Interval Arithmetic: Fundamentals, Successes, and Pitfalls

Ralph Baker Kearfott

Department of Mathematics
University of Louisiana at Lafayette

Universidad EAFIT, Friday, July 27, 2018

Classical Interval Arithmetic
Definition

- Operations are defined over the set of closed and bounded intervals $\boldsymbol{x} = [\underline{x}, \overline{x}]$.

- Operations are defined over the set of closed and bounded intervals $\boldsymbol{x} = [\underline{x}, \overline{x}]$.
- The result of the operation is defined logically for $\odot \in \{+, -, \times, \div\}$ as $\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}$.

Classical Interval Arithmetic

Definition

- Operations are defined over the set of closed and bounded intervals $\boldsymbol{x} = [\underline{x}, \overline{x}]$.
- The result of the operation is defined logically for $\odot \in \{+, -, \times, \div\}$ as $\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}$.
- The logical definition leads to operational definitions:

$$\begin{aligned}
\boldsymbol{x} + \boldsymbol{y} &= [\underline{x} + \underline{y}, \overline{x} + \overline{y}], \\
\boldsymbol{x} - \boldsymbol{y} &= [\underline{x} - \overline{y}, \overline{x} - \underline{y}], \\
\boldsymbol{x} \times \boldsymbol{y} &= [\min\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}, \max\{\underline{x}\underline{y}, \underline{x}\overline{y}, \overline{x}\underline{y}, \overline{x}\overline{y}\}] \\
\frac{1}{\boldsymbol{x}} &= [\frac{1}{\overline{x}}, \frac{1}{\underline{x}}] \qquad \text{if } \underline{x} > 0 \text{ or } \overline{x} < 0 \\
\boldsymbol{x} \div \boldsymbol{y} &= \boldsymbol{x} \times \frac{1}{\boldsymbol{y}}
\end{aligned}$$

(There are alternatives for $\times$ and $\div$ more efficient for certain architectures.)

Classical Interval Arithmetic

What does this definition do?

- In *exact arithmetic*, the operational definitions give the exact ranges of the elementary operations.

Classical Interval Arithmetic

What does this definition do?

- In *exact arithmetic*, the operational definitions give the exact ranges of the elementary operations.
- Evaluating an expression in interval arithmetic does not give an exact range of the expression, but does give bounds on the range of the expression.

- In *exact arithmetic*, the operational definitions give the exact ranges of the elementary operations.
- Evaluating an expression in interval arithmetic does not give an exact range of the expression, but does give bounds on the range of the expression.

- Example (interval dependence)

If $f(x) = (x+1)(x-1)$, then

$$\begin{aligned} \boldsymbol{f}([-2,2]) &= \big([-2,2]+1\big)\big([-2,2]-1\big) \\ &= [-1,3][-3,1] = [-9,3], \end{aligned}$$

whereas the exact range is $[-1,3]$.

- In *exact arithmetic*, the operational definitions give the exact ranges of the elementary operations.
- Evaluating an expression in interval arithmetic does not give an exact range of the expression, but does give bounds on the range of the expression.

- Example (interval dependence)

If $f(x) = (x+1)(x-1)$, then

$$\begin{aligned} \boldsymbol{f}([-2,2]) &= \big([-2,2]+1\big)\big([-2,2]-1\big) \\ &= [-1,3][-3,1] = [-9,3], \end{aligned}$$

whereas the exact range is $[-1,3]$.

- The interval $[-9,3]$ represents the exact range of $\tilde{f}(x,y) = (x+1)(y-1)$ over the rectangle $x \in [-2,2]$, $y \in [-2,2]$ (when x and y vary independently).

- The operational definitions give approximate end points.

- The operational definitions give approximate end points.
- Modern computational environments (such as IEEE 754-compliant ones) allow *rounding down* to the nearest machine number less than the exact result and *rounding up* to the nearest machine number greater than the exact result.

- The operational definitions give approximate end points.
- Modern computational environments (such as IEEE 754-compliant ones) allow *rounding down* to the nearest machine number less than the exact result and *rounding up* to the nearest machine number greater than the exact result.
- If we use downward rounding to compute the lower end point and upward rounding to compute the upper end point, the result of each elementary operation contains the exact range of that operation.

- The operational definitions give approximate end points.
- Modern computational environments (such as IEEE 754-compliant ones) allow *rounding down* to the nearest machine number less than the exact result and *rounding up* to the nearest machine number greater than the exact result.
- If we use downward rounding to compute the lower end point and upward rounding to compute the upper end point, the result of each elementary operation contains the exact range of that operation.
- Hence, an interval evaluation of an expression on a machine mathematically rigorously contains the range of the expression.

Algebraic Properties
(or lack thereof)

- Interval arithmetic is commutative and associative.

Algebraic Properties
(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.

Algebraic Properties
(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.
- For example:
$$\begin{array}{ccccc} [1,2] & - & [1,2] & = & [-1,1] \\ [1,2] & / & [1,2] & = & \left[\frac{1}{2},2\right] \end{array}$$

Algebraic Properties
(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.
- For example: $$\begin{array}{ccccc} [1,2] & - & [1,2] & = & [-1,1] \\ [1,2] & / & [1,2] & = & \left[\frac{1}{2},2\right] \end{array}$$
- Interval arithmetic is only subdistributive: $\boldsymbol{a}(\boldsymbol{b}+\boldsymbol{c}) \subseteq \boldsymbol{ab}+\boldsymbol{ac}$.

Algebraic Properties
(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.
- For example:

$$\begin{array}{ccccc} [1,2] & - & [1,2] & = & [-1,1] \\ [1,2] & / & [1,2] & = & \left[\frac{1}{2},2\right] \end{array}$$

- Interval arithmetic is only subdistributive: $\boldsymbol{a}(\boldsymbol{b}+\boldsymbol{c}) \subseteq \boldsymbol{ab}+\boldsymbol{ac}$.
- For example, $[-1,1]\big([-3,-2]+[2,3]\big) = [-1,1][-1,1] = [-1,1]$, while $[-1,1][-3,-2]+[-1,1][2,3] = [-3,3]+[-3,3] = [-6,6]$.

Algebraic Properties

(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.
- For example: $$\begin{array}{ccccc} [1,2] & - & [1,2] & = & [-1,1] \\ [1,2] & / & [1,2] & = & \left[\frac{1}{2},2\right] \end{array}$$
- Interval arithmetic is only subdistributive: $\boldsymbol{a}(\boldsymbol{b}+\boldsymbol{c}) \subseteq \boldsymbol{ab}+\boldsymbol{ac}$.
- For example, $[-1,1]\big([-3,-2]+[2,3]\big) = [-1,1][-1,1] = [-1,1]$, while $[-1,1][-3,-2]+[-1,1][2,3] = [-3,3]+[-3,3] = [-6,6]$.

- **Theorem (Single Use Expressions — SUE)**

  In an algebraic expression evaluated in exact interval arithmetic, the result is the exact range if each variable occurs only once in the expression.

Algebraic Properties

(or lack thereof)

- Interval arithmetic is commutative and associative.
- There are no additive and multiplicative inverses.
- For example: $$\begin{array}{rcccl} [1,2] & - & [1,2] & = & [-1,1] \\ [1,2] & / & [1,2] & = & \left[\frac{1}{2},2\right] \end{array}$$
- Interval arithmetic is only subdistributive: $\boldsymbol{a}(\boldsymbol{b}+\boldsymbol{c}) \subseteq \boldsymbol{ab}+\boldsymbol{ac}$.
- For example,
$[-1,1]\big([-3,-2]+[2,3]\big) = [-1,1][-1,1] = [-1,1]$, while
$[-1,1][-3,-2]+[-1,1][2,3] = [-3,3]+[-3,3] = [-6,6]$.

- Theorem (Single Use Expressions — SUE)
In an algebraic expression evaluated in exact interval arithmetic, the result is the exact range if each variable occurs only once in the expression.
    - *Note: The converse is not true.*

Midpoint-radius arithmetic: Intervals represented in terms of midpoint and error; addition gives exact range but multiplication just gives an enclosure for the range.

Alternative "Interval" Systems

(Different representations or different semantics)

Midpoint-radius arithmetic: Intervals represented in terms of midpoint and error; addition gives exact range but multiplication just gives an enclosure for the range.

Circular arithmetic: Representation as midpoint-radius, but with the midpoint in the complex plane. Elementary operations are not exact, but are mere enclosures.

Alternative "Interval" Systems

(Different representations or different semantics)

Midpoint-radius arithmetic: Intervals represented in terms of midpoint and error; addition gives exact range but multiplication just gives an enclosure for the range.

Circular arithmetic: Representation as midpoint-radius, but with the midpoint in the complex plane. Elementary operations are not exact, but are mere enclosures.

Rectangular arithmetic: An alternative complex interval arithmetic. Addition is exact, but multiplication just gives an enclosure.

Alternative "Interval" Systems

(Different representations or different semantics)

Midpoint-radius arithmetic: Intervals represented in terms of midpoint and error; addition gives exact range but multiplication just gives an enclosure for the range.

Circular arithmetic: Representation as midpoint-radius, but with the midpoint in the complex plane. Elementary operations are not exact, but are mere enclosures.

Rectangular arithmetic: An alternative complex interval arithmetic. Addition is exact, but multiplication just gives an enclosure.

Kaucher arithmetic, modal arithmetic etc.: Algebraically completes interval arithmetic with additive inverses. It has uses, but interpretation of the results is more complicated, sometimes depending on monotonicity properties.

Extensions

What do we do with this?

Consider $\frac{x}{y} = [1, 2]/[-3, 4]$.

Extensions

What do we do with this?

Consider $\frac{x}{y} = [1, 2]/[-3, 4]$.

- In our operational definition, $\frac{1}{y} = \left[\frac{1}{4}, -\frac{1}{3}\right]$???

Consider $\frac{\boldsymbol{x}}{\boldsymbol{y}} = [1,2]/[-3,4]$.

- In our operational definition, $\frac{1}{\boldsymbol{y}} = \left[\frac{1}{4}, -\frac{1}{3}\right]$???
- The arguments contain undefined quantities $\frac{a}{0}$ for $a \in [1,2]$, but . . .

Extensions

What do we do with this?

Consider $\dfrac{\boldsymbol{x}}{\boldsymbol{y}} = [1,2]/[-3,4]$.

- In our operational definition, $\dfrac{1}{\boldsymbol{y}} = \left[\dfrac{1}{4}, -\dfrac{1}{3}\right]$???
- The arguments contain undefined quantities $\dfrac{a}{0}$ for $a \in [1,2]$, but ...
- The range of the operation over defined quantities is $\left(-\infty, -\frac{1}{3}\right] \bigcup \left[\frac{1}{4}, \infty\right)$.

Extensions

What do we do with this?

Consider $\frac{x}{y} = [1,2]/[-3,4]$.

- In our operational definition, $\frac{1}{y} = \left[\frac{1}{4}, -\frac{1}{3}\right]$???
- The arguments contain undefined quantities $\frac{a}{0}$ for $a \in [1,2]$, but . . .
- The range of the operation over defined quantities is $\left(-\infty, -\frac{1}{3}\right] \bigcup \left[\frac{1}{4}, \infty\right)$.
- Different definitions for the operation's result and different interpretations are appropriate in different contexts. (More to be said later.)

Extensions

What do we do with this?

Consider $\frac{x}{y} = [1,2]/[-3,4]$.

- In our operational definition, $\frac{1}{y} = \left[\frac{1}{4}, -\frac{1}{3}\right]$???
- The arguments contain undefined quantities $\frac{a}{0}$ for $a \in [1,2]$, but . . .
- The range of the operation over defined quantities is $\left(-\infty, -\frac{1}{3}\right] \bigcup \left[\frac{1}{4}, \infty\right)$.
- Different definitions for the operation's result and different interpretations are appropriate in different contexts. (More to be said later.)
- This has been carefully considered and defined in an exception-tracking framework in the IEEE 1788-2015 standard for interval arithmetic.

The same basic interval operations described in all of the early work, although it was apparently done independently.

History
Early work

The same basic interval operations described in all of the early work, although it was apparently done independently.
Rosaline Cecily Young (*Mathematische Annalen*, 1931,)

History
Early work

The same basic interval operations described in all of the early work, although it was apparently done independently.

Rosaline Cecily Young (*Mathematische Annalen*, 1931,)
: prior to digital computers) "The Algebra of Many-Valued Quantities." The focus is on an arithmetic on limits, where $\liminf_{x \to x_0} f(x)$ and $\limsup_{x \to x_0} f(x)$ are distinct (such as in in generalized gradients of nonsmooth functions). Ranges and roundoff error do not seem to have been the primary motivation.

The same basic interval operations described in all of the early work, although it was apparently done independently.

Rosaline Cecily Young (*Mathematische Annalen*, 1931,) prior to digital computers) “The Algebra of Many-Valued Quantities.” The focus is on an arithmetic on limits, where $\liminf_{x\to x_0} f(x)$ and $\limsup_{x\to x_0} f(x)$ are distinct (such as in in generalized gradients of nonsmooth functions). Ranges and roundoff error do not seem to have been the primary motivation.

Paul S. Dwyer (Chapter in *Linear Computations*, 1951)

The same basic interval operations described in all of the early work, although it was apparently done independently.

Rosaline Cecily Young (*Mathematische Annalen*, 1931,)
prior to digital computers) "The Algebra of Many-Valued Quantities." The focus is on an arithmetic on limits, where $\liminf_{x \to x_0} f(x)$ and $\limsup_{x \to x_0} f(x)$ are distinct (such as in in generalized gradients of nonsmooth functions). Ranges and roundoff error do not seem to have been the primary motivation.

Paul S. Dwyer (Chapter in *Linear Computations*, 1951)
"Computation with Approximate Numbers." Interval computations are introduced as an integral part of roundoff error analysis.

The same basic interval operations described in all of the early work, although it was apparently done independently.

Rosaline Cecily Young (*Mathematische Annalen*, 1931,) prior to digital computers) "The Algebra of Many-Valued Quantities." The focus is on an arithmetic on limits, where $\liminf_{x\to x_0} f(x)$ and $\limsup_{x\to x_0} f(x)$ are distinct (such as in in generalized gradients of nonsmooth functions). Ranges and roundoff error do not seem to have been the primary motivation.

Paul S. Dwyer (Chapter in *Linear Computations*, 1951) "Computation with Approximate Numbers." Interval computations are introduced as an integral part of roundoff error analysis.

Mieczyslaw Warmus (*Calculus of Approximations*, 1956)

The same basic interval operations described in all of the early work, although it was apparently done independently.

Rosaline Cecily Young (*Mathematische Annalen*, 1931,) prior to digital computers) "The Algebra of Many-Valued Quantities." The focus is on an arithmetic on limits, where $\liminf_{x \to x_0} f(x)$ and $\limsup_{x \to x_0} f(x)$ are distinct (such as in in generalized gradients of nonsmooth functions). Ranges and roundoff error do not seem to have been the primary motivation.

Paul S. Dwyer (Chapter in *Linear Computations*, 1951) "Computation with Approximate Numbers." Interval computations are introduced as an integral part of roundoff error analysis.

Mieczyslaw Warmus (*Calculus of Approximations*, 1956) The motivation is apparently to provide a sound theoretical backing to numerical computation.

Really Early Work
(from a talk on the Origin of Intervals by Siegfried Rump)

Rump mentions

Really Early Work
(from a talk on the Origin of Intervals by Siegfried Rump)

Rump mentions

- a 1900 book *Lectures on Numerical Computing* (in German) with error bounds for $+$, $-$, $\cdot$, $/$ and innacurate input data;

Really Early Work

(from a talk on the Origin of Intervals by Siegfried Rump)

Rump mentions

- a 1900 book *Lectures on Numerical Computing* (in German) with error bounds for $+$, $-$, $\cdot$, $/$ and innacurate input data;
- an 1896 article “On computing with inexact numbers” (in German) in the *Journal for Junior Highschool Studies*, giving the impression interval computations were standard fare in middle schools;

Really Early Work
(from a talk on the Origin of Intervals by Siegfried Rump)

Rump mentions

- a 1900 book *Lectures on Numerical Computing* (in German) with error bounds for $+$, $-$, $\cdot$, $/$ and innacurate input data;
- an 1896 article “On computing with inexact numbers” (in German) in the *Journal for Junior Highschool Studies*, giving the impression interval computations were standard fare in middle schools;
- 1887, 1879, and 1854 French work where explicit formulas for the elementary operations and rigorous error bounds were given;

Really Early Work

(from a talk on the Origin of Intervals by Siegfried Rump)

Rump mentions

- a 1900 book *Lectures on Numerical Computing* (in German) with error bounds for $+$, $-$, $\cdot$, $/$ and innacurate input data;
- an 1896 article “On computing with inexact numbers” (in German) in the *Journal for Junior Highschool Studies*, giving the impression interval computations were standard fare in middle schools;
- 1887, 1879, and 1854 French work where explicit formulas for the elementary operations and rigorous error bounds were given;
- An 1809 work by Gauß in Latin where explicit computation of error bounds, including rounding errors, appears.

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

"Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

> "Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

Ray Moore (*Lockheed Technical Report*, 1959)

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

"Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

Ray Moore (*Lockheed Technical Report*, 1959)

"Automatic Error Analysis in Digital Computation." The basic operations are given in this monograph.

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

"Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

Ray Moore (*Lockheed Technical Report*, 1959)

"Automatic Error Analysis in Digital Computation." The basic operations are given in this monograph.

- Numerical solution of ODEs, numerical integration, etc. based on intervals appear in Moore's 1962 dissertation.

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)

"Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

Ray Moore (*Lockheed Technical Report*, 1959)

"Automatic Error Analysis in Digital Computation." The basic operations are given in this monograph.

- Numerical solution of ODEs, numerical integration, etc. based on intervals appear in Moore's 1962 dissertation.

Eldon Hansen collaborated with Ray Moore; wrote two editions of *Global Optimization with Interval Analysis*, with with William Walster.

History of Interval Arithmetic

It takes off.

(beginning of the era of modern computers)

Teruro Sunaga (*RAAG Memoirs*, 1958)
: "Theory of an Interval Algebra and its Application to Numerical Analysis." Motivation is automatically accounting for uncertainty and error in measurement and computation.

Ray Moore (*Lockheed Technical Report*, 1959)
: "Automatic Error Analysis in Digital Computation." The basic operations are given in this monograph.

- Numerical solution of ODEs, numerical integration, etc. based on intervals appear in Moore's 1962 dissertation.

Eldon Hansen collaborated with Ray Moore; wrote two editions of *Global Optimization with Interval Analysis*, with with William Walster.

William Kahan Proposed extended interval arithmetic, saw to directed roundings in IEEE 754 and mentored several currently prominent students.

History of Interval Arithmetic

Others

(Karlsruhe)

History of Interval Arithmetic

Others

(Karlsruhe)

Rudolf Krawczyk published his famous Krawczyk method for existence / uniqueness proofs (1969)

History of Interval Arithmetic

Others

(Karlsruhe)

Rudolf Krawczyk published his famous Krawczyk method for existence / uniqueness proofs (1969)

Ulrich Kulisch at Karlsruhe mentored many students. The group produced the “SC” languages, with an interval data type.

History of Interval Arithmetic

Others

(Karlsruhe)

Rudolf Krawczyk published his famous Krawczyk method for existence / uniqueness proofs (1969)

Ulrich Kulisch at Karlsruhe mentored many students. The group produced the “SC” languages, with an interval data type.

Götz Alefeld wrote *Introduction to Interval Computations*, among other

History of Interval Arithmetic

Others

(Karlsruhe)

Rudolf Krawczyk published his famous Krawczyk method for existence / uniqueness proofs (1969)

Ulrich Kulisch at Karlsruhe mentored many students. The group produced the “SC” languages, with an interval data type.

Götz Alefeld wrote *Introduction to Interval Computations*, among other

Siegfried Rump developed INTLAB, a Matlab toolbox for IA, and founded the Institute for Reliable Computing at Hamburg.

History
(Others)

History
(Others)

E.T.H. Zürich (Peter Henrici)

History
(Others)

E.T.H. Zürich (Peter Henrici)

Universität Freiburg (Karl Nickel)

History
(Others)

E.T.H. Zürich (Peter Henrici)

Universität Freiburg (Karl Nickel)

Universität Vienna (Arnold Neumaier) – Important for the group's work in global optimization and a global optimization web site.

History
(Others)

E.T.H. Zürich (Peter Henrici)

Universität Freiburg (Karl Nickel)

Universität Vienna (Arnold Neumaier) – Important for the group's work in global optimization and a global optimization web site.

Various locations in Russia Extensive publications.

History
(Others)

E.T.H. Zürich (Peter Henrici)

Universität Freiburg (Karl Nickel)

Universität Vienna (Arnold Neumaier) – Important for the group's work in global optimization and a global optimization web site.

Various locations in Russia Extensive publications.

Various locations in France Notable successes in autonomous robots, navigation, etc. Also, perhaps the most significant locus of current applied research.

History
(Others)

E.T.H. Zürich (Peter Henrici)

Universität Freiburg (Karl Nickel)

Universität Vienna (Arnold Neumaier) – Important for the group's work in global optimization and a global optimization web site.

Various locations in Russia Extensive publications.

Various locations in France Notable successes in autonomous robots, navigation, etc. Also, perhaps the most significant locus of current applied research.

Czech Republic Jiri Rohn and associates have developed theory, methods, and software for interval linear systems.

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

Reasons for Interval Arithmetic

(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

- provides a polynomial-time computation that often gives helpful bounds, for . . .

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

- provides a polynomial-time computation that often gives helpful bounds, for . . .
  - proving the hypotheses of fixed point theorems,

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

- provides a polynomial-time computation that often gives helpful bounds, for . . .
  - proving the hypotheses of fixed point theorems,
  - bounding the objective function and proving or disproving feasibility in global optimization algorithms,

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

- provides a polynomial-time computation that often gives helpful bounds, for . . .
  - proving the hypotheses of fixed point theorems,
  - bounding the objective function and proving or disproving feasibility in global optimization algorithms,
  - proving collision avoidance in robotics, navigation systems, celestial mechanics,

Reasons for Interval Arithmetic
(general uses)

Rigorously bounding roundoff error in floating point computations.

- Interval widths start out small, on the order of the machine precision, but . . .
- overestimation can make results meaningless, and obtaining meaningful results is often tricky.

Bounding function ranges over large domains

- provides a polynomial-time computation that often gives helpful bounds, for . . .
  - proving the hypotheses of fixed point theorems,
  - bounding the objective function and proving or disproving feasibility in global optimization algorithms,
  - proving collision avoidance in robotics, navigation systems, celestial mechanics,
  - etc.

Theorem (Brouwer fixed point theorem)

If g is a continuous mapping from a compact convex set $\boldsymbol{x}$ into itself, there is a fixed-point $x \in \boldsymbol{x}$ of g, i.e. $g(x) = x$.

Theorem (Brouwer fixed point theorem)

If g is a continuous mapping from a compact convex set $\boldsymbol{x}$ into itself, there is a fixed-point $x \in \boldsymbol{x}$ of g, i.e. $g(x) = x$.

- If we evaluate $g : \boldsymbol{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ over an interval vector $\boldsymbol{x}$ and the interval value $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, this proves existence of a fixed point of g in $\boldsymbol{x}$.

Theorem (Brouwer fixed point theorem)

If g is a continuous mapping from a compact convex set $\boldsymbol{x}$ into itself, there is a fixed-point $x \in \boldsymbol{x}$ of g, i.e. $g(x) = x$.

- If we evaluate $g : \boldsymbol{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ over an interval vector $\boldsymbol{x}$ and the interval value $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, this proves existence of a fixed point of g in $\boldsymbol{x}$.
- Interval techniques based on this prove existence, uniqueness, and error bounds:

Theorem (Brouwer fixed point theorem)

If g is a continuous mapping from a compact convex set $\boldsymbol{x}$ into itself, there is a fixed-point $x \in \boldsymbol{x}$ of g, i.e. $g(x) = x$.

- If we evaluate $g : \boldsymbol{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ over an interval vector $\boldsymbol{x}$ and the interval value $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, this proves existence of a fixed point of g in $\boldsymbol{x}$.
- Interval techniques based on this prove existence, uniqueness, and error bounds:
  - the Krawczyk method.

Theorem (Brouwer fixed point theorem)

If g is a continuous mapping from a compact convex set $\boldsymbol{x}$ *into itself, there is a fixed-point* $x \in \boldsymbol{x}$ *of g, i.e.* $g(x) = x$*.*

- If we evaluate $g : \boldsymbol{x} \subset \mathbb{R}^n \to \mathbb{R}^n$ over an interval vector $\boldsymbol{x}$ and the interval value $\boldsymbol{g}(\boldsymbol{x}) \subseteq \boldsymbol{x}$, this proves existence of a fixed point of g in $\boldsymbol{x}$.
- Interval techniques based on this prove existence, uniqueness, and error bounds:
  - the Krawczyk method.
  - general interval Newton methods, such as the interval Gauss–Seidel method.

Successes
Proof of Important Conjectures
prize-winning work

- The Kepler Conjecture: (made by Johannes Kepler in 1611, proved with interval arithmetic by Thomas Hales) — no packing of spheres in 3-dimensional space is denser than face-centered cubic packing. See https://en.wikipedia.org/wiki/Kepler_conjecture.

Successes
Proof of Important Conjectures
prize-winning work

- The Kepler Conjecture: (made by Johannes Kepler in 1611, proved with interval arithmetic by Thomas Hales) — no packing of spheres in 3-dimensional space is denser than face-centered cubic packing. See `https://en.wikipedia.org/wiki/Kepler_conjecture`.
- Chaos and attractors for the Lorenz equations: (a simplified ODE model of weather prediction).

Successes

Proof of Important Conjectures

prize-winning work

- The Kepler Conjecture: (made by Johannes Kepler in 1611, proved with interval arithmetic by Thomas Hales) — no packing of spheres in 3-dimensional space is denser than face-centered cubic packing. See https://en.wikipedia.org/wiki/Kepler_conjecture.
- Chaos and attractors for the Lorenz equations: (a simplified ODE model of weather prediction).
  - Hassard, Zhang, Hastings, and Troy use a mathematically rigorous interval-arithmetic-based ODE integrator to prove existence of chaotic solutions.
  - (and earlier) Mischaikov and Mrozek use Conley index theory and interval arithmetic to do the same.

Successes

Proof of Important Conjectures

prize-winning work

- The Kepler Conjecture: (made by Johannes Kepler in 1611, proved with interval arithmetic by Thomas Hales) — no packing of spheres in 3-dimensional space is denser than face-centered cubic packing. See https://en.wikipedia.org/wiki/Kepler_conjecture.
- Chaos and attractors for the Lorenz equations: (a simplified ODE model of weather prediction).
  - Hassard, Zhang, Hastings, and Troy use a mathematically rigorous interval-arithmetic-based ODE integrator to prove existence of chaotic solutions.
  - (and earlier) Mischaikov and Mrozek use Conley index theory and interval arithmetic to do the same.
  - Warwick Tucker (in dissertation work) used normal form theory and interval arithmetic to solve Stephen Smale's 14-th problem, namely, that the Lorenz equations have a persistent strange attractor.

Proof of Important Conjectures

Additional prize-winning work

The R. E. Moore Prize for application of interval arithmetic has been awarded to various researchers. See http://www.cs.utep.edu/interval-comp/honors.html. Among these are:

Proof of Important Conjectures

Additional prize-winning work

The R. E. Moore Prize for application of interval arithmetic has been awarded to various researchers. See `http://www.cs.utep.edu/interval-comp/honors.html`. Among these are:

2014 Kenta Kobayashi for Computer-Assisted Uniqueness Proof for Stokes' Wave of Extreme Form, and

Proof of Important Conjectures

Additional prize-winning work

The R. E. Moore Prize for application of interval arithmetic has been awarded to various researchers. See `http://www.cs.utep.edu/interval-comp/honors.html`. Among these are:

2014 Kenta Kobayashi for Computer-Assisted Uniqueness Proof for Stokes' Wave of Extreme Form, and

2016 Banhelyi, Csendes, Krisztin, and Neumaier for Global attractivity of the zero solution for Wright's equation (a model in population genetics)

Proof of Important Conjectures

Additional prize-winning work

The R. E. Moore Prize for application of interval arithmetic has been awarded to various researchers. See http://www.cs.utep.edu/interval-comp/honors.html. Among these are:

2014 Kenta Kobayashi for Computer-Assisted Uniqueness Proof for Stokes' Wave of Extreme Form, and

2016 Banhelyi, Csendes, Krisztin, and Neumaier for Global attractivity of the zero solution for Wright's equation (a model in population genetics)

2018 Figueras, Haro, and Luque for Rigorous Computer-Assisted Application of KAM Theory: A Modern Approach

Proof of Important Conjectures

Additional prize-winning work

The R. E. Moore Prize for application of interval arithmetic has been awarded to various researchers. See `http://www.cs.utep.edu/interval-comp/honors.html`. Among these are:

2014 Kenta Kobayashi for Computer-Assisted Uniqueness Proof for Stokes' Wave of Extreme Form, and

2016 Banhelyi, Csendes, Krisztin, and Neumaier for Global attractivity of the zero solution for Wright's equation (a model in population genetics)

2018 Figueras, Haro, and Luque for Rigorous Computer-Assisted Application of KAM Theory: A Modern Approach

2018 (a notable runner-up) Jean-Pierre Merlet for Simulation of discrete-time controlled cable-driven parallel robots on a trajectory

These include:

These include:

1. Simple use of range bounds;

These include:

1. Simple use of range bounds;
2. Incorporation of range bounds in exhaustive domain searches (branch and bound algorithms) to enclose a global optimum of a minimization problem;

2. Stadtherr et al Correction of major errors in widely used tables of vapor-liquid equilibria.

Engineering Questions Rigorously Resolved

Physics and chemical engineering

These include:

1. Simple use of range bounds;
2. Incorporation of range bounds in exhaustive domain searches (branch and bound algorithms) to enclose a global optimum of a minimization problem;
3. Incorporation of range bounds to rigorously enclose solution sets to differential equations in sophisticated mathematically rigorous ODE integrators.

2. Stadtherr et al Correction of major errors in widely used tables of vapor-liquid equilibria.
3. Berz et al Proof of stability of the beam, given assumed tolerances on the geometry and magnets, of the once-proposed superconducting supercollider (and the software continues to be used for other cyclotrons).

Engineering Questions Rigorously Resolved

Robotics

- Luc Jaulin et al have used interval constraint propagation to increase both reliability and efficiency of underwater robot control and data analysis in generating maps. (Luc is the 2012 Moore Prize recipient.)

Engineering Questions Rigorously Resolved
Robotics

- Luc Jaulin et al have used interval constraint propagation to increase both reliability and efficiency of underwater robot control and data analysis in generating maps. (Luc is the 2012 Moore Prize recipient.)
- (Earlier work continuing to the present) The forward manipulator problem (computation of joint angles for a particular robot hand location) is easily solved with exhaustive search (branch and bound) to the corresponding systems of nonlinear equations.

Engineering Questions Rigorously Resolved

Robotics

- Luc Jaulin et al have used interval constraint propagation to increase both reliability and efficiency of underwater robot control and data analysis in generating maps. (Luc is the 2012 Moore Prize recipient.)
- (Earlier work continuing to the present) The forward manipulator problem (computation of joint angles for a particular robot hand location) is easily solved with exhaustive search (branch and bound) to the corresponding systems of nonlinear equations.
- Interval arithmetic can be used in collision avoidance.

- Luc Jaulin et al have used interval constraint propagation to increase both reliability and efficiency of underwater robot control and data analysis in generating maps. (Luc is the 2012 Moore Prize recipient.)
- (Earlier work continuing to the present) The forward manipulator problem (computation of joint angles for a particular robot hand location) is easily solved with exhaustive search (branch and bound) to the corresponding systems of nonlinear equations.
- Interval arithmetic can be used in collision avoidance.

In early work (1988) yours truly used Fortran-77-based software to show the set of published solutions to a manipulator problem posed by Alexander Morgan at General Motors was incorrect. This led to discovery of an incorrectly-given coefficient in the paper and to improvement in the software in use at General Motors.

Pitfalls

Where should interval arithmetic be used?

- Replacing floating point data types by intervals generally does not work. Due to interval dependency, this commonly results in output intervals of (∞, ∞).

Pitfalls

Where should interval arithmetic be used?

- Replacing floating point data types by intervals generally does not work. Due to interval dependency, this commonly results in output intervals of (∞, ∞).
- Different algorithms are used for interval computations Also, different algorithms are used, depending on whether there are just small roundoff errors or large uncertainties in the data.

Pitfalls

Where should interval arithmetic be used?

- Replacing floating point data types by intervals generally does not work. Due to interval dependency, this commonly results in output intervals of (∞, ∞).
- Different algorithms are used for interval computations Also, different algorithms are used, depending on whether there are just small roundoff errors or large uncertainties in the data.
- Rule of thumb: Use floating point computations where verification is not needed, and use intervals only to provide bounds in strategic places.

Pitfalls

Where should interval arithmetic be used?

- Replacing floating point data types by intervals generally does not work. Due to interval dependency, this commonly results in output intervals of (∞, ∞).
- Different algorithms are used for interval computations Also, different algorithms are used, depending on whether there are just small roundoff errors or large uncertainties in the data.
- Rule of thumb: Use floating point computations where verification is not needed, and use intervals only to provide bounds in strategic places.
- Keep the interval computations as simple as possible.

Pitfalls

Where should interval arithmetic be used?

- Replacing floating point data types by intervals generally does not work. Due to interval dependency, this commonly results in output intervals of (∞, ∞).
- Different algorithms are used for interval computations Also, different algorithms are used, depending on whether there are just small roundoff errors or large uncertainties in the data.
- Rule of thumb: Use floating point computations where verification is not needed, and use intervals only to provide bounds in strategic places.
- Keep the interval computations as simple as possible.

The above considerations are how success is achieved.

Consider proving existence of a solution with the Brouwer fixed point theorem.

Pitfalls

Care should be taken in the logic.

Consider proving existence of a solution with the Brouwer fixed point theorem.

- **Example**

  Consider $g(x) = \sqrt{x-1} + 0.9$, with a fixed point at $x \approx 1.0127$ and $x \approx 1.7873$.

Pitfalls

Care should be taken in the logic.

Consider proving existence of a solution with the Brouwer fixed point theorem.

- **Example**

  Consider $g(x) = \sqrt{x-1} + 0.9$, with a fixed point at $x \approx 1.0127$ and $x \approx 1.7873$.

  - On $x \in [1.5, 2]$, an interval evaluation gives $\boldsymbol{g}(\boldsymbol{x}) \subseteq [1.6071, 1.9001] \subset [1.5, 2]$, and we correctly conclude g has a fixed point in $[1.6071, 1.9001]$.
    However, $\cdots$

Pitfalls

Care should be taken in the logic.

Consider proving existence of a solution with the Brouwer fixed point theorem.

- **Example**

  Consider $g(x) = \sqrt{x-1} + 0.9$, with a fixed point at $x \approx 1.0127$ and $x \approx 1.7873$.

  - On $x \in [1.5, 2]$, an interval evaluation gives $\boldsymbol{g}(\boldsymbol{x}) \subseteq [1.6071, 1.9001] \subset [1.5, 2]$, and we correctly conclude g has a fixed point in $[1.6071, 1.9001]$. However, $\cdots$
  - if $\boldsymbol{x} = [0, 1]$, $\sqrt{x-1} = \sqrt{[-1, 0]}$ evaluates to $[0, 0]$, so $\boldsymbol{g}(\boldsymbol{x}) = [0.9, 0.9] \subset \boldsymbol{x}$, for an incorrect conclusion.

Pitfalls

Care should be taken in the logic.

Consider proving existence of a solution with the Brouwer fixed point theorem.

- **Example**

  Consider $g(x) = \sqrt{x-1} + 0.9$, with a fixed point at $x \approx 1.0127$ and $x \approx 1.7873$.

  - On $x \in [1.5, 2]$, an interval evaluation gives $\boldsymbol{g}(\boldsymbol{x}) \subseteq [1.6071, 1.9001] \subset [1.5, 2]$, and we correctly conclude g has a fixed point in $[1.6071, 1.9001]$. However, $\cdots$
  - if $\boldsymbol{x} = [0, 1]$, $\sqrt{x-1} = \sqrt{[-1, 0]}$ evaluates to $[0, 0]$, so $\boldsymbol{g}(\boldsymbol{x}) = [0.9, 0.9] \subset \boldsymbol{x}$, for an incorrect conclusion.

  - This is due to g not continuous on $[0, 1]$, combined with loose evaluation (returning the range only over the intersection of the domain of g with the interval).

Consider proving existence of a solution with the Brouwer fixed point theorem.

- **Example**

  Consider $g(x) = \sqrt{x-1} + 0.9$, with a fixed point at $x \approx 1.0127$ and $x \approx 1.7873$.

  - On $x \in [1.5, 2]$, an interval evaluation gives $\boldsymbol{g}(\boldsymbol{x}) \subseteq [1.6071, 1.9001] \subset [1.5, 2]$, and we correctly conclude g has a fixed point in $[1.6071, 1.9001]$. However, $\cdots$
  - if $\boldsymbol{x} = [0, 1]$, $\sqrt{x-1} = \sqrt{[-1, 0]}$ evaluates to $[0, 0]$, so $\boldsymbol{g}(\boldsymbol{x}) = [0.9, 0.9] \subset \boldsymbol{x}$, for an incorrect conclusion.

  - This is due to g not continuous on $[0, 1]$, combined with loose evaluation (returning the range only over the intersection of the domain of g with the interval).
  - Loose evaluation is appropriate in other contexts.

Taming Interval Dependency

Constraint Propagation, Subdivision

To reduce the overestimation in evaluating an interval expression, we may

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.

Taming Interval Dependency

Constraint Propagation, Subdivision

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.
- Use constraint propagation (solving for variables or subexpressions in terms of other variables with known smaller uncertainties)

Taming Interval Dependency

Constraint Propagation, Subdivision

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.
- Use constraint propagation (solving for variables or subexpressions in terms of other variables with known smaller uncertainties)
- Subdivide the intervals of uncertainty, compute interval values over these subintervals, and form the union of these interval values.

Taming Interval Dependency

Constraint Propagation, Subdivision

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.
- Use constraint propagation (solving for variables or subexpressions in terms of other variables with known smaller uncertainties)
- Subdivide the intervals of uncertainty, compute interval values over these subintervals, and form the union of these interval values.
- Single out dependencies as separate variables.

Taming Interval Dependency

Constraint Propagation, Subdivision

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.
- Use constraint propagation (solving for variables or subexpressions in terms of other variables with known smaller uncertainties)
- Subdivide the intervals of uncertainty, compute interval values over these subintervals, and form the union of these interval values.
- Single out dependencies as separate variables.
- Use any of many techniques in the literature.

To reduce the overestimation in evaluating an interval expression, we may

- Rearrange the expression(s) or systems of equations.
- Use constraint propagation (solving for variables or subexpressions in terms of other variables with known smaller uncertainties)
- Subdivide the intervals of uncertainty, compute interval values over these subintervals, and form the union of these interval values.
- Single out dependencies as separate variables.
- Use any of many techniques in the literature.
- Cleverly use properties of the specific problem.

Available Software

Packages for Verified Solution of ODE Systems

- COSY-infinity, the cyclotron beam software by Berz and Makino. This is the most successful software for handling input data with wide intervals. However, the old version is no longer distributed and the new version is not yet publicly available. (C / C++ / Fortran)

Available Software

Packages for Verified Solution of ODE Systems

- COSY-infinity, the cyclotron beam software by Berz and Makino. This is the most successful software for handling input data with wide intervals. However, the old version is no longer distributed and the new version is not yet publicly available. (C / C++ / Fortran)
- Codes from Mark Stadtherr et al: These have also been successful for somewhat wider intervals, but not as much as that of Berz and Makino. They are research codes, not of “industrial strength." (C++)

Available Software

Packages for Verified Solution of ODE Systems

- COSY-infinity, the cyclotron beam software by Berz and Makino. This is the most successful software for handling input data with wide intervals. However, the old version is no longer distributed and the new version is not yet publicly available. (C / C++ / Fortran)
- Codes from Mark Stadtherr et al: These have also been successful for somewhat wider intervals, but not as much as that of Berz and Makino. They are research codes, not of “industrial strength." (C++)
- Other generally available codes: Some of these are polished and packaged, but may be practical only for narrow or point input data, to bound computational errors. Two such packages are:

Available Software

Packages for Verified Solution of ODE Systems

- COSY-infinity, the cyclotron beam software by Berz and Makino. This is the most successful software for handling input data with wide intervals. However, the old version is no longer distributed and the new version is not yet publicly available. (C / C++ / Fortran)
- Codes from Mark Stadtherr et al: These have also been successful for somewhat wider intervals, but not as much as that of Berz and Makino. They are research codes, not of “industrial strength." (C++)
- Other generally available codes: Some of these are polished and packaged, but may be practical only for narrow or point input data, to bound computational errors. Two such packages are:
  - VNODE-LP (Nedialkov et al, literate programming/C++),
  - ValEncIA-IVP (Rauh, Hofer, Auer, C++, somewhat older)

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.
  - IBEX, a C++ library for constraint propagation. (extensively used by its developers in pattern recognition and robotics).

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.
  - IBEX, a C++ library for constraint propagation. (extensively used by its developers in pattern recognition and robotics).
  - GlobSol, primarily a Fortran 90 library, with some C / C++ and a Matlab interface. (This is my work – it is somewhat dated and not optimally efficient, but certainly available.)

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.
  - IBEX, a C++ library for constraint propagation. (extensively used by its developers in pattern recognition and robotics).
  - GlobSol, primarily a Fortran 90 library, with some C / C++ and a Matlab interface. (This is my work – it is somewhat dated and not optimally efficient, but certainly available.)
- Some general toolboxes.

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.
  - IBEX, a C++ library for constraint propagation. (extensively used by its developers in pattern recognition and robotics).
  - GlobSol, primarily a Fortran 90 library, with some C / C++ and a Matlab interface. (This is my work – it is somewhat dated and not optimally efficient, but certainly available.)
- Some general toolboxes.
  - INTLAB, a very popular Matlab toolbox for interval computations.

Existence Verification Packages

Usually part of larger libraries

(included in constraint propagation or global optimization packages)

- Two constraint propagation / global optimization packages, from among many.
  - IBEX, a C++ library for constraint propagation. (extensively used by its developers in pattern recognition and robotics).
  - GlobSol, primarily a Fortran 90 library, with some C / C++ and a Matlab interface. (This is my work – it is somewhat dated and not optimally efficient, but certainly available.)
- Some general toolboxes.
  - INTLAB, a very popular Matlab toolbox for interval computations.
  - See IEEE-1788-compliant packages in the following.

IEEE 1788-2015

Standard for Interval Arithmetic

- Defines basic interval arithmetic, specifying accuracy, required elementary functions, etc.

IEEE 1788-2015

Standard for Interval Arithmetic

- Defines basic interval arithmetic, specifying accuracy, required elementary functions, etc.
- Defines an optional binding to the IEEE 754-2008 standard for floating point arithmetic.

IEEE 1788-2015

Standard for Interval Arithmetic

- Defines basic interval arithmetic, specifying accuracy, required elementary functions, etc.
- Defines an optional binding to the IEEE 754-2008 standard for floating point arithmetic.
- Specifies how extended interval arithmetic is handled, from various special cases.

IEEE 1788-2015

Standard for Interval Arithmetic

- Defines basic interval arithmetic, specifying accuracy, required elementary functions, etc.
- Defines an optional binding to the IEEE 754-2008 standard for floating point arithmetic.
- Specifies how extended interval arithmetic is handled, from various special cases.

Example (The underlying set is $\mathbb{R}$, not $\overline{\mathbb{R}}$.)

$$\left[\frac{1}{2},\infty\right) \leftarrow \frac{[2,3]}{[0,4]}.$$

IEEE 1788-2015

Standard for Interval Arithmetic

- Defines basic interval arithmetic, specifying accuracy, required elementary functions, etc.
- Defines an optional binding to the IEEE 754-2008 standard for floating point arithmetic.
- Specifies how extended interval arithmetic is handled, from various special cases.

Example (The underlying set is $\mathbb{R}$, not $\overline{\mathbb{R}}$.)

$$\left[\frac{1}{2}, \infty\right) \leftarrow \frac{[2,3]}{[0,4]}.$$

- Contains a decoration system for tracking continuity of an expression, if extended interval arithmetic has been used, etc. This can be viewed as a generalization of IEEE 754 exception handling.

Conforming

Gnu Octave (Matlab-like) by Oliver Heimlich.

See `http://octave.sourceforge.net/interval/`

Conforming

Gnu Octave (Matlab-like) by Oliver Heimlich.

See `http://octave.sourceforge.net/interval/`

`JInterval` (Java) by Dmitry Nadezhin and Sergei Zhilin.

See `https://java.net/projects/jinterval`

Conforming

Gnu Octave (Matlab-like) by Oliver Heimlich.
See `http://octave.sourceforge.net/interval/`

`JInterval` (Java) by Dmitry Nadezhin and Sergei Zhilin.
See `https://java.net/projects/jinterval`

C++ by Marco Nehmeier (J. Wolff v. Gudenberg).
See `https://github.com/nehmeier/libieeep1788`

IEEE 1788-2015 Standard
Implementations

Conforming

Gnu Octave (Matlab-like) by Oliver Heimlich.
See `http://octave.sourceforge.net/interval/`

`JInterval` (Java) by Dmitry Nadezhin and Sergei Zhilin.
See `https://java.net/projects/jinterval`

C++ by Marco Nehmeier (J. Wolff v. Gudenberg).
See `https://github.com/nehmeier/libieeep1788`

Conformance in Progress

`ValidatedNumerics.jl` (Julia) by David P. Sanders and Luis Benet (UNAM)
See `https://github.com/dpsanders/ValidatedNumerics.jl`

- In many applications, full implementation of the IEEE standard is not needed.

Interval Arithmetic Packages

Good non-conforming packages

- In many applications, full implementation of the IEEE standard is not needed.

  For example, in simple calculations, the decorations (exception handling) and extended arithmetic would not play a part.

Interval Arithmetic Packages

Good non-conforming packages

- In many applications, full implementation of the IEEE standard is not needed.

  For example, in simple calculations, the decorations (exception handling) and extended arithmetic would not play a part.
- For efficiency, ease of implementation, or other reasons, some packages are not totally conforming.

  Siegfried Rump's INTLAB is one widely-used such package.

Further Information

Feel free to contact me

Further Information

Feel free to contact me

- for details,

Further Information

Feel free to contact me

- for details,
- for advice,

Further Information

Feel free to contact me

- for details,
- for advice,
- for references,

Further Information

Feel free to contact me

- for details,
- for advice,
- for references,
- to tell me about your work.

Further Information

Feel free to contact me

- for details,
- for advice,
- for references,
- to tell me about your work.

See `interval.louisiana.edu/kearfott.html`

Further Information

Feel free to contact me

- for details,
- for advice,
- for references,
- to tell me about your work.

See `interval.louisiana.edu/kearfott.html`

Thank you!

Further Information

Feel free to contact me

- for details,
- for advice,
- for references,
- to tell me about your work.

See `interval.louisiana.edu/kearfott.html`

Thank you!

Questions?