186.11

(Athlon 1GHz)

Other libraries

Most of the interval arithmetic libraries are based on machine floating-point precision (Profil/BIAS, Intlib, Sun Forte. . .)

Some libraries offer multiple precision:

- **Maple, Mathematica:** floating-point arithmetic and roundings bugged
- **range:** "the interval is only crudely represented" (Aberth)
- **XSC:** limited use (in long accumulators)
- **IntLab:** limited set of operations and functions

Agenda

- **Introduction: when is more precision wanted?**
  - global optimization (Hansen's algorithm)
  - linear system solving (Hansen & Sengupta's algorithm)
- **MPFI: a library for arbitrary precision interval arithmetic**
  - general view of MPFI
  - MPFI in details
  - other libraries
- **Applications**
  - robot kinematics
  - isolation of real roots of a polynomial
  - approximation of zeros of a function

Parallel robot kinematics

Formulation: algebraic system depending on 7 to 12 variables.

Goal: computing all the real solutions (up to 40).

Usual numerical computational strategies failed on this example using hardware doubles.

Solution obtained by

- a few exact methods
- MPFI using 128 bits of precision

Remark: some preliminary computations were performed using exact computations (Gröbner bases).

Isolation of polynomial real roots

Rouillier and Zimmermann 2002

Descartes' rule of sign

Let $P = \sum_{i=0}^{d} p_i x^i$ be a polynomial,
$V(P) =$ number of sign changes in $(p_o, \ldots, p_d)$
and $pos(P) =$ number of > 0 real roots of P, counted with mult.
Then $pos(P) \leq V(P)$ and $V(P) - pos(P)$ is even.

Isolation of polynomial real roots algorithm

"Uspensky's" algo.

Assumption: P is a square-free polynomial with roots in $[0, 1]$.
Transform P into $Q = \sum_i q_i x^i$ such that $V(Q) = 0$ or 1
$i.e.$ P has 0 or 1 root on a given interval:
a root of P is **isolated**.

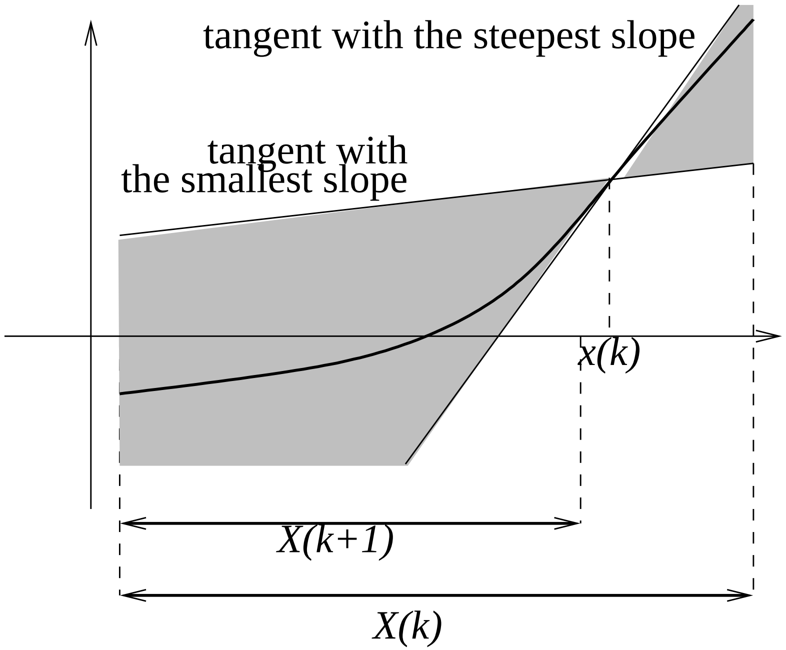
Modified algorithm using MPFI

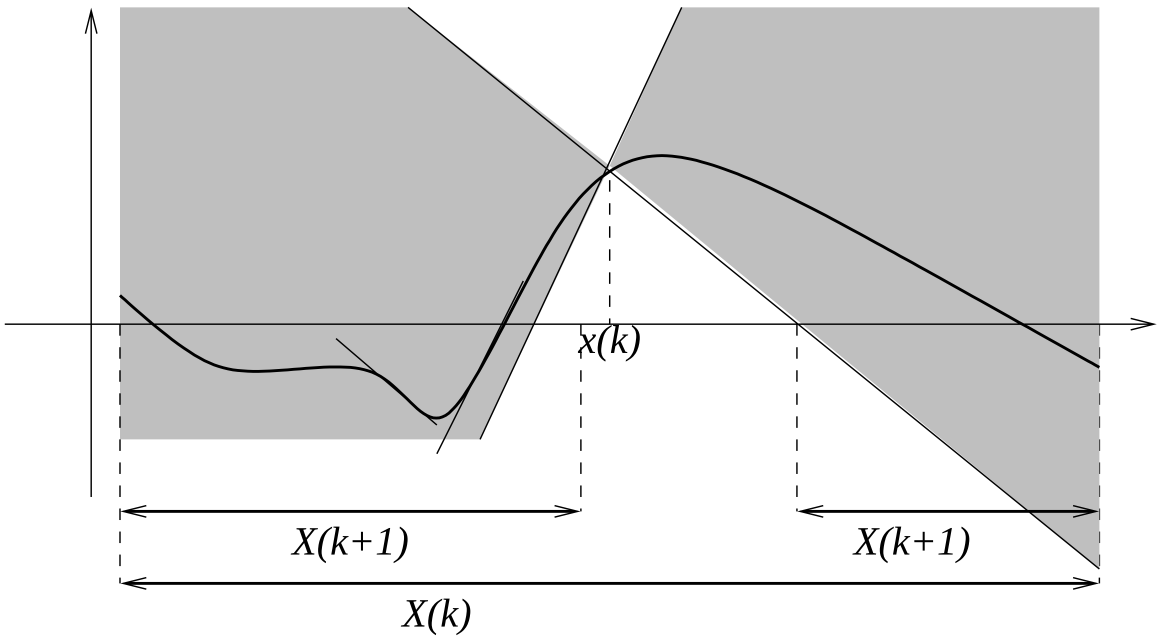
replace exact computations in $Transform\ P\ into\ Q$
by MPFI computations:
q_i is now an interval.
If $q_i \ni 0$ then stop interval computation, increase MPFI precision or go on with exact computation.

N. Revol & F. Rouillier

Approximation of zeros: 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 // 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.

Interval Newton algorithm: experiments

Chebychev polynomials: $C_n(\cos(\theta)) = \cos(n\theta)$.

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

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 a proof of existence (but not uniqueness). A lot of intervals are not eliminated: $[0.96, 1.02]$ et $[1.62, 20.984]$.

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: $[0.96, 22.64]$.

Conclusion

MPFI:

- library for arbitrary precision interval computation
- at a reasonable cost.

Applications

- guaranteed results with arbitrary accuracy
- algorithms: increase precision when needed and continue

Coming soon

- MPFI++
- linear algebra
- more algorithms.