Introduction to the IEEE 1788-2015 Standard for Interval Arithmetic

Nathalie Revol and the WG 1788
Nathalie.Revol@inria.fr

NSV 2017
10th International Workshop on Numerical Software Verification
Heidelberg, Germany, 23 July 2017

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

Agenda

- Interval arithmetic in a nutshell
  - In a nutshell
  - Historical remarks
- Precious features of interval arithmetic
  - FTIA
  - Constraint solving
  - Newton and Brouwer
  - Newton and the extended division
- IEEE 1788-2015 standard
- Present
- Future
- Conclusion

Verified, guaranteed computations: Interval arithmetic

Principle

Numbers are replaced by intervals.

π is replaced by $[3.14159, 3.14160]$ or $[3.14, 3.15]$ ou $[3, 4]$.

Fundamental theorem (Thou Shalt Not Lie): the interval contains the exact value(s).

Counting without errors: Interval arithmetic

Example

my purse contains between 10 Euros and 20 Euros, $\in [10, 20]$ €.

Counting without errors: Interval arithmetic

Example

my purse contains between 10 Euros and 20 Euros, $\in [10, 20]$ €.

Your purse contains between 5 Euros and 10 Euros, $\in [5, 10]$ €.

Counting without errors: Interval arithmetic

Example

my purse contains between 10 Euros and 20 Euros, $\in [10, 20]$ €.

Your purse contains between 5 Euros and 10 Euros, $\in [5, 10]$ €.

Together, we have at least 15 Euros and no more than 30 Euros,
$[10, 20] + [5, 10] = [15, 30]$ €.

Interval arithmetic: first difficulty

Content of my purse: between 10 Euros and 20 Euros, $\in [10, 20]$ €.

Interval arithmetic: first difficulty

Content of my purse: between 10 Euros and 20 Euros, $\in [10, 20]$ €.

I visit your grand-parents and they give me an envelope for you.
This envelope contains between 10 and 20 Euros.
I put this money in my purse; my purse contains now between 20 and 40 Euros: $[10, 20] + [10, 20] = [20, 40]$ €.

Interval arithmetic: first difficulty

Content of my purse: between 10 Euros and 20 Euros, $\in [10, 20]$ €.

I visit your grand-parents and they give me an envelope for you.
This envelope contains between 10 and 20 Euros.
I put this money in my purse; my purse contains now between 20 and 40 Euros: $[10, 20] + [10, 20] = [20, 40]$ €.

I meet you and I give you your money, between 10 and 20 Euros.
My purse now contains $[20, 40] - [10, 20] = [0, 30]$ €.

Interval arithmetic: first difficulty

Content of my purse: between 10 Euros and 20 Euros, $\in [10, 20]$ €.

I visit your grand-parents and they give me an envelope for you.
This envelope contains between 10 and 20 Euros.
I put this money in my purse; my purse contains now between 20 and 40 Euros: $[10, 20] + [10, 20] = [20, 40]$ €.

I meet you and I give you your money, between 10 and 20 Euros.
My purse now contains $[20, 40] - [10, 20] = [0, 30]$ €.

In other words,

$$\text{purse} + \text{ envelope } - \text{ envelope } \neq \text{ purse.}$$

Interval arithmetic: first difficulty

Content of my purse: between 10 Euros and 20 Euros, $\in [10, 20]$ €.

I visit your grand-parents and they give me an envelope for you.
This envelope contains between 10 and 20 Euros.
I put this money in my purse; my purse contains now between 20 and 40 Euros: $[10, 20] + [10, 20] = [20, 40]$ €.

I meet you and I give you your money, between 10 and 20 Euros.
My purse now contains $[20, 40] - [10, 20] = [0, 30]$ €.

In other words,

$$\text{purse} + \text{ envelope } - \text{ envelope } \neq \text{ purse.}$$

Research on the design and writing of algorithms is needed.

Interval arithmetic: second difficulty

Regarding the implementation of interval arithmetic: each interval operation requires 2 (or 4, or ore. . .) "usual" operations.

Because of practical reasons, related to hardware issues, each interval operation may be 100 times slower (or even worse) than an operation on "usual" numbers.

Research on the efficient implementation of interval arithmetic is needed: when each operation is only 10 to 15 times slower, it is already an achievement!

Regarding algorithms, their efficiency and their complexity?
Most problems are NP-hard.

Interval arithmetic: third problem

Remember: (real) square numbers are always nonnegative $\Rightarrow$ the square root is defined only for (real) nonnegative numbers.

Interval arithmetic: third problem

Remember: (real) square numbers are always nonnegative $\Rightarrow$ the square root is defined only for (real) nonnegative numbers.

How should $\sqrt{[-1, 2]}$ be defined?

Interval arithmetic: third problem

How should $\sqrt{[-1,2]}$ be defined?

By convention, $\sqrt{[-1,2]} = \sqrt{[0,2]} = [0, \sqrt{2}]$.
The computation should signal that something odd happened:

Interval arithmetic: third problem

How should $\sqrt{[-1,2]}$ be defined?

By convention, $\sqrt{[-1,2]} = \sqrt{[0,2]} = [0,\sqrt{2}]$.
The computation should signal that something odd happened:

- how? by "decorating" intervals: 

Interval arithmetic: third problem

How should $\sqrt{[-1,2]}$ be defined?

By convention, $\sqrt{[-1,2]} = \sqrt{[0,2]} = [0, \sqrt{2}]$.
The computation should signal that something odd happened:

- how? by "decorating" intervals: 
- how can it be done without harm for the performances, in terms on computing time, memory usage?

Interval arithmetic: third problem

How should $\sqrt{[-1, 2]}$ be defined?

By convention, $\sqrt{[-1, 2]} = \sqrt{[0, 2]} = [0, \sqrt{2}]$.
The computation should signal that something odd happened:

- how? by "decorating" intervals: 
- how can it be done without harm for the performances, in terms on computing time, memory usage?

Research is needed to get efficient implementations of interval algorithms.

Interval arithmetic: third problem

How should $\sqrt{[-1,2]}$ be defined?

By convention, $\sqrt{[-1,2]} = \sqrt{[0,2]} = [0, \sqrt{2}]$.
The computation should signal that something odd happened:

- how? by "decorating" intervals: 
- how can it be done without harm for the performances, in terms on computing time, memory usage?

Research is needed to get efficient implementations of interval algorithms.

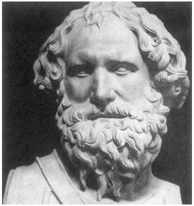
Regarding the chosen definitions and conventions: this is the topic of this talk.

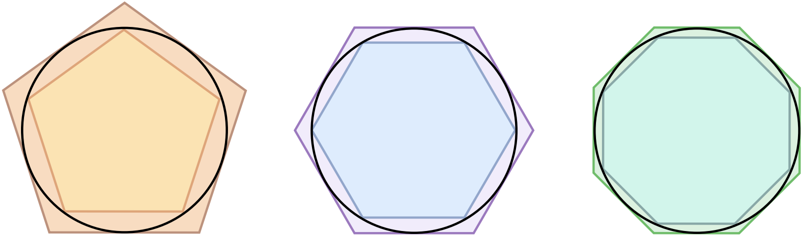
Agenda

- Interval arithmetic in a nutshell
  - In a nutshell
  - Historical remarks
- Precious features of interval arithmetic
  - FTIA
  - Constraint solving
  - Newton and Brouwer
  - Newton and the extended division
- IEEE 1788-2015 standard
- Present
- Future
- Conclusion

Archimedes and an enclosure of π

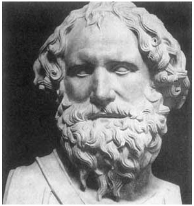
(3rd century BC)

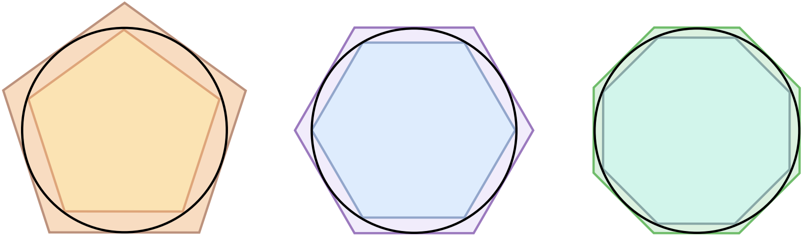




Archimedes and an enclosure of π

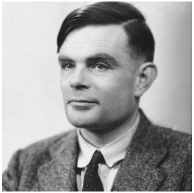
(3rd century BC)





$$3 + \frac{10}{71} \simeq 3.1408 \leq \pi \leq 3 + \frac{1}{7} \simeq 3.1429.$$

Historical remarks



Alan Turing (23-06-1912 - 07-06-1954)
according to Wilkinson in his Turing lecture (1970).

Foundations in the late fifties and in the sixties.

Childhood until the seventies.

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

IEEE-754 standard for floating-point arithmetic in 1985.

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

Fundamental Theorem of Interval Arithmetic

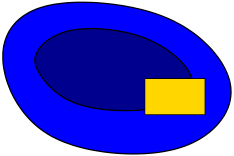
Interval computations are **guaranteed** computations.

FTIA: Fundamental Theorem of Interval Arithmetic

Let $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ be intervals,
let f be an arithmetic expression defined over $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$,
the $\boldsymbol{y}$ obtained by interval evaluation of f over $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n$ encloses the range $f(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n)$.

Set computing

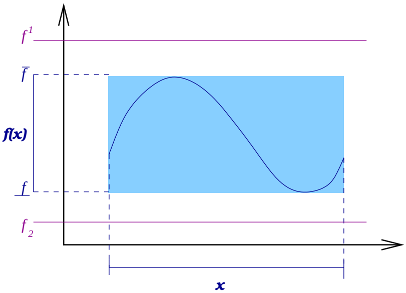
Behaviour safe? controllable? dangerous?



always controllable.

Set computing

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



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

Relation between numerical analysis of roundoff errors and interval arithmetic

Old and nontrivial question **von Neumann, Turing, Wilkinson, ...**

At least 2 ways to look at this question:

A priori analysis:

- Goal: bound on $\|\hat{x} - x\| / \|x\|$ where x is the exact value and $\hat{x}$ the computed value, for **any** input and format
- Tool: the many nice properties of **floating-point**
- Ideal: readable, provably tight bound + short proof

A posteriori, automatic analysis:

- Goal: $\hat{x}$ and enclosure of $\hat{x} - x$ for **given** input and format
- Tool: interval arithmetic based on **floating-point**
- Ideal: a narrow interval computed fast

An example: Kahan's algorithm for *ad-bc*

Kahan's algorithm uses the FMA to evaluate $\det \begin{bmatrix} a & b \\ c & d \end{bmatrix} = ad - bc$:

$$\boxed{\begin{array}{l} \hat{w} := \mathrm{RN}\,(bc); \\ \hat{f} := \mathrm{RN}\,(ad - \hat{w}); \quad e := \mathrm{RN}\,(\hat{w} - bc); \\ \hat{r} := \mathrm{RN}\,(\hat{f} + e); \end{array}}$$

For Kahan's algorithm, Jeannerod-Louvet-Muller (2013) established

$$\frac{|\hat{r} - r|/|r|}{2u} = \frac{1}{1+2u} = 1 - 2u + O(u^2).$$

They exhibit values for a, b, c, d that show that this bound is asymptotically optimal. With a dozen lines of code, it was possible to establish that

$$\frac{(\boldsymbol{r} - r)/r}{2u} \in [0, 3] \text{ for the naive method.}$$

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

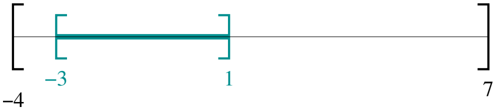
Future

Conclusion

Constraint solving

Can we solve

$$[-3.1] + \boldsymbol{z} = [-4, 7]?$$



$$\boldsymbol{z} = [-4, 7] - [-3, 1] = [-5, 10]$$

is not the solution:

$$[-3, 1] + [-5, 10] = [-8, 11] \neq [-4, 7].$$

A (the) solution exists: $\boldsymbol{z} = [-1, 6]$.

$$\boldsymbol{z} = [-4 - (-3), 7 - 1].$$

Constraint solving

Can we solve

$$\boldsymbol{x} + \boldsymbol{z} = \boldsymbol{y} \quad : \quad [\underline{x}, \overline{x}] + [\underline{z}, \overline{z}] = [\underline{y}, \overline{y}]?$$

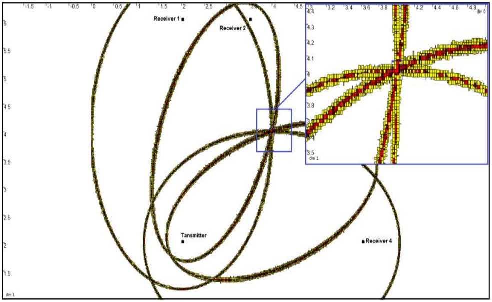
Yes if $\mathrm{width}(\boldsymbol{y}) \geq \mathrm{width}(\boldsymbol{x})$ and the solution is given by

$$\boldsymbol{z} = [\underline{y} - \underline{x}, \overline{y} - \overline{x}].$$

This is neither the addition nor the subtraction of intervals, it is a new operation so that the addition has a reciprocal. It is very useful to solve constraints such as $x + z \geq -4$ and $x + z \leq 7$ with $x \in [-1, 3]$ for instance.

Range-only Multistatic Radar Detection of a Wind farm based on Interval Analysis

Multistatic Radar Range-only Separators:



Courtesy E. Codres, W. Al Mashhadani, A. Brown, A. Stancu and L. Jaulin (SWIM 2016).

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

Algorithm: solving a nonlinear system: Newton

Why a specific iteration for interval computations?

Usual formula:

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

Direct interval transposition:

$$\boldsymbol{x}_{k+1} = \boldsymbol{x}_k - \frac{f(\boldsymbol{x}_k)}{f'(\boldsymbol{x}_k)}$$

Algorithm: solving a nonlinear system: Newton

Why a specific iteration for interval computations?

Usual formula:

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

Direct interval transposition:

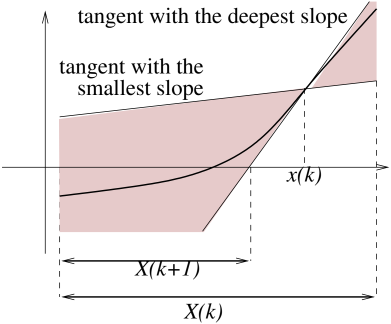
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{f(\mathbf{x}_k)}{f'(\mathbf{x}_k)}$$

Width of the resulting interval:

$$w(\mathbf{x}_{k+1}) = w(\mathbf{x}_k) + w\left(\frac{f(\mathbf{x}_k)}{f'(\mathbf{x}_k)}\right) > w(\mathbf{x}_k)$$

divergence!

Algorithm: interval Newton (Hansen-Greenberg 83, Baker Kearfott 95-97, Mayer 95, van Hentenryck et al. 97)

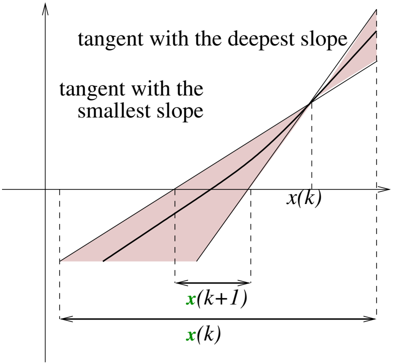


$$\mathbf{x}_{k+1} := \left(x_k - \frac{\mathbf{f}(\{x_k\})}{\mathbf{f}'(\mathbf{x}_k)} \right) \bigcap \mathbf{x}_k$$

Interval Newton: Brouwer theorem

If the new iterate (before intersection) is a subset of the previous iterate, then f has a zero on it.

Furthermore, if it is included in its interior, then this zero is unique.



Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

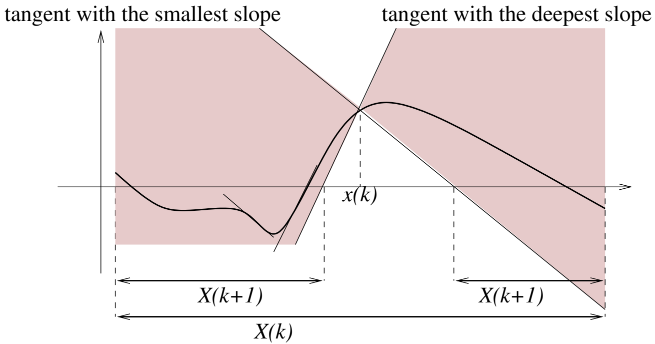
IEEE 1788-2015 standard

Present

Future

Conclusion

Interval Newton and the extended division



$$(\boldsymbol{x}_{k+1,1}, \boldsymbol{x}_{k+1,2}) := \left(x_k - \frac{\boldsymbol{f}(\{x_k\})}{\boldsymbol{f'}(\boldsymbol{x}_k)} \right) \bigcap \boldsymbol{x}_k$$

Agenda

Interval arithmetic in a nutshell
In a nutshell
Historical remarks
Precious features of interval arithmetic
FTIA
Constraint solving
Newton and Brouwer
Newton and the extended division
IEEE 1788-2015 standard
Present
Future
Conclusion

Precious features of interval arithmetic

- **Fundamental theorem of interval arithmetic ("Thou shalt not lie"):** the returned result contains the sought result: abbreviated as **FTIA**;
- **constraint solving:** reverse operations are needed: abbreviated as **CS**;
- **Brouwer theorem:** proof of existence (and uniqueness) of a solution: abbreviated as **Brouwer**;
- **ad hoc, extended division:** gap between two semi-infinite intervals is preserved: abbreviated as **extended division**.

Precious features of interval arithmetic

- **Fundamental theorem of interval arithmetic ("Thou shalt not lie"):** the returned result contains the sought result: abbreviated as FTIA;
- **constraint solving:** reverse operations are needed: abbreviated as CS;
- **Brouwer theorem:** proof of existence (and uniqueness) of a solution: abbreviated as Brouwer;
- **ad hoc, extended division:** gap between two semi-infinite intervals is preserved: abbreviated as extended division.

Goal of a standardization: keep the nice properties, have common definitions.

IEEE P1788 working group

Goal of a standardization: keep the nice properties, have common definitions.

IEEE P1788 working group

Goal of a standardization: keep the nice properties, have common definitions.

Advantage of a standardization: have common definitions, be able to compare algorithms, results, libraries, have common benchmarks and test cases.

IEEE P1788 working group

Creation of the IEEE P1788 project: Initiated by 15 attenders at Dagstuhl, Jan 2008. Standard published in July 2015: IEEE 1788-2015.

How P1788's work is done

- The bulk of work is carried out by email - electronic voting.
- Proposals = motions: 3 weeks discussion; 3 weeks vote.
- IEEE had given us a 4-year deadline + 2 more + 1 more.
- One "in person" meeting per year.
- IEEE auspices: 1 report + 1 teleconference quarterly

IEEE P1788 working group

Creation of the IEEE P1788 project: Initiated by 15 attenders at Dagstuhl, Jan 2008. Standard published in July 2015: IEEE 1788-2015.

How P1788's work is done

- The bulk of work is carried out by email - electronic voting.
- Proposals = motions: 3 weeks discussion; 3 weeks vote.
- IEEE had given us a 4-year deadline + 2 more + 1 more.
- One "in person" meeting per year.
- IEEE auspices: 1 report + 1 teleconference quarterly

URL of the working group:

http://grouper.ieee.org/groups/1788/

or google **1788 interval arithmetic**.

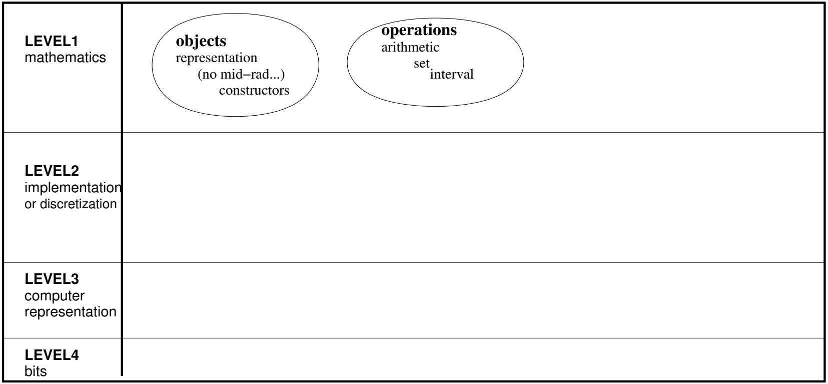
IEEE-1788 standard: the big picture

LEVEL1 mathematics	
LEVEL2 implementation or discretization	
LEVEL3 computer representation	
LEVEL4 bits	

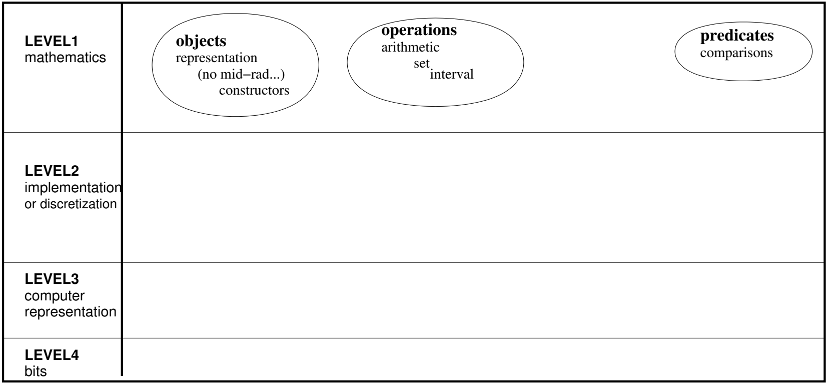
IEEE-1788 standard: the big picture

LEVEL1 mathematics	**objects** representation (no mid–rad...) constructors
LEVEL2 implementation or discretization	
LEVEL3 computer representation	
LEVEL4 bits	

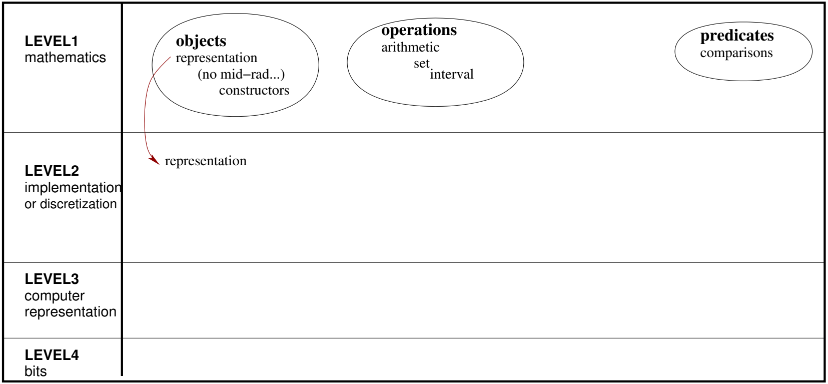
IEEE-1788 standard: the big picture



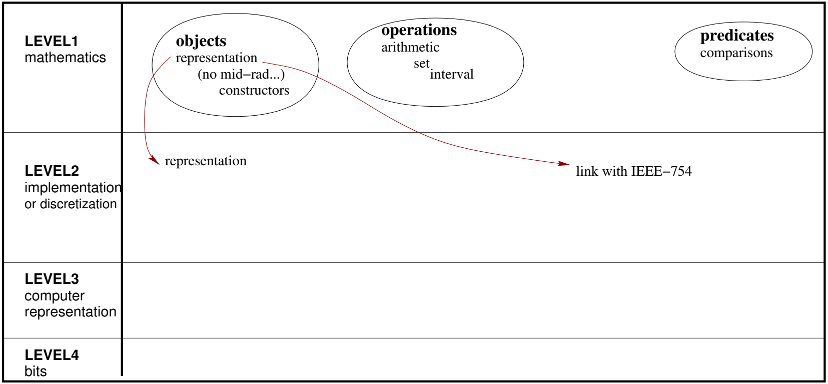
IEEE-1788 standard: the big picture



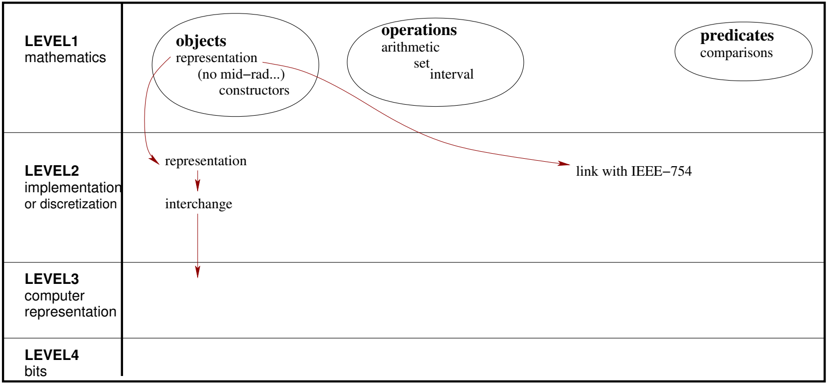
IEEE-1788 standard: the big picture



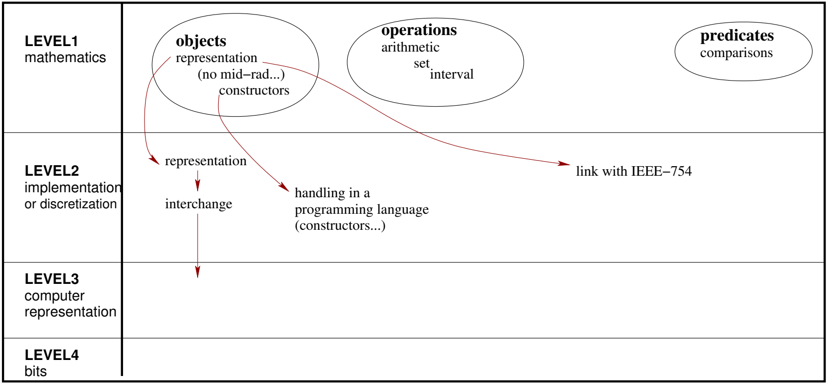
IEEE-1788 standard: the big picture



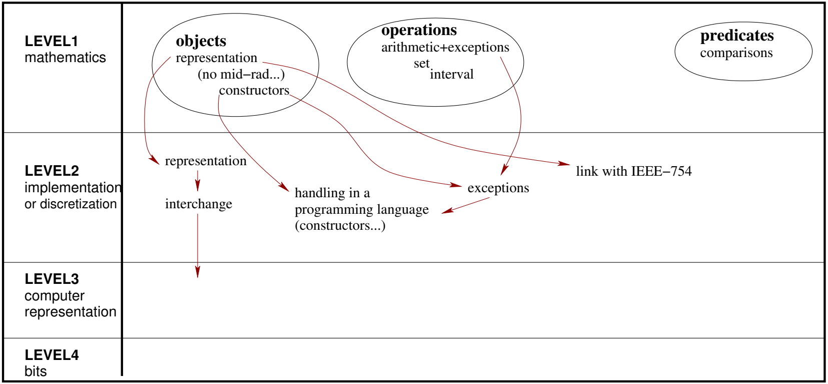
IEEE-1788 standard: the big picture



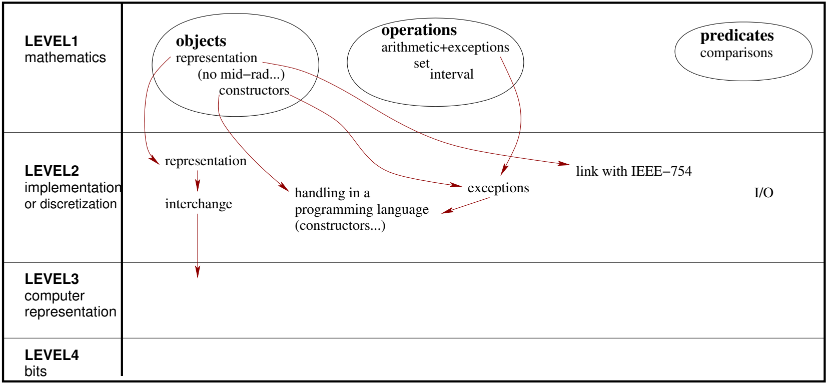
IEEE-1788 standard: the big picture



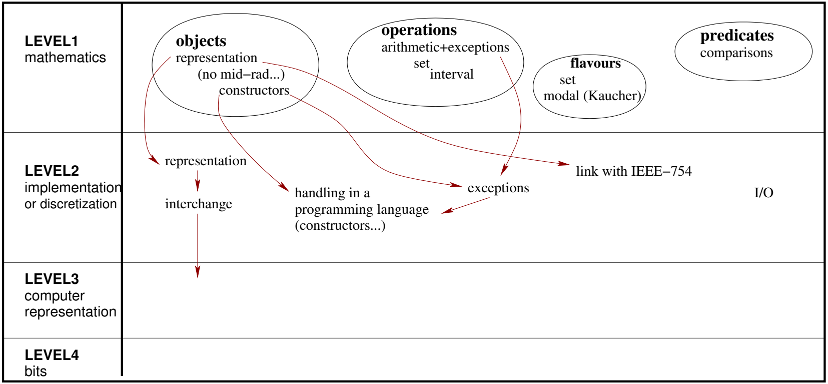
IEEE-1788 standard: the big picture



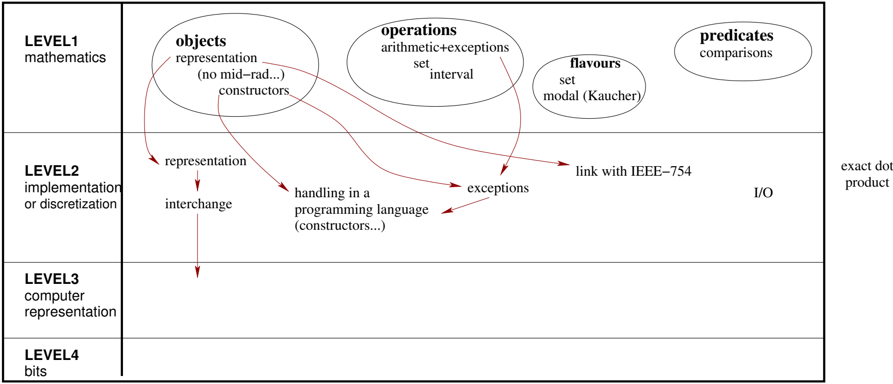
IEEE-1788 standard: the big picture



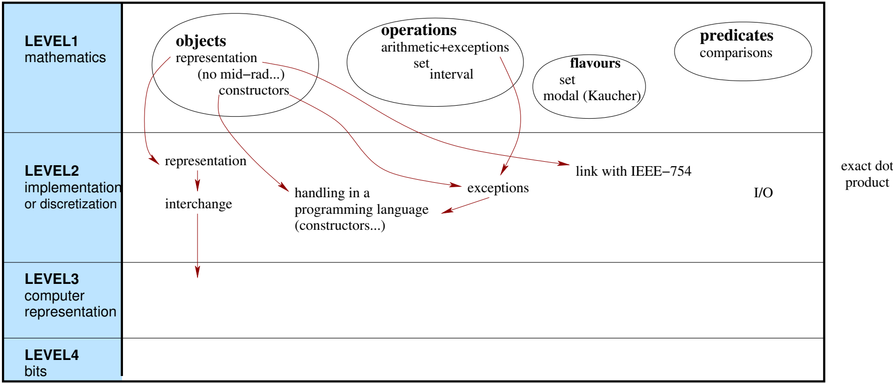
IEEE-1788 standard: the big picture



IEEE-1788 standard: the big picture



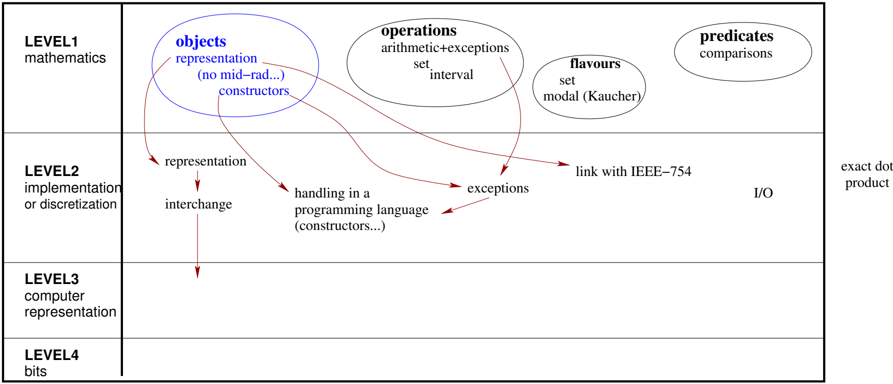
IEEE-1788 standard: levels



IEEE-1788 standard: levels

- **Level 1: mathematical level**
  $[0, \pi]$
- **Level 2: discretization level**
  $[\mathrm{rl}(0), \mathrm{ru}(\pi)]$ where rl and ru map a real number to a number from an abstract finite set; **issues: going from a continuous to a discrete, finite set**;
- **Level 3: computer representation level**
  $[\mathrm{zero}, \mathrm{pi}]$ where $\mathrm{zero} = \mathsf{RD}\,(0)$, $\mathrm{pi} = \mathsf{RU}\,(\pi)$ are two `binary64` numbers: **the discrete finite set is specified**;
- **Level 4: encoding level**
  `0x8000000000000000`
  `0x400921fb54442d19`

IEEE-1788 standard: intervals



IEEE-1788 standard: intervals

Everybody agrees on the meaning of $[1, 3]$.

IEEE-1788 standard: intervals

Everybody agrees on the meaning of $[1, 3]$.

What about $\emptyset$?

IEEE-1788 standard: intervals

Everybody agrees on the meaning of $[1,3]$.

What about $\emptyset$? $[5,+\infty)$?

IEEE-1788 standard: intervals

Everybody agrees on the meaning of $[1, 3]$.

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?

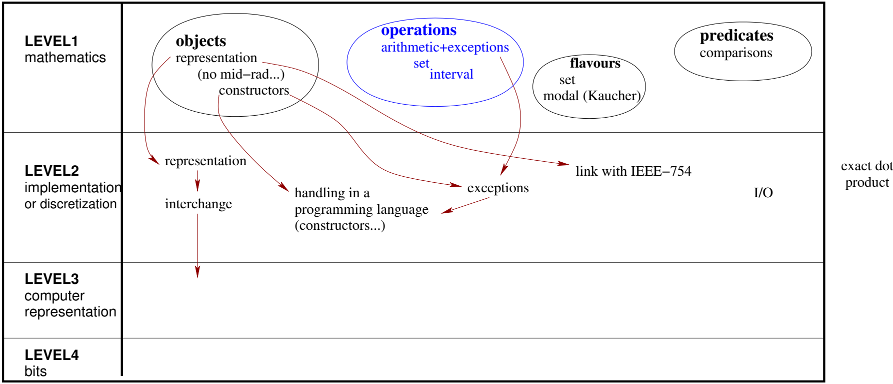
IEEE-1788 standard: intervals

Everybody agrees on the meaning of $[1, 3]$.

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?

Common basis: **an interval is a non-empty bounded closed connected subset of $\mathbb{R}$.**

IEEE-1788 standard: operations



IEEE-1788 standard: operations

Operations are defined so as to satisfy FTIA:

- $[-1, 3] + [4, 8] = [3, 11]$
- $[2, 5] - [1, 2] = [0, 4]$
- $[2, 3] * [-2, 1] = [-6, 3]$
- $[2, 3]/[1, 2] = [1, 3]$
- $\sqrt{[1, 2]} = [1, \sqrt{2}]$
- $\sin([3, 5]) \subset [-1, +0.14113]$
- ...

IEEE-1788 standard: operations

Operations are defined so as to satisfy FTIA:

- $[-1,3]+[4,8]=[3,11]$
- $[2,5]-[1,2]=[0,4]$
- $[2,3]*[-2,1]=[-6,3]$
- $[2,3]/[1,2]=[1,3]$
- $\sqrt{[1,2]}=[1,\sqrt{2}]$
- $\sin([3,5])\subset[-1,+0.14113]$
- ...

Common basis: **an operation φ evaluated on interval arguments $\mathbf{x}_1,\ldots,\mathbf{x}_k$ within its domain returns its range on these arguments (or an enclosure of it)**.

IEEE-1788 standard: operations

Operations specific to sets:

- intersection: $[2, 4] \cap [3, 7] = [3, 4]$
- convex hull of the union: $[-2, -1] \cup [3, 7] = [-2, 7]$

and to intervals:

- infimum, supremum: $\inf([-1, 3]) = -1$, $\sup([-1, 3]) = 3$
- midpoint: $\mathrm{mid}([-1, 3]) = 1$
- width, radius: $\mathrm{wid}([-1, 3]) = 4$, $\mathrm{rad}([-1, 3]) = 2$
- magnitude, mignitude: $\mathrm{mag}([-1, 3]) = 3$, $\mathrm{mig}([-1, 3]) = 0$

IEEE-1788 standard: operations

`cancelPlus` and `cancelMinus` are defined as reverse operations for $+$ and $-$:

- $\texttt{cancelMinus}([2,5],[1,3]) = [1,2]$
- $\texttt{cancelPlus}([2,5],[1,3]) =$
  $\texttt{cancelMinus}([2,5],-[1,3]) = [5,6]$

so as to satisfy **CS**.

IEEE-1788 standard: operations

Operations are defined so as to satisfy **FTIA**:

- $[-1, 3] + [4, 8] = [3, 11]$
- $[2, 5] - [1, 2] = [0, 4]$
- $[2, 3] * [-2, 1] = [-6, 3]$
- $[2, 3]/[1, 2] = [1, 3]$
- $\sqrt{[1, 2]} = [1, \sqrt{2}]$
- $\sin([3, 5]) \simeq [-1, +0.14113]$
- ...

IEEE-1788 standard: operations

Operations are defined so as to satisfy **FTIA**:

- $[-1, 3] + [4, 8] = [3, 11]$
- $[2, 5] - [1, 2] = [0, 4]$
- $[2, 3] * [-2, 1] = [-6, 3]$
- $[2, 3]/[1, 2] = [1, 3]$
- $\sqrt{[1, 2]} = [1, \sqrt{2}]$
- $\sin([3, 5]) \simeq [-1, +0.14113]$
- ...

What about $[2, 3]/[0, 2]$?

IEEE-1788 standard: operations

Operations are defined so as to satisfy FTIA:

- $[-1, 3] + [4, 8] = [3, 11]$
- $[2, 5] - [1, 2] = [0, 4]$
- $[2, 3] * [-2, 1] = [-6, 3]$
- $[2, 3]/[1, 2] = [1, 3]$
- $\sqrt{[1, 2]} = [1, \sqrt{2}]$
- $\sin([3, 5]) \simeq [-1, +0.14113]$
- ...

What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$?

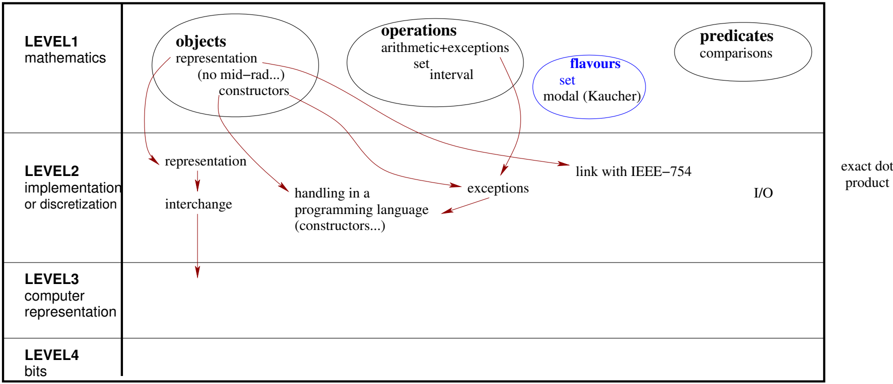
IEEE-1788 standard: operations

Operations are defined so as to satisfy **FTIA**:

- $[-1, 3] + [4, 8] = [3, 11]$
- $[2, 5] - [1, 2] = [0, 4]$
- $[2, 3] * [-2, 1] = [-6, 3]$
- $[2, 3]/[1, 2] = [1, 3]$
- $\sqrt{[1, 2]} = [1, \sqrt{2}]$
- $\sin([3, 5]) \simeq [-1, +0.14113]$
- ...

What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

IEEE-1788 standard: flavors



IEEE-1788 standard: flavors

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic but not in all of them, and not the same.

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic but not in all of them, and not the same.

IEEE 1788-2015 defines a common basis (seen so far) and provides "hooks" to accommodate a theory,

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic but not in all of them, and not the same.

IEEE 1788-2015 defines a common basis (seen so far) and provides "hooks" to accommodate a theory, called a **flavor**.

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic but not in all of them, and not the same.

IEEE 1788-2015 defines a common basis (seen so far) and provides "hooks" to accommodate a theory, called a **flavor**.

The only available flavor in IEEE 1788-2015 is the **set-based flavor**.

IEEE-1788 standard: flavors

What about $\emptyset$? $[5, +\infty)$? $[3, 1]$?
What about $[2, 3]/[0, 2]$? $[2, 3]/[-1, 2]$? $\sqrt{[-1, 2]}$?

Each has a meaning in some theory: set theory, cset theory, Kaucher arithmetic, modal arithmetic but not in all of them, and not the same.

IEEE 1788-2015 defines a common basis (seen so far) and provides "hooks" to accommodate a theory, called a **flavor**.

The only available flavor in IEEE 1788-2015 is the **set-based flavor**. Any implementation is IEEE 1788-2015 compliant if it provides at least one flavor.

IEEE-1788 standard: set-based flavor

Set-based interval: a ~~non-empty bounded~~ closed connected subset of $\mathbb{R}$. Ex.: $\emptyset$ $\quad [5, +\infty)$ $\quad \cancel{[3, 1]}$

IEEE-1788 standard: set-based flavor

Set-based interval: a ~~non-empty bounded~~ closed connected subset of $\mathbb{R}$. Ex.: $\emptyset$ $[5, +\infty)$ $\cancel{[3, 1]}$

Set-based operation:

$$\varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) = \mathrm{Hull}\{\varphi(x_1, \ldots, x_k) : (x_1, \ldots, x_k) \in (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k) \cap \mathrm{Dom}(\varphi)\}.$$

$$[2, 3]/[0, 2] = [1, +\infty), \quad [2, 3]/[-1, 2] = (-\infty, +\infty), \quad \sqrt{[-1, 2]} = [0, \sqrt{2}].$$

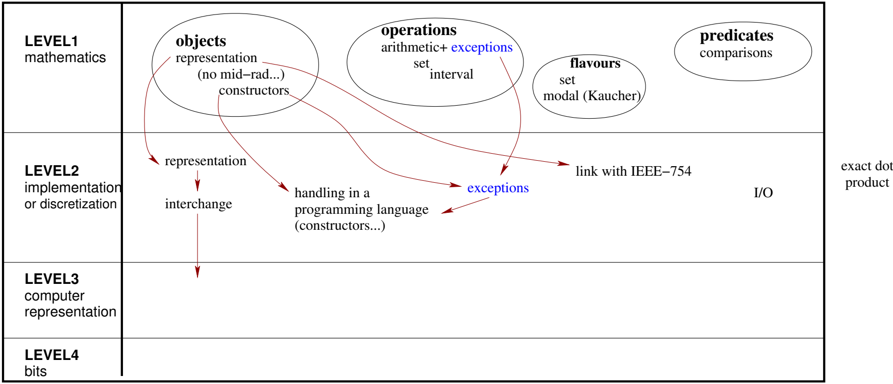
IEEE-1788 standard: set-based flavor

Some more operations, specific to the set-based flavor:

- more reverse operations: for $|.|$, sqr, sin. . . .
  Ex.: $\mathrm{sqrRev}([1,4]) = \mathrm{Hull}([-2,-1] \cup [1,2]) = [-2,2]$,
  $\mathrm{sinRev}([-0.3, 0.5], [3\pi, 5\pi]) \subset [9.4247, 15.708]$.

- `mulRevToPair` corresponds to the extended division:
  $[2,3]/[-1,2] = ((-\infty, -2], [1, +\infty))$.

IEEE-1788 standard: decorations



IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.

IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

Brouwer theorem does not apply because f is not continuous on $[-4, 9]$.

IEEE-1788 standard: decorations

Let $f \,:\, x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

Brouwer theorem does not apply because f is not continuous on $[-4, 9]$. f is not everywhere defined on $[-4, 9]$.

IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

Brouwer theorem does not apply because f is not continuous on $[-4, 9]$. f is not everywhere defined on $[-4, 9]$.

Required: provide a means to tell the user that Brouwer theorem does not apply:

IEEE-1788 standard: decorations

Let $f : x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

Brouwer theorem does not apply because f is not continuous on $[-4, 9]$. f is not everywhere defined on $[-4, 9]$.

Required: provide a means to tell the user that Brouwer theorem does not apply: handle an exception.
This satisfies our requirement regarding Brouwer.

IEEE-1788 standard: decorations

Let $f\ :\ x \mapsto \sqrt{x} - 2$.
f has no real fixed point $\Leftrightarrow g(x) = x - \sqrt{x} + 2$ has no real zero.

Evaluate $f([-4, 9]) = \sqrt{[-4, 9]} - 2 = [0, 3] - 2 = [-2, 1] \subset [-4, 9]$.

Brouwer theorem does not apply because f is not continuous on $[-4, 9]$. f is not everywhere defined on $[-4, 9]$.

Required: provide a means to tell the user that Brouwer theorem does not apply: handle an exception.
This satisfies our requirement regarding Brouwer.

Mechanism for handling exception: **decoration**.

IEEE-1788 standard: decorations

Best way to handle exceptions?

To avoid global flags, "tags" attached to each interval: decorations.

Meaning (in the set-based flavor): piece of information regarding the history, the computation that led to the interval.

Discussions about what should be in the decorations.

For set-based flavor:

- `com` for common,
- `dac` for defined and continuous,
- `def` for defined,
- `trv` for no information (trivial),
- `ill` for nowhere defined, or ill-formed.

IEEE-1788 standard: decorations

Propagation rule (for the set-based flavor):

$$(\boldsymbol{y}, d_y) = \varphi((\boldsymbol{x}_1, d_1), \ldots, (\boldsymbol{x}_k, d_k))$$

where

- $\boldsymbol{y} = \varphi(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k)$;
- d is a decoration corresponding to the application of φ to $\boldsymbol{x}_1, \ldots, \boldsymbol{x}_k$;
- $d_y = \min(d, d_1, \ldots, d_k)$ where the order is $\texttt{com} > \texttt{dac} > \texttt{def} > \texttt{trv} > \texttt{ill}$.

IEEE-1788 standard: decorations

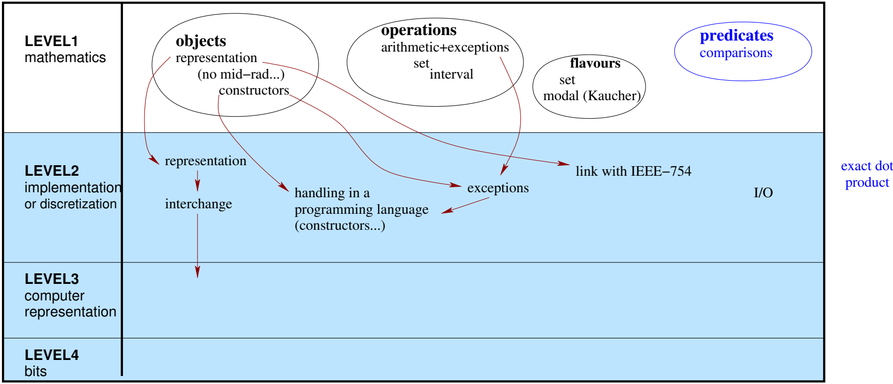
Mandatory: every flavor must provide a FTDIA: Fundamental Theorem of Decorated Interval Arithmetic.

IEEE-1788 standard: decorations

Mandatory: every flavor must provide a FTDIA: Fundamental Theorem of Decorated Interval Arithmetic.

FTDIA for the set-based flavor: the FTIA holds and the decorations mean what they are meant to mean.

IEEE-1788 standard: misc



IEEE-1788 standard: comparisons

How to compare two intervals?

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

IEEE-1788 standard: comparisons

How to compare two intervals?

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

- **7 relations:** equal ($=$), subset ($\subset$), less than or equal to ($\leq$), precedes or touches ($\preceq$), interior to, less than ($<$), precedes ($\prec$).

IEEE-1788 standard: comparisons

How to compare two intervals?

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

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

IEEE-1788 standard: exact dot product

Recommended operation: exact dot product or **edp**.
This operation concerns vectors of floating-point numbers, not vectors of intervals.

IEEE-1788 standard: levels

Level 2 issues:

- rounding: at Level 1, $\boldsymbol{x} = [\underline{x}, \overline{x}]$ – at Level 2, $\boldsymbol{x}$ is represented as $[\,\mathrm{RD}\,(\underline{x}),\, \mathrm{RU}\,(\overline{x})]$;
- similar issue for the result of each operation;
- cornercases: $\mathrm{wid}(\emptyset)$? $\mathrm{mid}(\mathbb{R})$? By convention: NaN, or 0...
- representation: by endpoints, by midpoint-radius
- constructors.

Level 3 issues:

- issues mostly related to IEEE 754-2008.

Level 4 issues:

- issues mostly related to IEEE 754-2008;
- encoding of decorations is specified.

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

IEEE 1788.1-2017: Standard for Interval Arithmetic (simplified)

This standard is a simplified version and a subset of the IEEE Std 1788-2015 for Interval Arithmetic and includes those operations and features of the latter that in the the editors' view are most commonly used in practice. IEEE P1788.1 specifies interval arithmetic operations based on intervals whose endpoints are IEEE Std 754-2008 binary64 floating-point numbers and a decoration system for exception-free computations and propagation of properties of the computed results.

A program built on top of an implementation of IEEE P1788.1 should compile and run, and give identical output within round off, using an implementation of IEEE Std 1788-2015, or any superset of the former.

Compared to IEEE Std 1788-2015, this standard aims to be minimalistic, yet to cover much of the functionality needed for interval computations. As such, it is more accessible and will be much easier to implement, and thus will speed up production of implementations.

IEEE 1788-2015 compliant libraries

A compliant library implements common interval arithmetic and at least one flavor.

IEEE 1788-2015 compliant libraries

A compliant library implements common interval arithmetic and at least one flavor.

The IEEE 1788-2015 compliant libraries have been developed after the standard, as proof-of-concept mostly:

- **Octave interval**
- **libieee1788**
- jInterval : Java interface for several libraries + compliance-tests

Agenda

Interval arithmetic in a nutshell
In a nutshell
Historical remarks
Precious features of interval arithmetic
FTIA
Constraint solving
Newton and Brouwer
Newton and the extended division
IEEE 1788-2015 standard
Present
Future
Conclusion

IEEE 1788-2015 compliant libraries

The IEEE 1788-2015 compliant libraries have been developed after the standard, as proof-of-concept mostly:

- **Octave interval** by Oliver Heimlich
- **libieee1788** by Marco Nehmeier
- jInterval : Java interface for several libraries + compliance-tests: Dmitry Nadezhin, Sergei Zhilin

IEEE 1788-2015 compliant libraries

The IEEE 1788-2015 compliant libraries have been developed after the standard, as proof-of-concept mostly:

- Octave interval by Oliver Heimlich
- libieee1788 by Marco Nehmeier
- jInterval : Java interface for several libraries + compliance-tests: Dmitry Nadezhin, Sergei Zhilin

Both developers/maintainers have left academia: what is the future of these libraries?

New flavors?

Flavors mentioned during the development of IEEE 1788-2015:

- cset
- Kaucher arithmetic
- modal arithmetic
- Rump's proposal to handle overflow

New flavors?

Flavors mentioned during the development of IEEE 1788-2015:

- cset
- Kaucher arithmetic
- modal arithmetic
- Rump's proposal to handle overflow

Any person proposing a new flavor should submit it as a revision of the standard.

New flavors?

Flavors mentioned during the development of IEEE 1788-2015:

- cset
- Kaucher arithmetic
- modal arithmetic
- Rump's proposal to handle overflow

Any person proposing a new flavor should submit it as a revision of the standard.

Anyway, the standard will incur a revision for 2025.

Agenda

Interval arithmetic in a nutshell
- In a nutshell
- Historical remarks

Precious features of interval arithmetic
- FTIA
- Constraint solving
- Newton and Brouwer
- Newton and the extended division

IEEE 1788-2015 standard

Present

Future

Conclusion

Why do I like working on interval arithmetic?

Why do I like working on interval arithmetic?

because there is plenty of research to do. . .

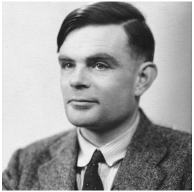
Why do I like working on interval arithmetic?

because there is plenty of research to do...

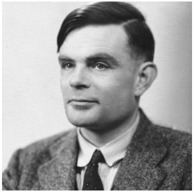
because interval arithmetic is magic!

Magic interval arithmetic

Alan Turing (23-06-1912 - 07-06-1954)



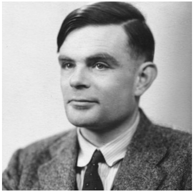
Magic interval arithmetic



Alan Turing (23-06-1912 - 07-06-1954)

- number between 1 and 100 $\in [1, 100]$

Magic interval arithmetic

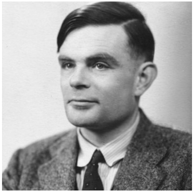


Alan Turing (23-06-1912 - 07-06-1954)

- number between 1 and 100 $\in [1, 100]$
- multiply by birth day: $\in [1, 31]$ $\in [3, 3100]$

Magic interval arithmetic

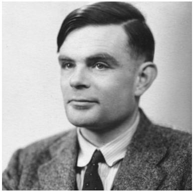
Alan Turing (23-06-1912 - 07-06-1954)



- number between 1 and 100 $\in [1, 100]$
- multiply by birth day: $\in [1, 31]$ $\in [3, 3100]$
- divide by 8xxx: $\in [8000, 8999]$ $\in (0, 0.4)$

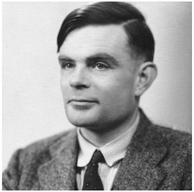
Magic interval arithmetic

Alan Turing (23-06-1912 - 07-06-1954)



- number between 1 and 100 $\in [1, 100]$
- multiply by birth day: $\in [1, 31]$ $\in [3, 3100]$
- divide by 8xxx: $\in [8000, 8999]$ $\in (0, 0.4)$
- subtract 241 $\in (-241, -240.6)$

Magic interval arithmetic



Alan Turing (23-06-1912 - 07-06-1954)

- number between 1 and 100 $\in [1, 100]$
- multiply by birth day: $\in [1, 31]$ $\in [3, 3100]$
- divide by 8xxx: $\in [8000, 8999]$ $\in (0, 0.4)$
- subtract 241 $\in (-241, -240.6)$
- divide by 4 $\in (-60.25, -60.15)$