IEEE-1788 standardization of interval arithmetic: introduction - link with MPFI

Nathalie Revol
INRIA - Université de Lyon
LIP (UMR 5668 CNRS - ENS Lyon - INRIA - UCBL)

Third MPFR-MPC Developers Meeting
Nancy, 20-22 January 2014

Agenda

Agenda

Agenda

A brief introduction

Interval arithmetic:
instead of numbers, use intervals and compute.

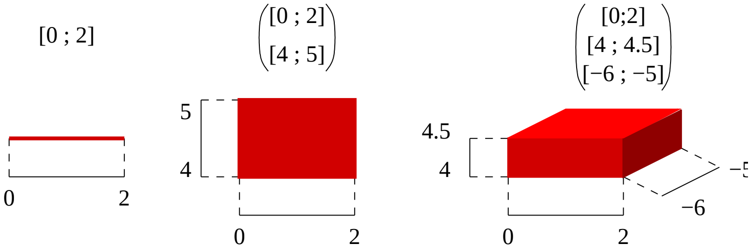
Fundamental theorem of interval arithmetic:
(or "Thou shalt not lie") (or "Inclusion property"):
the exact result (number or set) is contained in the computed interval.

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

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

$$\mathbf{x} \diamond \mathbf{y} = \mathbf{Hull}\{x \diamond y \;:\; x \in \mathbf{x}, y \in \mathbf{y}\}$$

Arithmetic and algebraic operations: use the monotonicity

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

Definitions: functions

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

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

Elementary functions: again, use the monotonicity.

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

Interval arithmetic:
implementation using floating-point arithmetic

Implementation using floating-point arithmetic:
use directed rounding modes (cf. IEEE 754 standard)

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

Definitions: function extension

$f(x) = x^2 - x + 1 = x(x-1) + 1 = (x - 1/2)^2 + 3/4$ **on** $[-2, 1]$.

Using $x^2 - x + 1$, one gets
$[-2, 1]^2 - [-2, 1] + 1 = [0, 4] + [-1, 2] + 1 = [0, 7]$.

Using $x(x-1) + 1$, one gets
$[-2, 1] \cdot ([-2, 1] - 1) + 1 = [-2, 1] \cdot [-3, 0] + 1 = [-3, 6] + 1 = [-2, 7]$.

Using $(x - 1/2)^2 + 3/4$, one gets
$([-2, 1] - 1/2)^2 + 3/4 = [-5/2, 1/2]^2 + 3/4 = [0, 25/4] + 3/4 =$
$[3/4, 7] = f([-2, 1])$.

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

How to choose the best extension? A good one?

Agenda

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:

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

Cons: overestimation (2/2)

Wrapping effect

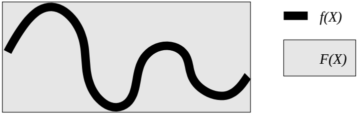
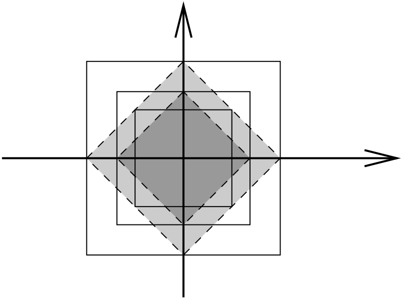


image of $f(\mathbf{x})$
with $f : \mathbf{R}^2 \rightarrow \mathbf{R}^2$



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

Cons: complexity and efficiency

Complexity: most problems are NP-hard (Gaganov, Rohn, Kreinovich. . .)

- evaluate a function on a box. . . even up to ε
- solve a linear system. . . even up to $1/4n^4$
- determine if the solution of a linear system is bounded

Efficiency

Implementation using floating-point arithmetic:
use directed roundings, towards $\pm\infty$.

Overhead in execution time:
in theory, at most 4, or 8, cf.

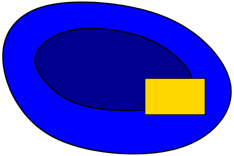
$$[\underline{x}, \overline{x}] \times [\underline{y}, \overline{y}] = [\ \min(RD(\underline{x} \times \underline{y}), RD(\underline{x} \times \overline{y}), RD(\overline{x} \times \underline{y}), RD(\overline{x} \times \overline{y})), \max(RU(\underline{x} \times \underline{y}), RU(\underline{x} \times \overline{y}), RU(\overline{x} \times \underline{y}), RU(\overline{x} \times \overline{y}))$$

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

Pros: set computing

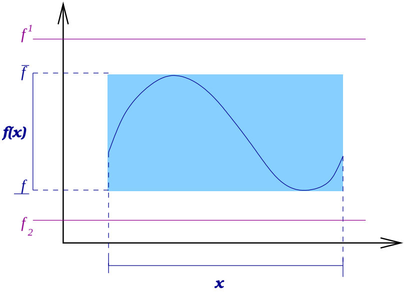
Computing with whole sets or with sets enclosing uncertainties.

Behaviour safe? controllable? dangerous?



always controllable.

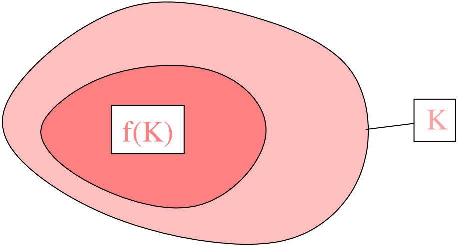
On **x**, are the extrema of the function f $> f^1$, $< f_2$?



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

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 $f(B) \subset \text{int}B$ 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

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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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!

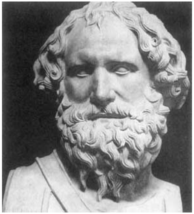
Cf. `http://www.cs.utep.edu/interval-comp/`, click on *Early papers by Others*.

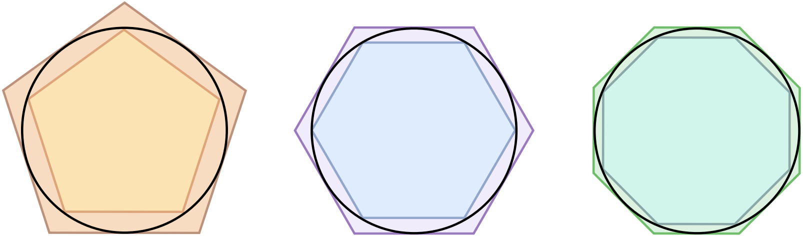
Who invented Interval Arithmetic?

- **1962:** Ramon Moore defines IA in his PhD thesis and then a rather exhaustive study of IA in a book in 1966
- **1958:** Tsunaga (MSc in Japanese)&Kantorovitch (in Russian)
- **1956:** Warmus
- **1951:** Dwyer, in the specific case of closed intervals
- **1931:** Rosalind Cecil Young in her PhD thesis in Cambridge (UK) has used some formulas
- **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 interval around π





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 (?).

IEEE-1788 standard for interval arithmetic in 2014? I hope so...

Precious features

- **Fundamental theorem of interval arithmetic ("Thou shalt not lie"):** the returned result contains the sought result;
- **Brouwer theorem:** proof of existence (and uniqueness) of a solution;
- **ad hoc division:** gap between two semi-infinite intervals is preserved.

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

Creation of the IEEE P1788 project: Initiated by 15 attenders at Dagstuhl, Jan 2008. Project authorised as IEEE-WG-P1788, Jun 2008.

Precious features

- **Fundamental theorem of interval arithmetic ("Thou shalt not lie"):** the returned result contains the sought result;
- **Brouwer theorem:** proof of existence (and uniqueness) of a solution;
- **ad hoc division:** gap between two semi-infinite intervals is preserved.

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

Creation of the IEEE P1788 project: Initiated by 15 attenders at Dagstuhl, Jan 2008. Project authorised as IEEE-WG-P1788, Jun 2008.

Precious features

- **Fundamental theorem of interval arithmetic ("Thou shalt not lie"):** the returned result contains the sought result;
- **Brouwer theorem:** proof of existence (and uniqueness) of a solution;
- **ad hoc division:** gap between two semi-infinite intervals is preserved.

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

Creation of the IEEE P1788 project: Initiated by 15 attenders at Dagstuhl, Jan 2008. Project authorised as IEEE-WG-P1788, Jun 2008.

Agenda

Agenda

How P1788's work is done

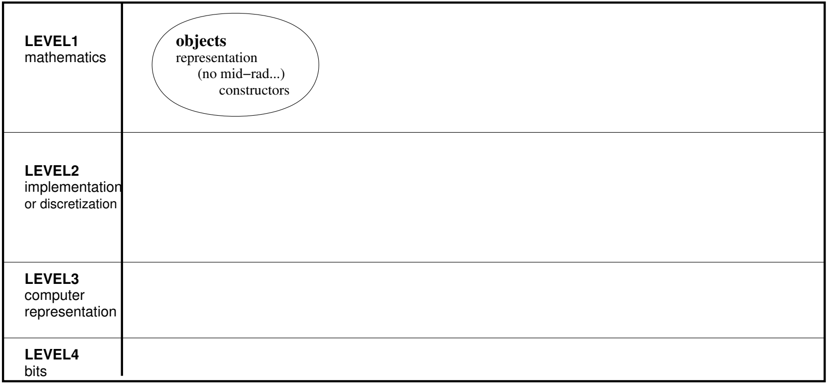
- The bulk of work is carried out by email - electronic voting.
- Motions are proposed, seconded; three weeks discussion period; three weeks voting period.
- IEEE has given us a $4 + 2$ year deadline. . . which expires end 2014.
- One "in person" meeting per year is planned — during the IFSA-NAFIPS conference in 2013 — next one during SCAN 2014.
- IEEE auspices: 1 report + 1 teleconference quarterly

URL of the working group:
`http://grouper.ieee.org/groups/1788/`
or google **1788 interval arithmetic**.

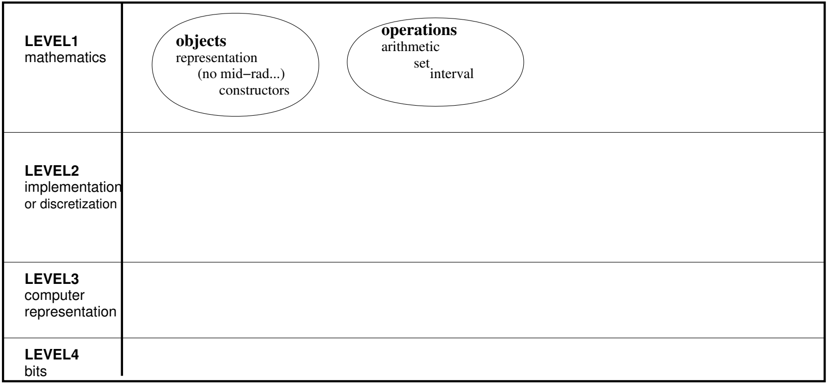
IEEE-1788 standard: the big picture

LEVEL1 mathematics	
LEVEL2 implementation or discretization	
LEVEL3 computer representation	
LEVEL4 bits	

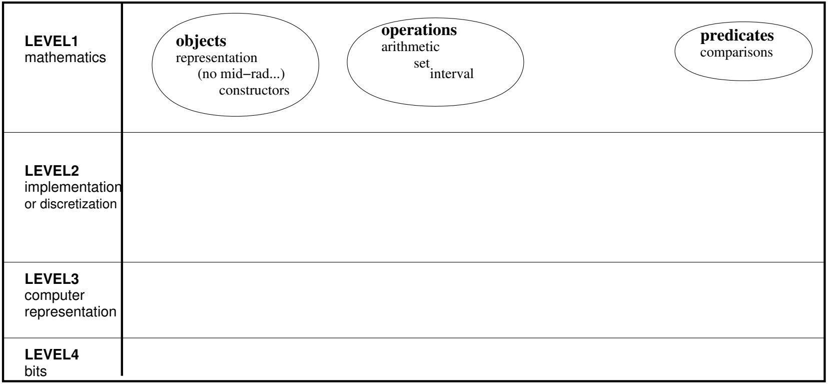
IEEE-1788 standard: the big picture



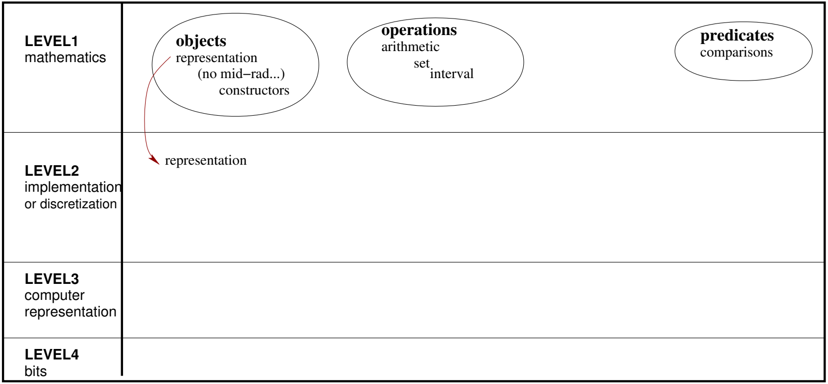
IEEE-1788 standard: the big picture



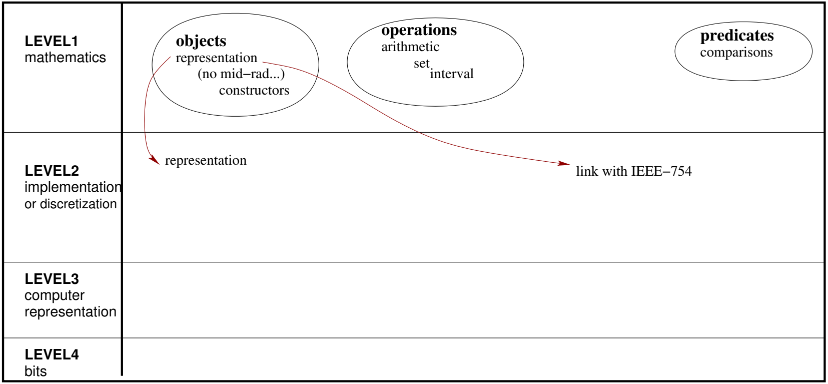
IEEE-1788 standard: the big picture



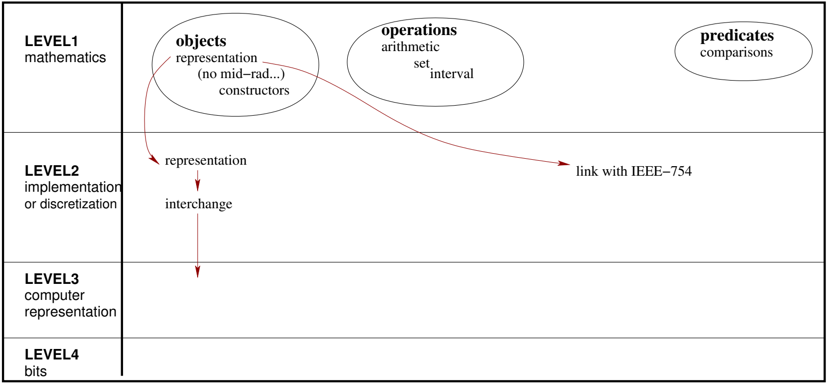
IEEE-1788 standard: the big picture



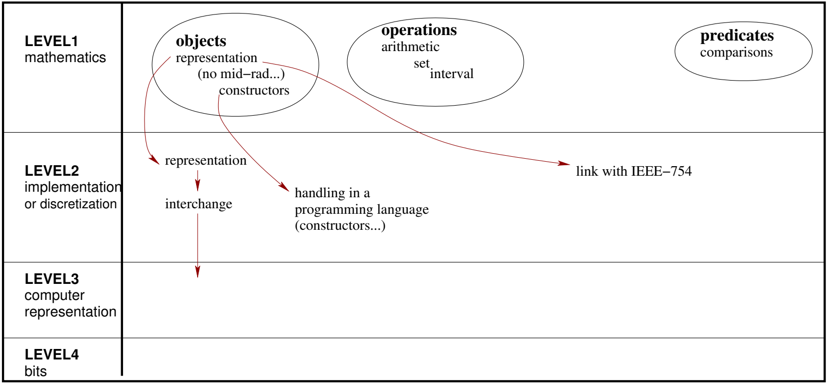
IEEE-1788 standard: the big picture



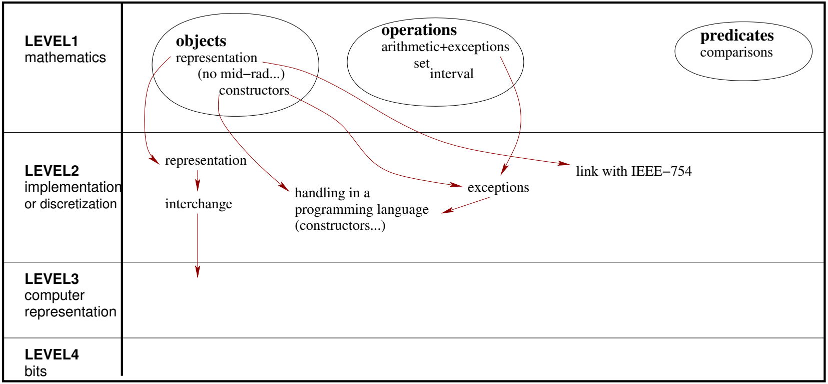
IEEE-1788 standard: the big picture



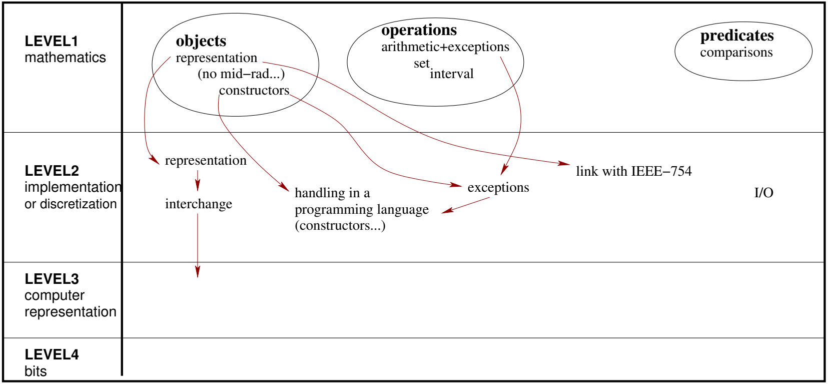
IEEE-1788 standard: the big picture



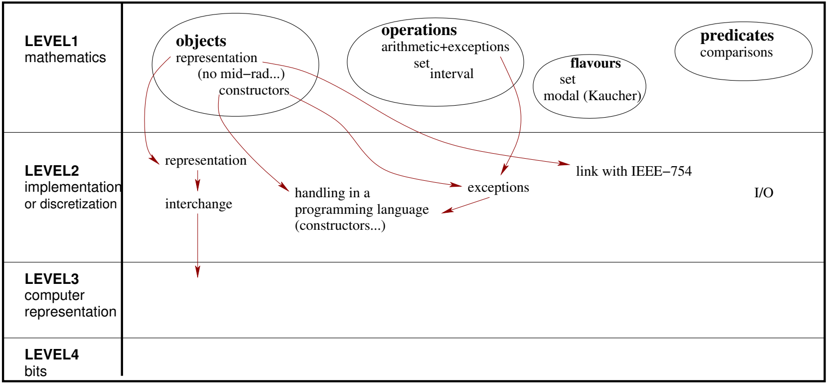
IEEE-1788 standard: the big picture



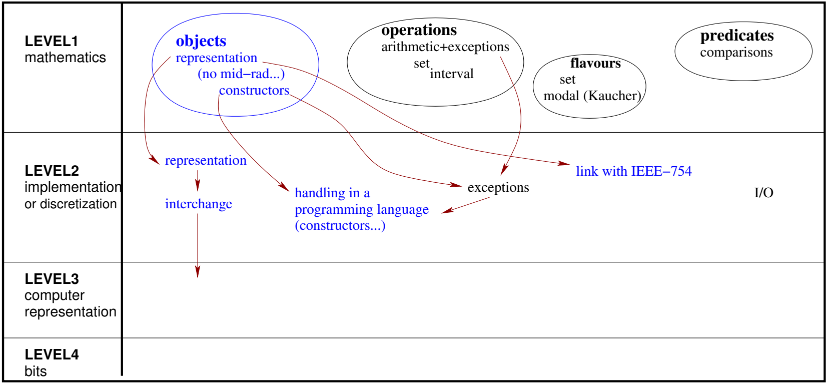
IEEE-1788 standard: the big picture



IEEE-1788 standard: the big picture



IEEE-1788 standard: the big picture



IEEE-1788 standard: intervals

Representation (some intervals, like $\emptyset$, may need a special representation):

- **adopted representation: by the bounds (inf-sup)**;
- other representations could be:
  - by the midpoint m and the radius r (**mid-rad**), $[\underline{x}, \overline{x}] = [m - r, m + r]$;
  - by a triple $\langle x_0, \underline{e}, \overline{e} \rangle$ (**triplex**): $[\underline{x}, \overline{x}] = x_0 + [\underline{e}, \overline{e}] = [x_0 + \underline{e}, x_0 + \overline{e}]$.

IEEE-1788 standard: intervals

Constructors:

- On the implementation (e.g. language) side: conversion of a floating-point number into an interval, e.g. `num2interval(0.1)`, where 0.1 is a decimal floating-point constant in the binary64 type?

  If c denotes the value of this constant (a binary64 number), normally 0.1 rounded to nearest, the answer would be $[c, c]$, not $[\bigtriangledown(0.1), \bigtriangleup(0.1)]$ (which would be expected by the average user).

  $\Rightarrow$ **Containment property not satisfied!**

IEEE-1788 standard: intervals

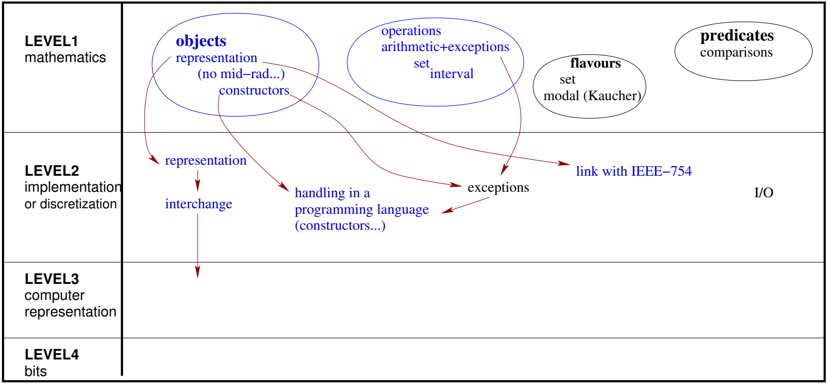
Constructors:

- On the implementation (e.g. language) side: conversion of a floating-point number into an interval, e.g. `num2interval(0.1)`, where 0.1 is a decimal floating-point constant in the binary64 type?

  If c denotes the value of this constant (a binary64 number), normally 0.1 rounded to nearest, the answer would be $[c, c]$, not $[\bigtriangledown(0.1), \bigtriangleup(0.1)]$ (which would be expected by the average user).

  $\Rightarrow$ **Containment property not satisfied!**

IEEE-1788 standard: the big picture



IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0, \sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0,\sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0,\sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0,\sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0,\sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: arithmetic operations

At level 1:

- $+$, $-$, $\times$: everybody agrees
- $/$: $[2,3]/[-1,2]$: 2 semi-infinites, $\mathbb{R}$? $\mathbb{R}$ for closedness
- $\sqrt{[-1,2]}$? $[0,\sqrt{2}]$.

no NaI but exception.

At level 2:
no NaI but exception?
The revenge of the NaI?

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? `MaxReal` in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? MaxReal in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? MaxReal in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? MaxReal in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? `MaxReal` in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? `MaxReal` in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

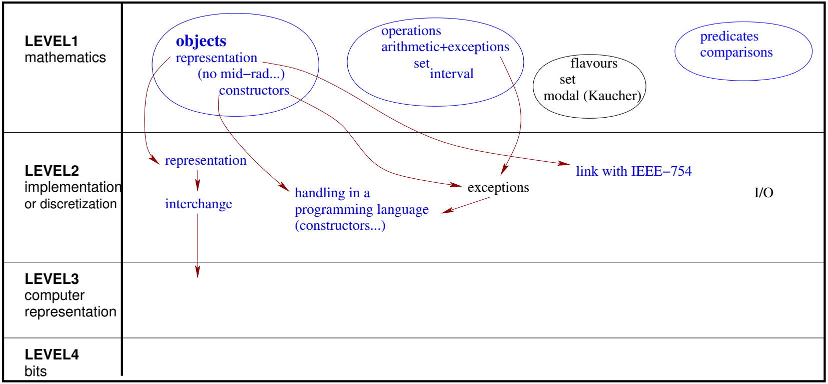
IEEE-1788 standard: operations on intervals

Example: midpoint

What is the midpoint of

- $\mathbb{R}$? 0?
- $[2, +\infty)$? `MaxReal` in $\mathbb{F}$? (at Level 2)
- $\emptyset$? ???

IEEE-1788 standard: the big picture



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 ($\prec$).

  Difficulties: relations defined by conditions on the bounds, but some errors with infinite bounds.
  Special rules... and lack of consistency for the empty set.

  No set-theoretic/topological definitions.

- **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 ($\prec$).

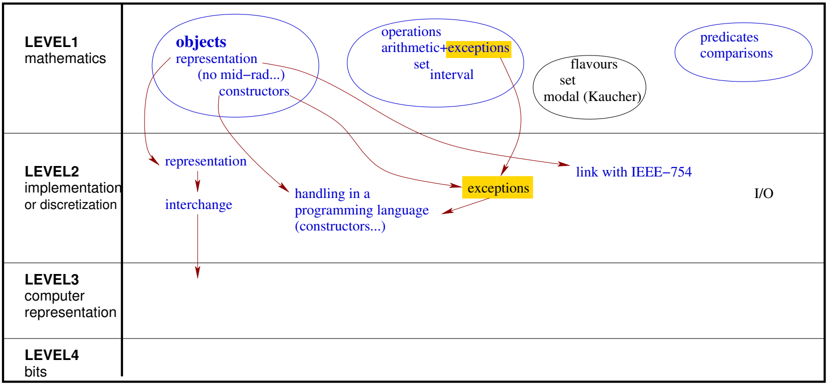
  Difficulties: relations defined by conditions on the bounds, but some errors with infinite bounds.
  Special rules... and lack of consistency for the empty set.

  No set-theoretic/topological definitions.

- **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: the big picture



IEEE-1788 standard: exception handling

Exceptions must be handled in some way, even if exceptions do look. . . exceptional. (It must have been the same for exception handling in IEEE-754 floating-point arithmetic.)

Best way to handle exceptions? To avoid global flags, “tags” attached to each interval: decorations.

Decorated intervals 

Discussions about what should be in the decorations (defined and continuous, defined, no-information, nowhere defined).

IEEE-1788: exception raised by a constructor

On the mathematical model & implementation sides:
`nums2interval(2,1)`, i.e. $\{x \in \mathbb{R} \mid 2 \leq x \leq 1\}$?

Empty? Exception (error)? Notion of Not-an-Interval ("NaI")?
Meaningful in Kaucher arithmetic: let this possibility open.

IEEE-1788: exception raised by a constructor

On the mathematical model & implementation sides:
`nums2interval(2,1)`, i.e. $\{x \in \mathbb{R} \mid 2 \leq x \leq 1\}$?

Empty? Exception (error)? Notion of Not-an-Interval ("NaI")?
Meaningful in Kaucher arithmetic: let this possibility open.

IEEE-1788: exception raised by a constructor

On the mathematical model & implementation sides:
`nums2interval(2,1)`, i.e. $\{x \in \mathbb{R} \mid 2 \leq x \leq 1\}$?

Empty? Exception (error)? Notion of Not-an-Interval ("NaI")?
Meaningful in Kaucher arithmetic: let this possibility open.

IEEE-1788: arguments outside the domain

How should $f(\mathbf{x})$ be handled when $\mathbf{x}$ is not included in the domain of f? E.g. $\sqrt{[-1, 2]}$?

- exit?
- return NaI (Not an Interval)? Ie. handle exceptional values such as NaI and infinities?
- return the set of every possible limits $\lim_{y \to x} f(y)$ for every possible x in the domain of f (but not necessarily y)?
- intersect $\mathbf{x}$ with the domain of f prior to the computation, silently?
- intersect $\mathbf{x}$ with the domain of f prior to the computation and mention it

Remark: arguments outside the domain

(Rump, Dagstuhl seminar 09471, 2009)

$$f(x) = |x-1|$$
$$g(x) = (\sqrt{x-1})^2$$

I know, it is not the best way of writing it...

What happens if $\mathbf{x} = [0, 1]$?

$$f(\mathbf{x}) = [0, 1]$$
$$g(\mathbf{x}) = [0]$$

Without exception handling, **the Thou shalt not lie principle is not valid.**

One has to check whether there has been a *possibly undefined* operation... Unexperienced programmers will not do it.

In other words, it is not possible to protect people from getting wrong results, however hard we try.

Remark: arguments outside the domain

(Rump, Dagstuhl seminar 09471, 2009)

$$\begin{aligned} f(x) &= |x-1| \\ g(x) &= (\sqrt{x-1})^2 \end{aligned}$$

I know, it is not the best way of writing it...

What happens if $\mathbf{x} = [0, 1]$?

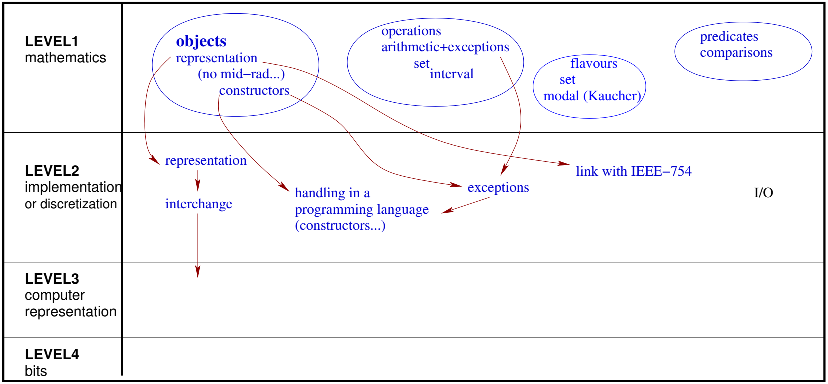
$$\begin{aligned} f(\mathbf{x}) &= [0, 1] \\ g(\mathbf{x}) &= [0] \end{aligned}$$

Without exception handling, **the Thou shalt not lie principle is not valid.**

One has to check whether there has been a *possibly undefined* operation... Unexperienced programmers will not do it.

In other words, it is not possible to protect people from getting wrong results, however hard we try.

IEEE-1788 standard: the big picture



Flavors: Moore's classic IA

Only non-empty and bounded intervals.

Discarded because useful intervals are missing.

Flavors: Moore's classic IA

Only non-empty and bounded intervals.

Discarded because useful intervals are missing.

Flavors: set-based IA

Most famous theory, sound.

Adopted.

Flavors: set-based IA

Most famous theory, sound.

Adopted.

Flavors: Kaucher. modal

$[1, 2]$ and $[2, 1]$ are allowed.
Unbounded intervals are not permitted.

Hook provided, but theory still missing.

Flavors: Kaucher. modal

$[1, 2]$ and $[2, 1]$ are allowed.
Unbounded intervals are not permitted.

Hook provided, but theory still missing.

Flavors: cset...

Cset: $1/[0,1] = \mathbb{R}$.
Conservative: $\sqrt{[-2,1]} =$ NaI?

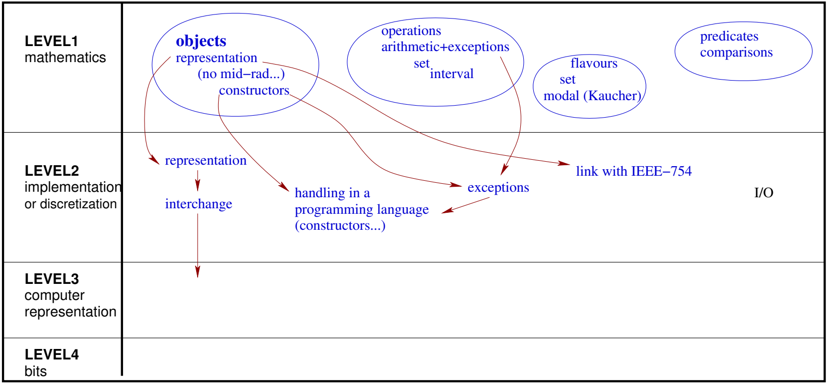
Discussed, alluded to, apparently cset is being developed... concurrently.

Flavors: cset...

Cset: $1/[0,1] = \mathbb{R}$.
Conservative: $\sqrt{[-2,1]} = \text{NaI}$?

Discussed, alluded to, apparently cset is being developed... concurrently.

IEEE-1788 standard: the big picture



Another hot topic: EDP

EDP means Exact Dot Product.
EDP = computing exactly the dot product of two floating-point vectors.
Either in hardware or in software.

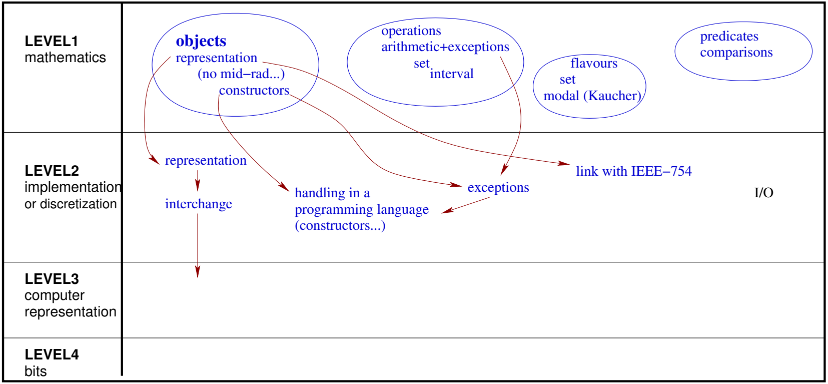
Rejected.

Another hot topic: EDP

EDP means Exact Dot Product.
EDP = computing exactly the dot product of two floating-point vectors.
Either in hardware or in software.

Rejected.

IEEE-1788 standard: the big picture



IEEE-1788 standard: the original project

P1788 Scope: "This standard specifies basic interval arithmetic (IA) operations selecting and following one of the commonly used mathematical interval models. This standard supports the IEEE-754/2008 floating point types of practical use in interval computations. Exception conditions will be defined and standard handling of these conditions will be specified. Consistency with the model is tempered with practical considerations based on input from representatives of vendors and owners of existing systems. The standard provides a layer between the hardware and the programming language levels. It does not mandate that any operations be implemented in hardware. It does not define any realization of the basic operations as functions in a programming language."

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases ($\emptyset$), hooks for variants (Kaucher...)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases (∅), hooks for variants (Kaucher. . .)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases ($\emptyset$), hooks for variants (Kaucher...)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases (∅), hooks for variants (Kaucher. . .)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases ($\emptyset$), hooks for variants (Kaucher...)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases (∅), hooks for variants (Kaucher...)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases (∅), hooks for variants (Kaucher. . .)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases ($\emptyset$),
  hooks for variants (Kaucher. . .)

IEEE-1788 standard: what is still missing

- **Level 4:** probably none of our business
- **Level 3:** chosen to be independent of IEEE-754: probably required features must be explicited and adopted
- **Level 2:** reference implementation
- **Level 1:** cornercases ($\emptyset$),
  hooks for variants (Kaucher. . .)

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: reference implementation

Requirements: $\simeq$ IEEE-754 compliant floating-point numbers

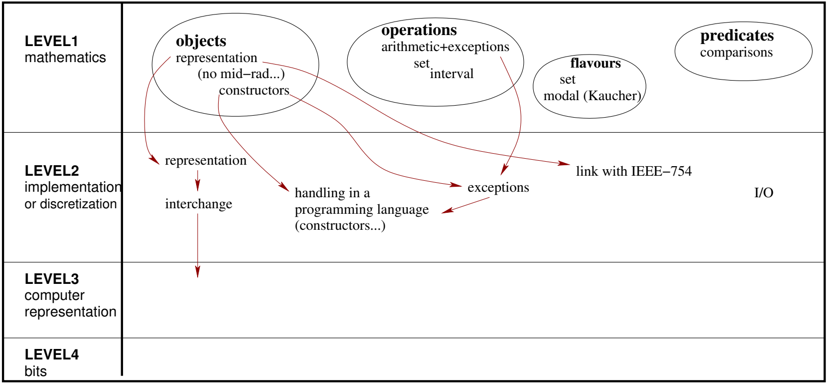
MPFI: interval arithmetic library based on MPFR.

Is MPFI a reference implementation?

No:

- **representation:** by endpoints (not the most efficient use of memory), not yet by mid-rad (**good!**)
- **flavor:** conservative (which is not defined in IEEE-1788)
- **exception:** NaI, no decoration
- **operations:** reverse operations are missing
- **predicates:** missing.

IEEE-1788 standard: the big picture



Motions 7, 8, 15 and 18: Exceptions

Issue: How to handle exceptions efficiently.

- Typical examples:
  (a) Invalid interval constructor
  `interval(3,2)`   `interval("[2.4,3;5]")`
  —interface between interval world and numbers or text strings.
  (b) Elementary function evaluated partly or wholly outside domain
  `sqrt([-1,4])`   `log([-4,-1])`   `[1,2]/[0,0]`
- Type (a) can simply cause nonsense if ignored.
- Type (b) are crucial for applications that depend on fixed-point theorems; but can be ignored by others, e.g. some optimisation algorithms.

Motions 7 and 8: Exceptions, cont.

What to do? A complicated issue.

- Risk that (Level 3) code to handle exceptions will slow down interval applications that don't need it.
- One approach to type (a) is to define an **NaI "Not an Interval"** datum at level 2, encoded at level 3 within the two FP numbers that represent an interval.

Motion 8: Exceptions by Decorations

- Alternative (Motion 8): An extra tag or **decoration** field (1 byte?) in level 3 representation.
- Divided into subfields that record different kinds of exceptional behaviour.
- Decoration is optional, can be added and dropped.
  – To compute at full speed, use "bare" intervals and corresponding "bare" elementary function library.
  – "Decorated" library records exceptions separate from numbers, hence code has fewer IFs & runs fast too. (We hope!)

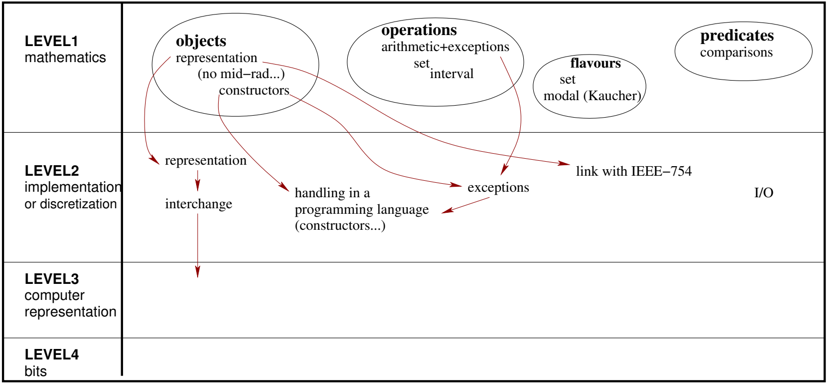
Motions 8, 15 and 18: Decoration issues

Decorations look promising but many Qs exist:

- Bare (double) interval is 16-byte object. Decoration increases this. Can compilers efficiently allocate memory for large arrays of such objects?
- Some proposed decoration-subfields record events in the past; others are properties of the current interval. Can semantic inconsistencies arise?
- Can decoration semantics be specified at Level 2 ...
- ...such that correctness of code can be proven ...
- ...and K.I.S.S. is preserved?

Much work on exceptions remains: list, order...

IEEE-1788 standard: the big picture



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: 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:

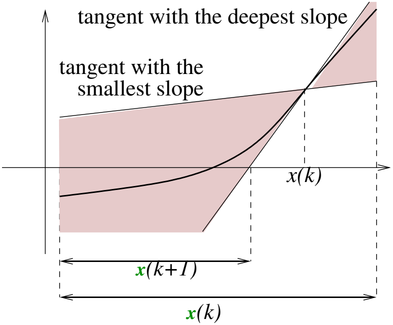
$$\mathbf{x}_{k+1} = \mathbf{x}_k - \frac{f(\mathbf{x}_k)}{f'(\mathbf{x}_k)}$$

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!

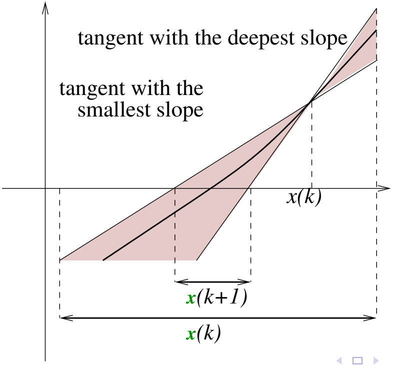
Algorithm: interval Newton (Hansen-Greenberg 83, Baker Kearfott 95-97, Mayer 95, van Hentenryck et al. 97)



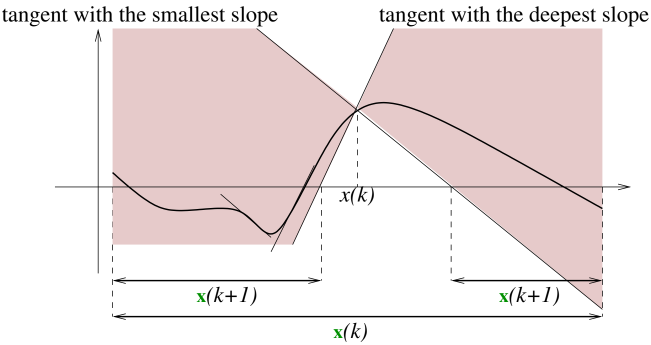
$$\mathbf{x}_{k+1} := \left(x_k - \frac{\mathbf{f}(\{x_k\})}{\mathbf{f}'(\mathbf{x}_k)} \right) \bigcap \mathbf{x}_k$$

Interval Newton: Brouwer theorem

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 $f(B) \subset \text{int}B$ then f has a unique fixed point on B.



Algorithm: interval Newton



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

IEEE-1788 standard: the big picture