Computer-Assisted Proofs
Introduction to Interval Arithmetic

Cours de recherche master informatique

25 November 2025

A brief introduction

Interval arithmetic: replace numbers by intervals and compute.

Fundamental theorem of interval arithmetic:
(or “Thou shalt not lie”):
the exact result (number or set) is contained in the computed interval.

No result is lost, the computed interval is guaranteed to contain every possible result.

A brief introduction

Interval arithmetic: replace numbers by intervals and compute.
Initially: introduced to take into account roundoff errors (Moore 1966)
and also uncertainties (on the physical data. . .).
Later: computations "in the large", computations with sets.

Interval analysis: develop algorithms for **reliable (or verified, or guaranteed, or certified) computing**,
that are suited for interval arithmetic,
i.e. different from the algorithms from classical numerical analysis.

A brief introduction: examples of applications

- control the roundoff errors, cf. computational geometry
- solve several problems with verified solutions: linear and nonlinear systems of equations and inequalities, constraints satisfaction, (non/convex, un/constrained) global optimization, integrate ODEs e.g. particules trajectories. . .
- mathematical proofs: cf. Hales' proof of Kepler's conjecture or Tucker's proof that Lorenz system has a strange attractor.

Cf. `http://www.cs.utep.edu/interval-comp/`

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

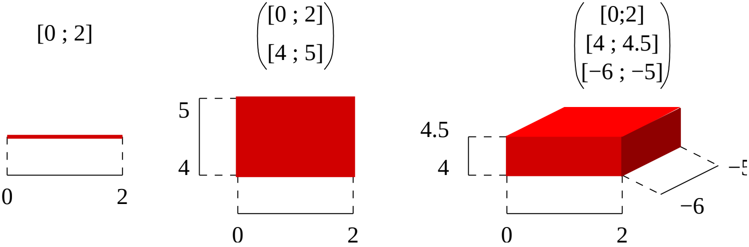
Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Definitions: intervals

Objects:

- intervals of real numbers = closed connected sets of $\mathbb{R}$
  - interval for π: $[3.14159, 3.14160]$
  - data d measured with an absolute error less than $\pm\varepsilon$: $[d-\varepsilon, d+\varepsilon]$
- interval vector: components = intervals; also called *box*



- interval matrix: components = intervals.

Definitions: operations

$$\boldsymbol{x} \diamond \boldsymbol{y} = \textbf{Hull}\{x \diamond y \ : \ x \in \boldsymbol{x}, y \in \boldsymbol{y}\}$$

Arithmetic and algebraic operations: use the monotonicity

$$\begin{aligned}
[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \\
[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \\
[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] &= [\min(\underline{x} \times \underline{y}, \underline{x} \times \overline{y}, \overline{x} \times \underline{y}, \overline{x} \times \overline{y}), \max(\text{ibid.})] \\
[\underline{x}, \overline{x}]^2 &= [\min(\underline{x}^2, \overline{x}^2), \max(\underline{x}^2, \overline{x}^2)] \ \text{ if } 0 \notin [\underline{x}, \overline{x}] \\
& [0, \max(\underline{x}^2, \overline{x}^2)] \ \text{ otherwise} \\
1/ [\underline{y}, \overline{y}] &= [\min(1/\underline{y}, 1/\overline{y}), \max(1/\underline{y}, 1/\overline{y})] \ \text{ if } 0 \notin [\underline{y}, \overline{y}] \\
[\underline{x}, \overline{x}] / [\underline{y}, \overline{y}] &= [\underline{x}, \overline{x}] \times (1/ [\underline{y}, \overline{y}]) \text{ if } 0 \notin [\underline{y}, \overline{y}] \\
\sqrt{[\underline{x}, \overline{x}]} &= [\sqrt{\underline{x}}, \sqrt{\overline{x}}] \ \text{ if } 0 \leq \underline{x}, [0, \sqrt{\overline{x}}] \ \text{ otherwise}
\end{aligned}$$

Definitions: operations

$\boldsymbol{x} \diamond \boldsymbol{y} = \textbf{Hull}\{x \diamond y \ : \ x \in \boldsymbol{x}, y \in \boldsymbol{y}\}$

Arithmetic and algebraic operations: use the monotonicity

$$
\begin{array}{rcl}
[\underline{x}, \overline{x}] + [\underline{y}, \overline{y}] & = & [\underline{x} + \underline{y}, \overline{x} + \overline{y}] \\
[\underline{x}, \overline{x}] - [\underline{y}, \overline{y}] & = & [\underline{x} - \overline{y}, \overline{x} - \underline{y}] \\
[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] & = & [\min(\underline{x} \times \underline{y}, \underline{x} \times \overline{y}, \overline{x} \times \underline{y}, \overline{x} \times \overline{y}), \max(\text{ibid.})] \\
[\underline{x}, \overline{x}]^2 & = & [\min(\underline{x}^2, \overline{x}^2), \max(\underline{x}^2, \overline{x}^2)] \text{ if } 0 \notin [\underline{x}, \overline{x}] \\
 & & [0, \max(\underline{x}^2, \overline{x}^2)] \text{ otherwise} \\
1/[\underline{y}, \overline{y}] & = & [\min(1/\underline{y}, 1/\overline{y}), \max(1/\underline{y}, 1/\overline{y})] \text{ if } 0 \notin [\underline{y}, \overline{y}] \\
[\underline{x}, \overline{x}]/[\underline{y}, \overline{y}] & = & [\underline{x}, \overline{x}] \times (1/[\underline{y}, \overline{y}]) \text{ if } 0 \notin [\underline{y}, \overline{y}] \\
\sqrt{[\underline{x}, \overline{x}]} & = & [\sqrt{\underline{x}}, \sqrt{\overline{x}}] \text{ if } 0 \le \underline{x}, [0, \sqrt{\overline{x}}] \text{ otherwise}
\end{array}
$$

Definitions: operations

Algebraic properties: associativity, commutativity hold, some are lost:

- subtraction is not the inverse of addition, in particular $\boldsymbol{x} - \boldsymbol{x} \neq [0]$
- division is not the inverse of multiplication
- squaring is tighter than multiplication by oneself
- multiplication is only sub-distributive wrt addition: $\boldsymbol{x} \times (\boldsymbol{y} + \boldsymbol{z}) \subset \boldsymbol{x} \times \boldsymbol{y} + \boldsymbol{x} \times \boldsymbol{z}$:

Lost algebraic properties: examples.

- subtraction is not the inverse of addition

$$[1,3]-[1,3]=[1,3]+[-3,-1]=[-2,2]\supset[0,0]$$

- division is not the inverse of multiplication

$$[1,3]/[1,3]=[1,3]\times[\frac{1}{3},1]=[\frac{1}{3},3]\supset[1,1]$$

- squaring is tighter than multiplication by oneself

$$[-1,2]\times[-1,2]=[-2,4]\supset[-1,2]^2=[0,4]$$

- multiplication is only sub-distributive wrt addition

$$[-1,2]\times([3,4]+[-6,2])=[-1,2]\times[-3,6]=[-6,12]$$

but $[-1,2]\times[3,4]+[-1,2]\times[-6,2]=[-4,8]+[-12,6]=[-16,14]$

Definitions: functions

Definition:
an interval extension $\boldsymbol{f}$ of a function f satisfies

$$\forall \boldsymbol{x},\ f(\boldsymbol{x}) \subset \boldsymbol{f}(\boldsymbol{x}),\ \text{and } \forall x,\ f(\{x\}) = \boldsymbol{f}(\{x\}).$$

Elementary functions: again, use the monotony.

$$\begin{array}{lcl}
\exp \boldsymbol{x} & = & [\exp \underline{x}, \exp \overline{x}] \\
\log \boldsymbol{x} & = & [\log \underline{x}, \log \overline{x}] \text{ if } \underline{x} \geq 0, [-\infty, \log \overline{x}] \text{ if } \overline{x} > 0 \\
\sin[\pi/6, 2\pi/3] & = & [1/2, 1] \\
\ldots & &
\end{array}$$

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of ”multi-valued quantities”
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of "multi-valued quantities"
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

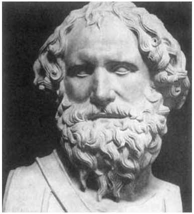
Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

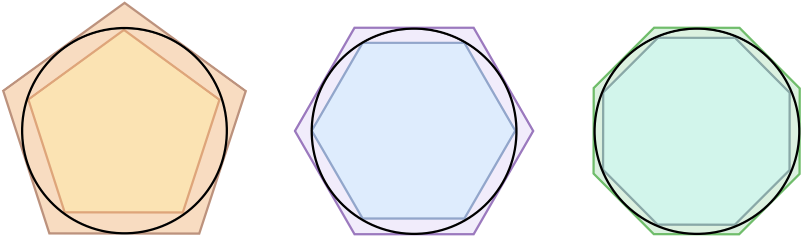
Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga, in his MSc thesis in Japanese
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has studied the algebra of ”multi-valued quantities”
- **1927:** Bradis, for positive quantities, in Russian
- **1908:** Young, for some bounded functions, in Italian
- **3rd century BC:** Archimedes, to compute an enclosure of π!

Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

Archimedes and an enclosure for π





$$\frac{223}{71} < \pi < \frac{22}{7}$$

Historical remarks

Childhood until the seventies.

Popularization in the 1980, German school (U. Kulisch).

IEEE-754 standard for floating-point arithmetic in 1985:
directed roundings are standardized and available (?).

Since the nineties: interval **algorithms**.

IEEE-1788 standard for interval arithmetic in 2015
simplified version in 2017.

Now: **applications**.

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Definition: function extension

Example: $f(x) = x^2 - x + 1$ with $x \in [-2, 1]$.
$[-2,1]^2 - [-2,1] + 1 = [0,4] + [-1,2] + 1 = [0,7]$.
Since $x^2 - x + 1 = x(x-1) + 1$, we get $[-2,1] \cdot ([-2,1] - 1) + 1 = [-2,1] \cdot [-3,0] + 1 = [-3,6] + 1 = [-2,7]$.
Since $x^2 - x + 1 = (x - 1/2)^2 + 3/4$, we get $([-2,1] - 1/2)^2 + 3/4 = [-5/2, 1/2]^2 + 3/4 = [0, 25/4] + 3/4 = [3/4, 7] = f([-2,1])$.

Problem with this definition: infinitely many interval extensions, syntactic use (instead of semantic).

How to choose the best extension? How to choose a good one?

Definition: function extension

Example: $f(x) = x^2 - x + 1$ with $x \in [-2, 1]$.
$[-2,1]^2 - [-2,1] + 1 = [0,4] + [-1,2] + 1 = [0,7]$.
Since $x^2 - x + 1 = x(x-1) + 1$, we get $[-2,1] \cdot ([-2,1] - 1) + 1 = [-2,1] \cdot [-3,0] + 1 = [-3,6] + 1 = [-2,7]$.
Since $x^2 - x + 1 = (x - 1/2)^2 + 3/4$, we get $([-2,1] - 1/2)^2 + 3/4 = [-5/2, 1/2]^2 + 3/4 = [0, 25/4] + 3/4 = [3/4, 7] = f([-2,1])$.

Problem with this definition: infinitely many interval extensions, syntactic use (instead of semantic).

How to choose the best extension? How to choose a good one?

Definition: function extension

Example: $f(x) = x^2 - x + 1$ with $x \in [-2, 1]$.
$[-2, 1]^2 - [-2, 1] + 1 = [0, 4] + [-1, 2] + 1 = [0, 7]$.
Since $x^2 - x + 1 = x(x - 1) + 1$, we get $[-2, 1] \cdot ([-2, 1] - 1) + 1 = [-2, 1] \cdot [-3, 0] + 1 = [-3, 6] + 1 = [-2, 7]$.
Since $x^2 - x + 1 = (x - 1/2)^2 + 3/4$, we get $([-2, 1] - 1/2)^2 + 3/4 = [-5/2, 1/2]^2 + 3/4 = [0, 25/4] + 3/4 = [3/4, 7] = f([-2, 1])$.

Problem with this definition: infinitely many interval extensions, syntactic use (instead of semantic).

How to choose the best extension? How to choose a good one?

Definition: function extension

Example: $f(x) = x^2 - x + 1$ with $x \in [-2, 1]$.
$[-2, 1]^2 - [-2, 1] + 1 = [0, 4] + [-1, 2] + 1 = [0, 7]$.
Since $x^2 - x + 1 = x(x - 1) + 1$, we get $[-2, 1] \cdot ([-2, 1] - 1) + 1 = [-2, 1] \cdot [-3, 0] + 1 = [-3, 6] + 1 = [-2, 7]$.
Since $x^2 - x + 1 = (x - 1/2)^2 + 3/4$, we get $([-2, 1] - 1/2)^2 + 3/4 = [-5/2, 1/2]^2 + 3/4 = [0, 25/4] + 3/4 = [3/4, 7] = f([-2, 1])$.

Problem with this definition: infinitely many interval extensions, syntactic use (instead of semantic).

How to choose the best extension? How to choose a good one?

Definition: function extension

Mean value theorem of order 1 (Taylor expansion of order 1):

$\forall x, \forall y, \exists \xi_{x,y} \in (x,y) \ : \ f(y) = f(x) + (y-x) \cdot f'(\xi_{x,y})$

Interval interpretation:

$\forall y \in \boldsymbol{x}, \forall \tilde{x} \in \boldsymbol{x}, \ f(y) \in f(\tilde{x}) + (y - \tilde{x}) \cdot \boldsymbol{f'}(\boldsymbol{x})$

$\Rightarrow f(\boldsymbol{x}) \subset f(\tilde{x}) + (\boldsymbol{x} - \tilde{x}) \cdot \boldsymbol{f'}(\boldsymbol{x})$

Same example: $\boldsymbol{x} = [-2, 1]$, $f(x) = x^2 - x + 1$
$\tilde{x} = \mathrm{mid}(\boldsymbol{x}) = \mathrm{mid}([-2, 1]) = -0.5$ and $f'(x) = 2x - 1$.

Taylor expansion of order 1: $f(\boldsymbol{x}) \subset f(\tilde{x}) + (\boldsymbol{x} - \tilde{x}) \cdot \boldsymbol{f'}(\boldsymbol{x})$

$$\begin{aligned}
f([-2,1]) &\subset f(-0.5) + ([-2,1] + 0.5) \cdot (2 \cdot [-2,1] - 1) \\
&\subset \tfrac{7}{4} + [-\tfrac{3}{2}, \tfrac{3}{2}] \cdot ([-4,2] - 1) \\
&\subset \tfrac{7}{4} + [-\tfrac{3}{2}, \tfrac{3}{2}] \cdot [-5,1] \\
&\subset \tfrac{7}{4} + [-\tfrac{15}{2}, \tfrac{15}{2}] \\
&\subset [-\tfrac{23}{4}, \tfrac{37}{4}] \\
&\subset [-5.75, 9.25]
\end{aligned}$$

Definition: function extension

Mean value theorem of order 2 (Taylor expansion of order 2):

$\forall x, \forall y, \exists \xi_{x,y} \in (x,y) : f(y) = f(x) + (y-x) \cdot f'(x) + \frac{(y-x)^2}{2} \cdot f''(\xi_{x,y})$

Interval interpretation:

$\forall y \in \boldsymbol{x}, \forall \tilde{x} \in \boldsymbol{x},\ f(y) \in f(\tilde{x}) + (y - \tilde{x}) \cdot f'(\tilde{x}) + \frac{(y-\tilde{x})^2}{2} \cdot \boldsymbol{f''}(\boldsymbol{x})$

$\Rightarrow f(\boldsymbol{x}) \subset f(\tilde{x}) + (\boldsymbol{x} - \tilde{x}) \cdot f'(\tilde{x}) + \frac{(\boldsymbol{x}-\tilde{x})^2}{2} \cdot \boldsymbol{f''}(\boldsymbol{x})$

Same example: $\boldsymbol{x} = [-2, 1]$, $f(x) = x^2 - x + 1$

$\tilde{x} = \text{mid}(\boldsymbol{x}) = -0.5$ and $f'(x) = 2x - 1$, $f''(x) = 2$.

Taylor expansion of order 2:

$$f(\boldsymbol{x}) \subset f(\tilde{x}) + (\boldsymbol{x} - \tilde{x}) \cdot f'(\tilde{x}) + \frac{(\boldsymbol{x} - \tilde{x})^2}{2} \cdot \boldsymbol{f''}(\boldsymbol{x})$$

$$\begin{aligned}
f([-2,1]) &\subset f(-0.5) + ([-2,1] + 0.5) \cdot f'(-0.5) + \frac{([-2,1]+0.5)^2}{2} \cdot f''([-2,1]) \\
&\subset \frac{7}{4} - 2 \cdot [-\frac{3}{2}, \frac{3}{2}] + 2 \cdot \frac{[-\frac{3}{2}, \frac{3}{2}]^2}{2} \\
&\subset \frac{7}{4} - [-3, 3] + [0, \frac{9}{4}] \\
&\subset [\frac{7-12}{4}, \frac{7+12+9}{4}] \\
&\subset [\frac{-5}{4}, \frac{28}{4}] \\
&\subset [-1.25, 7]
\end{aligned}$$

Definition: function extension

No need to go further:

- it is difficult to compute (automatically) the derivatives of higher order,
  especially for multivariate functions;
- there is no (theoretical) gain in quality.

Theorem:

- for the natural extension $\boldsymbol{f}$ of f, it holds $d(f(\boldsymbol{x}), \boldsymbol{f}(\boldsymbol{x})) \leq \mathcal{O}(w(\boldsymbol{x}))$
- for the first order Taylor expansion $\boldsymbol{f_{T_1}}$ of f, it holds $d(f(\boldsymbol{x}), \boldsymbol{f_{T_1}}(\boldsymbol{x})) \leq \mathcal{O}(w(\boldsymbol{x})^2)$
- getting an order higher than 3 is impossible without the squaring operation, is difficult even with it. . .

Definition: function extension

Some heuristics to increase the accuracy:

- try different evaluation schemes (naive extension, Taylor expansions, Horner scheme. . .)
- bisect the input interval and evaluate for each sub-interval, then take the hull (convex hull of the union)
- . . .

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Definitions: comparisons

How to compare two intervals?
how to compare $[-1, 2]$ and $[0, 3]$? or $[-1, 2]$ and $[0, 1]$?

Several approaches:

- use explicit names: CertainlyLess, PossiblyLess
- use trivalued logic (MPFI): $\boldsymbol{a} < \boldsymbol{b}$ returns
  - -1 if every element of $\boldsymbol{a}$ is $<$ than every element of $\boldsymbol{b}$,
  - $+1$ if every element of $\boldsymbol{a}$ is $>$ than every element of $\boldsymbol{b}$,
  - 0 if $\boldsymbol{a}$ and $\boldsymbol{b}$ overlap.
- use many more relation names, cf. IEEE 1788.

IEEE-1788 standard: comparison relations

- **7 relations:** equal ($=$), subset ($\subset$), less than or equal to ($\leq$), precedes or touches ($\preceq$), interior to, less than ($<$), precedes ($\prec$).
- **Interval overlapping relations:** before, meets, overlaps, starts, containedBy, finishes, equal, finishedBy, contains, startedBy, overlappedBy, metBy, after.

  Again, relations defined by conditions on the bounds.

IEEE-1788 standard: comparison relations

- **7 relations:** equal ($=$), subset ($\subset$), less than or equal to ($\leq$), precedes or touches ($\preceq$), interior to, less than ($<$), precedes ($\prec$).
- **Interval overlapping relations:** before, meets, overlaps, starts, containedBy, finishes, equal, finishedBy, contains, startedBy, overlappedBy, metBy, after.

  Again, relations defined by conditions on the bounds.

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

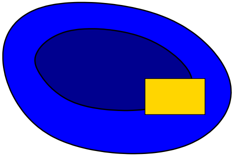
- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

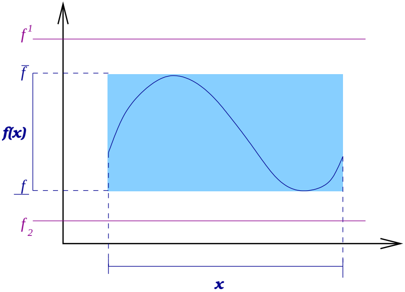
Pros: set computing

Behaviour safe? controllable? dangerous?



always controllable.

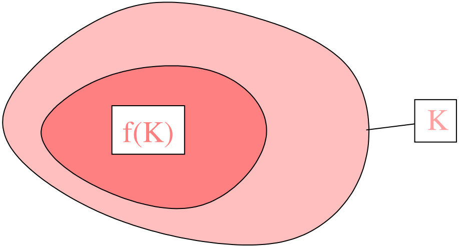
On $\boldsymbol{x}$, are the extrema of the function f $> f^1$, $< f_2$?



No if $f(\boldsymbol{x}) = [\underline{f}, \overline{f}] \subset [f_2, f^1]$.

Pros: Brouwer-Schauder theorem

A function f which is continuous on the unit ball B and which satisfies $f(B) \subset B$ has a fixed point on B.
Furthermore, if $f(B) \subset \mathring{B}$ and $|f'(B)| < 1$ then f has a unique fixed point on B.



The theorem remains valid if B is replaced by a compact K and in particular an interval.

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Cons: overestimation (1/2)

The result encloses the true result, but it is too large:
overestimation phenomenon.
Two main sources: variable dependency and wrapping effect.

(Loss of) Variable dependency:

$$\boldsymbol{x} - \boldsymbol{x} = \{x - y \ : \ x \in \boldsymbol{x}, y \in \boldsymbol{x}\} \neq \{x - x \ : \ x \in \boldsymbol{x}\} = \{0\}.$$

Cons: overestimation (2/2)

Wrapping effect

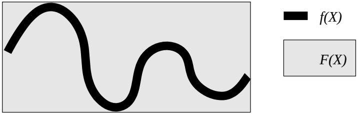
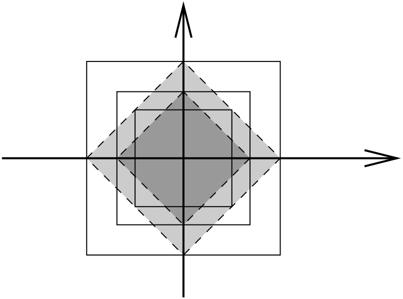


image of $f(\boldsymbol{x})$
with $f : \boldsymbol{R}^2 \to \boldsymbol{R}^2$



2 successive rotations of $\pi/4$
of the little central square

Cons: Complexity: almost every problem is NP-hard

Gaganov 1982, Rohn 1994 ff, Kreinovich. . .

- evaluate a function on a box (cartesian product of intervals)
- evaluate a function on a box up to ε
- solve a linear system
- solve a linear system up to $1/4n^4$ ($n =$ dim. of the system)
- determine if the solution of a linear system is bounded
- compute the matrix norm $\|\boldsymbol{A}\|_{\infty,1}$
- determine if an interval matrix ($=$ a matrix with interval coefficients) is regular, i.e. if every possible punctual matrix in it is regular
- . . .

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

Idea: reduce polynomially CNF-3 to this problem.

On n boolean variables $q_1, \cdots, q_n$, a formula f in CNF-3 is defined by

$$f = \bigwedge_{i=1}^{m} f_i \text{ with } f_i = \bigvee_{j=1}^{1,2 or 3} r_{i,j}$$

with $r_{i,j} = q_{k_{i,j}}$ or $r_{i,j} = \neg q_{k_{i,j}}$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

To each boolean variable q_i, let us associate a real variable $x_i \in [0,1]$.
Meaning: $x_i = 0$ if $q_i = F$ and $x_i = 1$ if $q_i = T$.

Goal: get a polynomial which takes only values in $[0,1]$
i.e. allow only product of terms or of $(1-$ term).
A product corresponds to a conjunction and $1-x$ to a negation
$\Rightarrow$ express f and the f_i using conjonctions and negations
$\Rightarrow$ express the f_i as $\neg \bigwedge_{j=1}^{1,2or3} \neg r_{i,j}$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

To each boolean variable q_i, let us associate a real variable $x_i \in [0,1]$.
Meaning: $x_i = 0$ if $q_i = F$ and $x_i = 1$ if $q_i = T$.

Goal: get a polynomial which takes only values in $[0,1]$
i.e. allow only product of terms or of $(1-$ term).
A product corresponds to a conjunction and $1-x$ to a negation
$\Rightarrow$ express f and the f_i using conjonctions and negations
$\Rightarrow$ express the f_i as $\neg \bigwedge_{j=1}^{1,2or3} \neg r_{i,j}$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

To each boolean variable q_i, let us associate a real variable $x_i \in [0, 1]$.
Meaning: $x_i = 0$ if $q_i = F$ and $x_i = 1$ if $q_i = T$.

Goal: get a polynomial which takes only values in $[0, 1]$
i.e. allow only product of terms or of (1− term).
A product corresponds to a conjunction and $1 - x$ to a negation
$\Rightarrow$ express f and the f_i using conjonctions and negations
$\Rightarrow$ express the f_i as $\neg \bigwedge_{j=1}^{1,2or3} \neg r_{i,j}$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

To each boolean variable q_i, let us associate a real variable $x_i \in [0, 1]$.
Meaning: $x_i = 0$ if $q_i = F$ and $x_i = 1$ if $q_i = T$.

Goal: get a polynomial which takes only values in $[0, 1]$
i.e. allow only product of terms or of $(1-$ term).
A product corresponds to a conjunction and $1 - x$ to a negation
$\Rightarrow$ express f and the f_i using conjonctions and negations
$\Rightarrow$ express the f_i as $\neg \bigwedge_{j=1}^{1,2or3} \neg r_{i,j}$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

More precisely:

1. to each $r_{i,j}$ let us associate a polynomial $y_{i,j}$ (corresponding to the negation of $r_{i,j}$) defined by
$$\begin{array}{rclcrcl} r_{i,j} & = & q_{k_{i,j}} & \rightarrow & y_{i,j}(x) & = & 1 - x_{k_{i,j}} \\ r_{i,j} & = & \neg q_{k_{i,j}} & \rightarrow & y_{i,j}(x) & = & x_{k_{i,j}} \end{array}$$
2. to each f_i, let us associate a polynomial p_i (corresponding to the negation of f_i) defined by
$f_i = \bigvee r_{i,j} = \neg \bigwedge \neg r_{i,j} \; \rightarrow \; p_i(x) = \prod y_{i,j}(x)$.
3. to f, let us associate the polynomial p defined by
$f = \bigwedge_{i=1}^{m} f_i \; \rightarrow \; p(x) = \prod_{i=1}^{m}(1 - p_i(x))$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

More precisely:

1. to each $r_{i,j}$ let us associate a polynomial $y_{i,j}$ (corresponding to the negation of $r_{i,j}$) defined by
$$\begin{array}{rclcrcl} r_{i,j} & = & q_{k_{i,j}} & \rightarrow & y_{i,j}(x) & = & 1 - x_{k_{i,j}} \\ r_{i,j} & = & \neg q_{k_{i,j}} & \rightarrow & y_{i,j}(x) & = & x_{k_{i,j}} \end{array}$$
2. to each f_i, let us associate a polynomial p_i (corresponding to the negation of f_i) defined by
$f_i = \bigvee r_{i,j} = \neg \bigwedge \neg r_{i,j} \ \rightarrow \ p_i(x) = \prod y_{i,j}(x).$
3. to f, let us associate the polynomial p defined by
$f = \bigwedge_{i=1}^{m} f_i \ \rightarrow \ p(x) = \prod_{i=1}^{m} (1 - p_i(x)).$

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

More precisely:

1. to each $r_{i,j}$ let us associate a polynomial $y_{i,j}$ (corresponding to the negation of $r_{i,j}$) defined by
$$\begin{array}{rclcrcl} r_{i,j} & = & q_{k_{i,j}} & \to & y_{i,j}(x) & = & 1 - x_{k_{i,j}} \\ r_{i,j} & = & \neg q_{k_{i,j}} & \to & y_{i,j}(x) & = & x_{k_{i,j}} \end{array}$$
2. to each f_i, let us associate a polynomial p_i (corresponding to the negation of f_i) defined by
$f_i = \bigvee r_{i,j} = \neg \bigwedge \neg r_{i,j} \;\to\; p_i(x) = \prod y_{i,j}(x)$.
3. to f, let us associate the polynomial p defined by
$f = \bigwedge_{i=1}^{m} f_i \;\to\; p(x) = \prod_{i=1}^{m}(1 - p_i(x))$.

Cons: Complexity: Gaganov 1982

evaluation of a multivariate polynomial with rational coeff. on a box is NP-hard

Lemma:

1. $\forall x \in [0,1]^n$, $p(x) \in [0,1]$.
2. if α is a boolean vector and β is the associated $0-1$ vector, then

$$\begin{array}{ccccccc} f(\alpha) & = & T & \Rightarrow & p(\beta) & = & 1 \\ f(\alpha) & = & F & \Rightarrow & p(\beta) & = & 0. \end{array}$$

3. if f is not feasible, then $\forall x \in [0,1]^n$, $p(x) \leq 7/8$.

Cons: Complexity: Gaganov 1982

Proof of (3): (proving (1) and (2) is easy).
$\forall x \in [0,1]^n$, let us consider β the 0-1 vector obtained by rounding x to the nearest.
Since f is not feasible, $p(\beta) = 0$.
Since $p(x) = \prod_{i=1}^{m}(1 - p_i(x))$, $\exists i_0$ such that $1 - p_{i_0}(\beta) = 0$.
One can prove that $p_{i_0}(x) \geq 1/8$, using the fact that it is the product of at most three terms, each of them $\geq 1/2$, using the fact that β is the rounding to nearest of x. Thus $1 - p_{i_0}(x) \leq 7/8$.
The remaining factors $1 - p_j(x)$ are less or equal to 1.
Thus $p(x) = \prod_{i=1}^{m}(1 - p_i(x)) \leq 7/8$.

Consequence: since checking the feasibility of a CNF-3 formula is NP-hard, evaluating a multivariate polynomial (up to a small ε) is NP-hard.

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Exo1: Using Julia and IntervalArithmetic.jl

Install Julia, install IntervalArithmetic

```
using IntervalArithmetic;
setdisplay(:full);

x=interval(0,10)
   Interval{Float64}(0.0, 10.0, com, true)
sin(pi*x)
   Interval{Float64}(-1.0, 1.0, com, true)
function f(x)
sin(pi*x)+3*exp(-x)-interval(-2,3)
end
f(x)
   Interval{Float64}(-3.9998638002107127, 6.0, com, false)
```

Several extensions (1/2)

Evaluate the function $f : x \mapsto (x-1)\cdot(x-2)\cdot(x-3)\cdot(x-4)\cdot(x-5)$ on the intervals $\boldsymbol{x} = [0,6]$ and $\boldsymbol{y} = [0,1]$ using different schemes:

- naive evaluation,
- naive evaluation of the developed form: $f(x) = x^5 - 15\cdot x^4 + 85\cdot x^3 - 225\cdot x^2 + 274\cdot x - 120$,
- using the developed form, Taylor expansions of orders 1, 2, 3, 4 around the midpoint,
- for $\boldsymbol{x}$ only, using the splitting of x into $\boldsymbol{x_1} = [0,1]$, $\boldsymbol{x_2} = [1,2]$, $\boldsymbol{x_3} = [2,3]$, $\boldsymbol{x_4} = [3,4]$, $\boldsymbol{x_5} = [4,5]$, $\boldsymbol{x_6} = [5,6]$ and the union of all results, for each of the previous methods.

What are the results of the intersections of all these evaluations, for $\boldsymbol{x}$, for $\boldsymbol{y}$?

Several extensions (2/2)

Evaluate the function $g : x \mapsto \exp(x) \cdot \sin(x) + \frac{x}{x^2+x+1}$ on the intervals $\boldsymbol{x} = [0, 6]$, $\boldsymbol{y} = [2, 3]$ and $\boldsymbol{z} = [2.3, 2.5]$ using different schemes:

- naive evaluation,
- using the Taylor expansions of orders 1, 2, 3 around the midpoint (see next slide),
- for $\boldsymbol{x}$ only, using the splitting of x into $\boldsymbol{x_1} = [0, 1]$, $\boldsymbol{x_2} = [1, 2]$, $\boldsymbol{x_3} = [2, 3]$, $\boldsymbol{x_4} = [3, 4]$, $\boldsymbol{x_5} = [4, 5]$, $\boldsymbol{x_6} = [5, 6]$ and the union of all results, for each of the previous methods.

What are the results of the intersections of all these evaluations, for $\boldsymbol{x}$, for $\boldsymbol{y}$, for $\boldsymbol{z}$?

Help (Maple is my friend)

$$
\begin{aligned}
g(x) &= \exp(x)\cdot\sin(x) + \tfrac{x}{x^2+x+1} \\
g'(x) &= \exp(x)\cdot\sin(x) + \exp(x)\cdot\cos(x) + \tfrac{1}{x^2+x+1} + \tfrac{2x^2+x}{(x^2+x+1)^2} \\
g^{(2)}(x) &= 2\cdot\exp(x)\cdot\cos(x) - \tfrac{6x+2}{(x^2+x+1)^2} + \tfrac{8x^3+8x^2+2x}{(x^2+x+1)^3} \\
g^{(3)} &= 2\cdot\exp(x)\cdot\cos(x) - 2\cdot\exp(x)\cdot\sin(x) - \tfrac{6}{(x^2+x+1)^2} \\
&\quad + \tfrac{48x^2+36x+6}{(x^2+x+1)^3} - \tfrac{48x^4+72x^3+36x^2+6x}{(x^2+x+1)^4}
\end{aligned}
$$

Send your results to Nathalie.Revol@ens-lyon.fr at latest Nov 27th, 10h15 am. Include your code lines along with the computed results and some comments about the results – around 250 words.

Agenda

- Definitions of interval arithmetic
  - Operations
  - History
  - Function extensions
  - Comparisons
- Pros and cons
  - Pros: contractant iterations, Brouwer's theorem
  - Cons: overestimation, complexity
- Complexity
- Homework
  - Exercise 1
  - Exercise 2*
- Implementation issues
  - Floating-point arithmetic
  - Representation
  - Arbitrary precision

Exo 2^*: Brouwer-Schauder theorem

For each question, exhibit one example of f along with an interval $\boldsymbol{x}$ such that

1. $f(\boldsymbol{x}) \subset \boldsymbol{x}$ and even an example of $\boldsymbol{f}(\boldsymbol{x}) \subset \boldsymbol{x}$ with a naive evaluation of f: what is/are the fixed point/s of f in $\boldsymbol{x}$?
2. $\boldsymbol{f}(\boldsymbol{x}) \subset \mathring{\boldsymbol{x}}$ and $|f'(\boldsymbol{x})| < 1$: f has a unique fixed point in $\boldsymbol{x}$, give the fixed point
3. $\boldsymbol{f}(\boldsymbol{x}) \subsetneq \boldsymbol{x}$ but the fixed point is not unique, give the fixed points

Notation: $\mathring{\boldsymbol{x}}$ is the interior of $\boldsymbol{x}$.

Expected: at most one page with examples and calculations, legibly hand-written, paper or scanned version are accepted, no later than Nov 27th, 10h15 am.

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Floating-point arithmetic

Floating-point number: cf. CR07.
Operations: performed as if performed exactly, then rounded to the target format.
Rounding modes

- RU: Round Upward or Round Toward $+\infty$
- RD: Round Downward or Round Toward $-\infty$
- RZ: Round Toward Zero
- RA: Round Away from Zero
- RN: Round to Nearest – in case of a tie: round to nearest-even

Floating-point arithmetic

Implementation using floating-point arithmetic:
use directed roundings, towards $\pm\infty$.
Overhead in execution time:
in theory, at most 4, or 8, cf.

$$\begin{aligned}[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] \quad = [\quad & \min(\mathrm{RD}(\underline{x} \times \underline{y}), \mathrm{RD}(\underline{x} \times \overline{y}), \mathrm{RD}(\overline{x} \times \underline{y}), \mathrm{RD}(\overline{x} \times \overline{y})), \\ & \max(\mathrm{RU}(\underline{x} \times \underline{y}), \mathrm{RU}(\underline{x} \times \overline{y}), \mathrm{RU}(\overline{x} \times \underline{y}), \mathrm{RU}(\overline{x} \times \overline{y}))]\end{aligned}$$

in practice, around 20: changing the rounding modes implies flushing the pipelines (on most architectures and implementations).

Example of use: geometrical computations

Melquiond, Pion (2005)

Compute the position of a point P with respect to a line $\mathcal{D}$ defined by two points A and B in $\mathbb{R}^2$:
coordinates $P : (x_P, y_P)$, $A : (x_A, y_A)$, $B : (x_B, y_B)$.

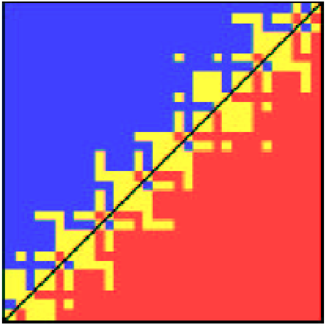
- $\mathcal{D}$ given by the equation $ax + by + c = 0$: sign of $ax_P + by_P + c$
- sign of one of the determinants

$$\det \begin{pmatrix} x_A & x_B & x_P \\ y_A & y_B & y_P \\ 1 & 1 & 1 \end{pmatrix} \text{ or } \det \begin{pmatrix} x_B - x_A & x_P - x_A \\ y_B - y_A & y_P - y_A \end{pmatrix}$$

Example of use: geometrical computations

L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap (2008)

With roundoff errors, P can be on either side of $\mathcal{D}$ or inside a "triangle":



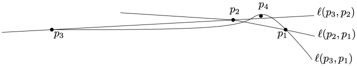
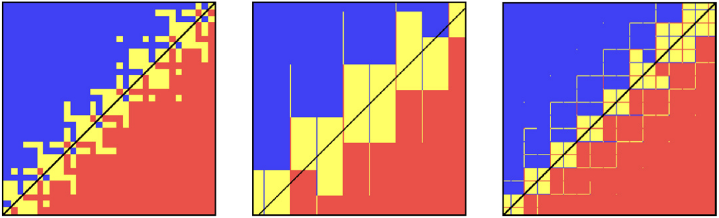


Fig. 4. Schematics: The point p_4 sees all edges of the triangle (p_1, p_2, p_3).

Example of use: geometrical computations

L. Kettner, K. Mehlhorn, S. Pion, S. Schirra, and C. Yap (2008)



$$p : \begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \quad q : \begin{pmatrix} 12 \\ 12 \end{pmatrix} \quad r : \begin{pmatrix} 24 \\ 24 \end{pmatrix}$$

(a)

$$\begin{pmatrix} 0.50000000000002531 \\ 0.5000000000000171 \end{pmatrix} \begin{pmatrix} 17.300000000000001 \\ 17.300000000000001 \end{pmatrix} \begin{pmatrix} 24.00000000000005 \\ 24.0000000000000517765 \end{pmatrix}$$

(b)

$$\begin{pmatrix} 0.5 \\ 0.5 \end{pmatrix} \begin{pmatrix} 8.8000000000000007 \\ 8.8000000000000007 \end{pmatrix} \begin{pmatrix} 12.1 \\ 12.1 \end{pmatrix}$$

(c)

Fig. 2. The weird geometry of the float-orientation predicate: The figure shows the results of *float_orient*$(p_x + Xu_x, p_y + Yu_y, q, r)$ for $0 \leqslant X, Y \leqslant 255$, where $u_x = u_y = 2^{-53}$ is the increment between adjacent floating-point numbers in the considered range. The result is color coded: Yellow (red, blue, resp.) pixels represent collinear (negative, positive, resp.) orientation. The line through q and r is shown in black.

Some libraries

Main difference: the employed programming language

- based on an existing, easy-to-use software: IntLab (not free) for MatLab (not free), interval for Octave (deprecated), Int4Sci for Scilab (deprecated)
- for Fortrain: IntLib, Cosy
- for Pascal, Fortran, C: XSC (no more maintained)
- for C / C++: Filib, Profil/BIAS (no more maintained)
- for C++: libieee1788 (no more maintained)
- for C++: Gaol, IBEX
- for Julia: IntervalArithmetic.jl

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Representation

By endpoints: the representation used so far, e.g. $[-2, 1]$.

$$[\underline{x}, \bar{x}] = \{x \; : \; \underline{x} \leq x \leq \bar{x}\}.$$

By mid-rad: e.g. $< -0.5, 1.5 >$.

$$< x_m, x_r >= \{x \; : \; |x - x_m| \leq x_r\} \text{ in } \mathbb{R}.$$

More generally, in $\mathbb{R}^n$,

$$< x_m, x_r >= \{x \; : \; x_m - x_r \leq x \leq x_m + x_r\}$$

where $\leq$ is taken componentwise.

Operations for mid-rad

Addition and subtraction:

$$< x_m, x_r > \pm < y_m, y_r > = < x_m \pm y_m, x_r + y_r >$$

Multiplication:

$$< x_m, x_r > \times < y_m, y_r > \neq < x_m \cdot y_m, f(x_r, y_r) >$$

for some function f.
Example: $< 2, 1 > \times < 5, 2 > \neq < 10, \ldots >$.
Back to endpoints:

$$< 2, 1 > \times < 5, 2 > = [1, 3] \times [3, 7] = [3, 21] = < 12, 9 >$$

Operations for mid-rad

Multiplication: But

$$< x_m, x_r > \times < y_m, y_r > \subset < x_m \cdot y_m, x_r \cdot |y_m| + |x_m| \cdot y_r + x_r \cdot y_r >$$

$$< 2, 1 > \times < 5, 2 > \subset < 2 \cdot 5, 1 \cdot 5 + 2 \cdot 2 + 1 \cdot 2 >=< 10, 11 >= [-1, 21]$$

Another example:

$$< 2, 0.1 > \times < 5, 0.5 >= [1.9, 2.1] \times [4.5, 5.5] = [8.55, 11.55]$$

is close to

$$\begin{aligned} < 2, 0.1 > \times < 5, 0.5 > \quad &\subset \quad < 10, 0.5 + 1 + 0.05 > \\ &= \quad < 10, 1.55 > \\ &= \quad [8.45, 11.55] \end{aligned}$$

Matrix operation for mid-rad

Let $\boldsymbol{A}$ and $\boldsymbol{B}$ be two interval matrices of compatible dimensions.

Addition/subtraction: $\boldsymbol{A} \pm \boldsymbol{B} = (\boldsymbol{A}_{i,j} \pm \boldsymbol{B}_{i,j})_{i,j}$:
easy for both representations.

Multiplication:

$$< A_m, A_r > \times < B_m, B_r > \subset < A_m \cdot B_m, A_r \cdot |B_m| + |A_m| \cdot B_r + A_r \cdot B_r >$$

which can also be expressed as

$$< A_m, A_r > \times < B_m, B_r > \subset < A_m \cdot B_m, A_r \cdot (|B_m| + B_r) + |A_m| \cdot B_r >$$

or

$$< A_m, A_r > \times < B_m, B_r > \subset < A_m \cdot B_m, (|A_m| + A_r) \cdot (|B_m| + B_r) - |A_m| \cdot |B_m$$

one multiplication less!

Matrix operation for mid-rad

Let ***A*** and ***B*** be two interval matrices of compatible dimensions.

Algorithm 9 MMMul2

```
Input: an $m$-by-$k$ interval matrix $\boldsymbol{A} = \langle M_{\boldsymbol{A}}, R_{\boldsymbol{A}} \rangle$,
    a $k$-by-$n$ interval matrix $\boldsymbol{B} = \langle M_{\boldsymbol{B}}, R_{\boldsymbol{B}} \rangle$
Output: $\widetilde{\boldsymbol{C}}_2 \supseteq \boldsymbol{AB}$
  $\widetilde{e} \leftarrow \max\{\mathrm{fl}_\triangle(R_{\boldsymbol{A}_{ij}}/|M_{\boldsymbol{A}_{ij}}|) : i = 1, \ldots, m \text{ and } j = 1, \ldots, k\}$
  $\widetilde{f} \leftarrow \max\{\mathrm{fl}_\triangle(R_{\boldsymbol{B}_{ij}}/|M_{\boldsymbol{B}_{ij}}|) : i = 1, \ldots, k \text{ and } j = 1, \ldots, n\}$
  $\widetilde{M}_2 \leftarrow \mathrm{fl}_\square\left(M_{\boldsymbol{A}} M_{\boldsymbol{B}}\right)$
  $\widetilde{\Gamma} \leftarrow \mathrm{fl}_\square\left(|M_{\boldsymbol{A}}||M_{\boldsymbol{B}}|\right)$
  $\widetilde{R}_2 \leftarrow \mathrm{fl}_\triangle\left((\widetilde{e} + \widetilde{f} + \widetilde{e}\widetilde{f} + \frac{k+2}{2}\mathtt{u})\widetilde{\Gamma} + \mathtt{realmin}\right)$
  return $\widetilde{\boldsymbol{C}}_2 = \langle \widetilde{M}_2, \widetilde{R}_2 \rangle$
```

Agenda

Definitions of interval arithmetic

- Operations
- History
- Function extensions
- Comparisons

Pros and cons

- Pros: contractant iterations, Brouwer's theorem
- Cons: overestimation, complexity

Complexity

Homework

- Exercise 1
- Exercise 2*

Implementation issues

- Floating-point arithmetic
- Representation
- Arbitrary precision

Arbitrary precision

Motivation: Kahan
Reference for this section

W. Kahan: *How Futile is Mindless Assessment of Roundoff in Floating-Point Computation?*, 2006.

Five approaches detailed in Kahan's paper

1. Repeat the computation in arithmetics of increasing precision, increase it until as many as desired of the results' digits agree.
2. Repeat the computation in arithmetic of the same precision but rounded differently, say *Down*, and then *Up*, and maybe *Towards Zero* too, besides *To Nearest*, and compare three or four results.
3. Repeat the computation a few times in arithmetic of the same precision rounding operations randomly, some *Up*, some *Down*, and treat results statistically.
4. Repeat the computation a few times in arithmetic of the same precision but with slightly different input data each time, and see how widely results spread.
5. Perform the computation in *Significance Arithmetic*, or in *Interval Arithmetic*.

The mindless use of these approaches is qualified as “futile” by Kahan.

Multiple Precision Interval Arithmetic

Almost foolproof is extendable-precision Interval Arithmetic.

Influence of the computing precision (1/2)

Influence on an interval computation: theoretically, the overestimation of the result is proportional to the ulp: $\mathrm{w}(\hat{\boldsymbol{x}}) - \mathrm{w}(\boldsymbol{x}) = \mathcal{O}(2^{-p}|\boldsymbol{x}|)$ where p is the computing precision.

Influence of the computing precision (2/2)

Influence on an interval computation: in practice,

- use the midpoint-radius representation for thin intervals: the radius accounts for roundoff errors,
- use iterative refinement to reduce the width,
- use higher precision for critical intermediate computations (residual) to hide the effect of the computing precision,

and get $\mathrm{w}(\hat{\boldsymbol{x}}) - \mathrm{w}(\boldsymbol{x}) \simeq 2^{-p}|\boldsymbol{x}|$, i.e. the best possible result.

Examples: linear systems solving, Newton iteration.

Higher precision: extended/arbitrary (not interval)

Extended precision (double-double, triple-double): (Moler, Priest, Dekker, Knuth, Shewchuk, Bailey. . .)
a number is represented as the sum of 2 (or 3 or . . .) floating-point numbers. Do not evaluate the sum using floating-point arithmetic!
Double-double arith. is implemented using IEEE-754 FP arith.

Arbitrary precision: the precision is chosen by the user, the only limit being the computer's memory.
Arithmetic is implemented in software, e.g. MPFR (**Zimmermann et al.**), MPFI (**Revol, Rouillier et al.), (Yamamoto, Krämer et al.**).

Tradeoff between accuracy and efficiency (and memory):
double-double: accuracy "$\times 2$", ≤ 1 order of magnitude slower
arbitrary prec.: accuracy "∞", $\geq$ 1-2 order of magnitude slower
(provided Higham's rule of thumb applies).

Arbitrary Precision Interval Arithmetic

Some libraries:

- **MPFI**
  MPFI stands for *Multiple Precision reliable Floating-point Interval library*;
  intervals are represented by their endpoints.
- **Arb**
  Arb is a C library for arbitrary-precision floating-point ball arithmetic;
  intervals are represented by a center and a radius.

Both are based on MPFR library for arbitrary precision: MPFR stands for *Multiple Precision Reliable Floating-point library*.
For both, the computing precision of each operation can be specified: no limit apart from the memory of your computer.