(slide 1)

Mainstream Contributions of Interval Computations in Engineering and Scientific Computing

R. Baker Kearfott

Department of Mathematics
University of Louisiana at Lafayette

November 13, 2008

Outline

Outline

Basic Tasks Intervals Might Accomplish

Account for uncertainty in measurements What range of outputs is expected from a range of inputs?

Account for roundoff error with mathematical rigor Provide numerical output with the certainty of a mathematical proof.

Compute bounds on ranges Lower bounds and upper bounds on quantities might be computed easily.

Handle multiple-valued quantities simply This includes direct computation with generalized gradients of non-smooth functions.

Basic Tasks Intervals Might Accomplish

Account for uncertainty in measurements What range of outputs is expected from a range of inputs?

Account for roundoff error with mathematical rigor Provide numerical output with the certainty of a mathematical proof.

Compute bounds on ranges Lower bounds and upper bounds on quantities might be computed easily.

Handle multiple-valued quantities simply This includes direct computation with generalized gradients of non-smooth functions.

Basic Tasks Intervals Might Accomplish

Account for uncertainty in measurements What range of outputs is expected from a range of inputs?

Account for roundoff error with mathematical rigor Provide numerical output with the certainty of a mathematical proof.

Compute bounds on ranges Lower bounds and upper bounds on quantities might be computed easily.

Handle multiple-valued quantities simply This includes direct computation with generalized gradients of non-smooth functions.

Basic Tasks Intervals Might Accomplish

Account for uncertainty in measurements What range of outputs is expected from a range of inputs?

Account for roundoff error with mathematical rigor Provide numerical output with the certainty of a mathematical proof.

Compute bounds on ranges Lower bounds and upper bounds on quantities might be computed easily.

Handle multiple-valued quantities simply This includes direct computation with generalized gradients of non-smooth functions.

Outline

Early Motivations for Interval Arithmetic

Prior to the Age of Digital Computation

The same basic interval operations were described in all of these works, but with somewhat different motivations. All of this early work is apparently independent.

Rosaline Cecily Young (*Mathematische Annalen*, 1931)
"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[1]. Describing ranges and encompassing roundoff error does not seem to have been the primary motivation.

---

[1] Such limits might occur in generalized gradients of non-smooth functions.

Early Motivations for Interval Arithmetic

From the Onset of the Age of Digital Computers

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)
"Calculus of Approximations." The motivation is apparently to provide a sound theoretical backing to numerical computation.

Early Motivations for Interval Arithmetic

From the Onset of the Age of Digital Computers

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)
"Calculus of Approximations." The motivation is apparently to provide a sound theoretical backing to numerical computation.

Early Motivations for Interval Arithmetic

From the Onset of the Age of Digital Computers (continued)

Teruro Sunaga (*RAAG Memoirs*, 1958) "Theory of an Interval Algebra and its Application to Numerical Analysis." Much of what is in Moore's initial work appears independently here.

Early Motivations for Interval Arithmetic

From the Onset of the Age of Digital Computers (continued)

Ray Moore (*Lockheed Technical Report*, 1959) "Automatic Error Analysis in Digital Computation." The motivation is given in the title. The basic operations are given in this monograph, and development of numerical solution of ODE's, numerical integration, etc. based on intervals is in Moore's 1962 dissertation. It is made clear that interval computations promise rigorous bounds on the exact result, even when finite (rounded) computer arithmetic is used.

References are from the interval computations website, at `http://www.cs.utep.edu/interval-comp`

The Basic Operations of Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

The Basic Operations of Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}\,.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

The Basic Operations of Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}\,.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

The Basic Operations of Interval Arithmetic

- Each basic interval operation $\odot \in \{+, -, \times, \div, \text{etc.}\}$ is defined by

$$\boldsymbol{x} \odot \boldsymbol{y} = \{x \odot y \mid x \in \boldsymbol{x} \text{ and } y \in \boldsymbol{y}\}.$$

- This definition can be made operational; for example, for $\boldsymbol{x} = [\underline{x}, \overline{x}]$ and $\boldsymbol{y} = [\underline{y}, \overline{y}]$, $\boldsymbol{x} + \boldsymbol{y} = [\underline{x} + \underline{y}, \overline{x} + \overline{y}]$; similarly, ranges of functions such as sin, exp can be computed.
- Evaluation of an expression with this interval arithmetic gives *bounds* on the range of the expression.
- With *directed rounding* (e.g. using IEEE standard arithmetic), the computer can give mathematically rigorous bounds on ranges.

Interval Operations

Advantages

- Computing sharp bounds on ranges is an NP-hard problem.
- Computing range bounds with interval computations is quick and simple.
- The range bounds get sharper asymptotically as the widths of the domain intervals tend to zero, and, with *second-order extensions*, will do so rapidly.

Interval Operations

Advantages

- Computing sharp bounds on ranges is an NP-hard problem.
- Computing range bounds with interval computations is quick and simple.
- The range bounds get sharper asymptotically as the widths of the domain intervals tend to zero, and with *second-order extensions*, will do so rapidly.

Interval Operations

Advantages

- Computing sharp bounds on ranges is an NP-hard problem.
- Computing range bounds with interval computations is quick and simple.
- The range bounds get sharper asymptotically as the widths of the domain intervals tend to zero, and, with *second-order extensions*, will do so rapidly.

Interval Operations

Pitfalls

- The interval values contain the actual ranges, but are possibly significantly larger. For example, 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 $[-1,3]$.

- However, if we write f equivalently as $f(x) = x^2 - 1$, and we suppose we compute the range of x^2 exactly, we obtain

$$\boldsymbol{f}([-2,2]) = [-2,2]^2 - 1 = [0,4] - 1 = [-1,3],$$

the exact range.

Interval Operations

Pitfalls

- The interval values contain the actual ranges, but are possibly significantly larger. For example, 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 $[-1,3]$.

- However, if we write f equivalently as $f(x) = x^2 - 1$, and we suppose we compute the range of x^2 exactly, we obtain

$$\boldsymbol{f}([-2,2]) = [-2,2]^2 - 1 = [0,4] - 1 = [-1,3],$$

  the exact range.

Dependency and Overestimation

Consequences

- This range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- Naively converting a floating point code to interval computation by simply changing the data types is far more likely to fail than succeed.
- However, there are some notable successes.

Dependency and Overestimation

Consequences

- This range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- Naively converting a floating point code to interval computation by simply changing the data types is far more likely to fail than succeed.
- However, there are some notable successes.

Dependency and Overestimation

Consequences

- This range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- Naively converting a floating point code to interval computation by simply changing the data types is far more likely to fail than succeed.
- However, there are some notable successes.

Dependency and Overestimation

Consequences

- This range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- Naively converting a floating point code to interval computation by simply changing the data types is far more likely to fail than succeed.
- However, there are some notable successes.

Dependency and Overestimation

Consequences

- This range overestimation above is caused by the arithmetic not taking account of the fact that, when $x = 2$ in $(x + 1)$, x must also equal 2 in $(x - 1)$.
- This phenomenon is at the root of many failures of interval arithmetic.
- For this reason, interval arithmetic should be used with skill, only in appropriate places.
- Naively converting a floating point code to interval computation by simply changing the data types is far more likely to fail than succeed.
- However, there are some notable successes.

Outline

Fast Bounds on Ranges

A *Filter* in Branch and Bound Methods

- In branch and bound methods for global optimization, methods are needed for rejecting regions $\boldsymbol{x}$ that cannot contain global optimizing points.
- $\boldsymbol{x}$ can contain a global optimizing point only if
  - It contains points x that satisfy constraints $c_i(x) = 0$ and $g_j(x) \leq 0$.
  - It contains points x such that the quantity φ to be optimized obeys $\varphi(x) \leq \overline{\varphi}$, where $\overline{\varphi}$ is a previously computed upper bound on the global optimum value.
- Regions $\boldsymbol{x}$ can sometimes be quickly eliminated by evaluating φ, the c_i, and the g_j and checking violation of the conditions. For example, if the lower bound on one of the $g_j(\boldsymbol{x})$ is greater than 0, then $\boldsymbol{x}$ may be eliminated from further consideration.
- This technique is widely acknowledged in the general literature and used in leading commercial software.

Fast Bounds on Ranges

A *Filter* in Branch and Bound Methods

- In branch and bound methods for global optimization, methods are needed for rejecting regions $\boldsymbol{x}$ that cannot contain global optimizing points.
- $\boldsymbol{x}$ can contain a global optimizing point only if
  - It contains points x that satisfy constraints $c_i(x) = 0$ and $g_j(x) \le 0$.
  - It contains points x such that the quantity φ to be optimized obeys $\varphi(x) \le \overline{\varphi}$, where $\overline{\varphi}$ is a previously computed upper bound on the global optimum value.
- Regions $\boldsymbol{x}$ can sometimes be quickly eliminated by evaluating φ, the c_i, and the g_j and checking violation of the conditions. For example, if the lower bound on one of the $g_j(\boldsymbol{x})$ is greater than 0, then $\boldsymbol{x}$ may be eliminated from further consideration.
- This technique is widely acknowledged in the general literature and used in leading commercial software.

Fast Bounds on Ranges

A *Filter* in Branch and Bound Methods

- In branch and bound methods for global optimization, methods are needed for rejecting regions $\boldsymbol{x}$ that cannot contain global optimizing points.
- $\boldsymbol{x}$ can contain a global optimizing point only if
  - It contains points x that satisfy constraints $c_i(x) = 0$ and $g_j(x) \leq 0$.
  - It contains points x such that the quantity φ to be optimized obeys $\varphi(x) \leq \overline{\varphi}$, where $\overline{\varphi}$ is a previously computed upper bound on the global optimum value.
- Regions $\boldsymbol{x}$ can sometimes be quickly eliminated by evaluating φ, the c_i, and the g_j and checking violation of the conditions. For example, if the lower bound on one of the $g_j(\boldsymbol{x})$ is greater than 0, then $\boldsymbol{x}$ may be eliminated from further consideration.
- This technique is widely acknowledged in the general literature and used in leading commercial software.

Fast Bounds on Ranges

A *Filter* in Branch and Bound Methods

- In branch and bound methods for global optimization, methods are needed for rejecting regions $\boldsymbol{x}$ that cannot contain global optimizing points.
- $\boldsymbol{x}$ can contain a global optimizing point only if
  - It contains points x that satisfy constraints $c_i(x) = 0$ and $g_j(x) \le 0$.
  - It contains points x such that the quantity φ to be optimized obeys $\varphi(x) \le \overline{\varphi}$, where $\overline{\varphi}$ is a previously computed upper bound on the global optimum value.
- Regions $\boldsymbol{x}$ can sometimes be quickly eliminated by evaluating φ, the c_i, and the g_j and checking violation of the conditions. For example, if the lower bound on one of the $g_j(\boldsymbol{x})$ is greater than 0, then $\boldsymbol{x}$ may be eliminated from further consideration.
- This technique is widely acknowledged in the general literature and used in leading commercial software.

Use in Branch and Bound Methods

Constraint Propagation

- This is based on a simple idea: Solve a relation in many variables for one of the variables, then plug in bounds on the other constraints to compute new bounds on the original constraint.
- The technique is the foundation of an entire field (constraint logic programming).
- The technique is incorporated into leading commercial global optimization software (BARON).

Use in Branch and Bound Methods

Constraint Propagation

- This is based on a simple idea: Solve a relation in many variables for one of the variables, then plug in bounds on the other constraints to compute new bounds on the original constraint.
- The technique is the foundation of an entire field (constraint logic programming).
- The technique is incorporated into leading commercial global optimization software (BARON).

Use in Branch and Bound Methods

Constraint Propagation

- This is based on a simple idea: Solve a relation in many variables for one of the variables, then plug in bounds on the other constraints to compute new bounds on the original constraint.
- The technique is the foundation of an entire field (constraint logic programming).
- The technique is incorporated into leading commercial global optimization software (BARON).

Constraint Propagation

A Simple Illustrative Example

Consider

$$\begin{aligned} &\text{minimize } \varphi(x) = x_1^2 - x_2^2 \\ &\text{subject to } x_1^2 + x_2^2 = 1, \\ &\qquad\qquad x_1 + x_2 \le 0. \end{aligned}$$

- Suppose we have already found the feasible point $\hat{x} = (0, -1)$ with $\varphi(\hat{x}) = -1$, so -1 is an upper bound on the optimum.
- Suppose we are searching in the box $([-1, 1], [-1, 1])$.
- Using the upper bound $\overline{\varphi}$ gives

$$x_1^2 - x_2^2 \le -1.$$

- Solving this for x_1 gives $\cdots$

Constraint Propagation

A Simple Illustrative Example

Consider

$$\begin{aligned} &\text{minimize } \varphi(x) = x_1^2 - x_2^2 \\ &\text{subject to } x_1^2 + x_2^2 = 1, \\ &\qquad\qquad x_1 + x_2 \le 0. \end{aligned}$$

- Suppose we have already found the feasible point $\hat{x} = (0, -1)$ with $\varphi(\hat{x}) = -1$, so -1 is an upper bound on the optimum.
- Suppose we are searching in the box $([-1, 1], [-1, 1])$.
- Using the upper bound $\overline{\varphi}$ gives

$$x_1^2 - x_2^2 \le -1.$$

- Solving this for x_1 gives $\cdots$

Constraint Propagation

A Simple Illustrative Example

Consider

$$\begin{aligned} &\text{minimize } \varphi(x) = x_1^2 - x_2^2 \\ &\text{subject to } x_1^2 + x_2^2 = 1, \\ &\qquad\qquad x_1 + x_2 \le 0. \end{aligned}$$

- Suppose we have already found the feasible point $\hat{x} = (0, -1)$ with $\varphi(\hat{x}) = -1$, so -1 is an upper bound on the optimum.
- Suppose we are searching in the box $([-1, 1], [-1, 1])$.
- Using the upper bound $\overline{\varphi}$ gives

$$x_1^2 - x_2^2 \le -1.$$

- Solving this for x_1 gives $\cdots$

Constraint Propagation

A Simple Illustrative Example

Consider

$$\begin{aligned} &\text{minimize } \varphi(x) = x_1^2 - x_2^2 \\ &\text{subject to } x_1^2 + x_2^2 = 1, \\ &\qquad\qquad\quad x_1 + x_2 \leq 0. \end{aligned}$$

- Suppose we have already found the feasible point $\hat{x} = (0, -1)$ with $\varphi(\hat{x}) = -1$, so -1 is an upper bound on the optimum.
- Suppose we are searching in the box $([-1, 1], [-1, 1])$.
- Using the upper bound $\overline{\varphi}$ gives

$$x_1^2 - x_2^2 \leq -1.$$

- Solving this for x_1 gives $\cdots$

Constraint Propagation

(Simple Example, Continued)

- (solving for x_1 in the objective condition)

$$x_1 \le \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}$$

and

$$x_1 \ge \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}.$$

- Here, it is appropriate to interpret $\sqrt{[-1,0]} = 0$, so we obtain $x_1 = 0$.
- We now solve for x_2 in $x_1^2 + x_2^2 = 1$ and plug in $x_1 = 0$ to get

$$x_2 = 1 \quad \text{or} \quad x_2 = -1,$$

- Plugging $x_1 = 0$, $x_2 = 1$ into $x_1 + x_2 \le 0$ gives a contradiction, leading to the unique point $x = (0, -1)$ in $([-1,1], [-1,1])$ that can be a global optimizer.

Constraint Propagation

(Simple Example, Continued)

- (solving for x_1 in the objective condition)

$$x_1 \le \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}$$

and

$$x_1 \ge \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}.$$

- Here, it is appropriate to interpret $\sqrt{[-1,0]} = 0$, so we obtain $x_1 = 0$.
- We now solve for x_2 in $x_1^2 + x_2^2 = 1$ and plug in $x_1 = 0$ to get

$$x_2 = 1 \quad \text{or} \quad x_2 = -1,$$

- Plugging $x_1 = 0$, $x_2 = 1$ into $x_1 + x_2 \le 0$ gives a contradiction, leading to the unique point $x = (0, -1)$ in $([-1,1], [-1,1])$ that can be a global optimizer.

Constraint Propagation

(Simple Example, Continued)

- (solving for x_1 in the objective condition)

$$x_1 \le \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}$$

and

$$x_1 \ge \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}.$$

- Here, it is appropriate to interpret $\sqrt{[-1,0]} = 0$, so we obtain $x_1 = 0$.
- We now solve for x_2 in $x_1^2 + x_2^2 = 1$ and plug in $x_1 = 0$ to get

$$x_2 = 1 \quad \text{or} \quad x_2 = -1,$$

- Plugging $x_1 = 0$, $x_2 = 1$ into $x_1 + x_2 \le 0$ gives a contradiction, leading to the unique point $x = (0, -1)$ in $([-1,1],[-1,1])$ that can be a global optimizer.

Constraint Propagation

(Simple Example, Continued)

- (solving for x_1 in the objective condition)

$$x_1 \le \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}$$

and

$$x_1 \ge \sqrt{[-1,1]^2 - 1} = \sqrt{[0,1] - 1} = \sqrt{[-1,0]}.$$

- Here, it is appropriate to interpret $\sqrt{[-1,0]} = 0$, so we obtain $x_1 = 0$.
- We now solve for x_2 in $x_1^2 + x_2^2 = 1$ and plug in $x_1 = 0$ to get

$$x_2 = 1 \quad \text{or} \quad x_2 = -1,$$

- Plugging $x_1 = 0$, $x_2 = 1$ into $x_1 + x_2 \le 0$ gives a contradiction, leading to the unique point $x = (0, -1)$ in $([-1,1],[-1,1])$ that can be a global optimizer.

Outline

Verified Paths

COSY Infinity and Taylor Arithmetic

- Martin Berz and Kyoko Makino have included these methods in the *COSY Infinity* software for modeling beams in particle accelerators. This software is used by thousands of beam theorists worldwide.
- The techniques have been used to predict
  - Bounds on orbits of near-Earth objects (proving they will not hit the Earth),
  - Bounds on paths of particles in actual and planned particle accelerators (proving feasibility before expensive accelerators are built).
- Berz and Makino are recipients of the 2008 Moore Prize for best paper on applications of interval analysis.

Verified Paths

COSY Infinity and Taylor Arithmetic

- Martin Berz and Kyoko Makino have included these methods in the *COSY Infinity* software for modeling beams in particle accelerators. This software is used by thousands of beam theorists worldwide.
- The techniques have been used to predict
  - Bounds on orbits of near-Earth objects (proving they will not hit the Earth),
  - Bounds on paths of particles in actual and planned particle accelerators (proving feasibility before expensive accelerators are built).
- Berz and Makino are recipients of the 2008 Moore Prize for best paper on applications of interval analysis.

Verified Paths

COSY Infinity and Taylor Arithmetic

- Martin Berz and Kyoko Makino have included these methods in the *COSY Infinity* software for modeling beams in particle accelerators. This software is used by thousands of beam theorists worldwide.
- The techniques have been used to predict
  - Bounds on orbits of near-Earth objects (proving they will not hit the Earth),
  - Bounds on paths of particles in actual and planned particle accelerators (proving feasibility before expensive accelerators are built).
- Berz and Makino are recipients of the 2008 Moore Prize for best paper on applications of interval analysis.

Verified Paths

COSY Infinity and Taylor Arithmetic

- Martin Berz and Kyoko Makino have included these methods in the *COSY Infinity* software for modeling beams in particle accelerators. This software is used by thousands of beam theorists worldwide.
- The techniques have been used to predict
  - Bounds on orbits of near-Earth objects (proving they will not hit the Earth),
  - Bounds on paths of particles in actual and planned particle accelerators (proving feasibility before expensive accelerators are built).
- Berz and Makino are recipients of the 2008 Moore Prize for best paper on applications of interval analysis.

Verified Paths

COSY Infinity and Taylor Arithmetic

- Martin Berz and Kyoko Makino have included these methods in the *COSY Infinity* software for modeling beams in particle accelerators. This software is used by thousands of beam theorists worldwide.
- The techniques have been used to predict
  - Bounds on orbits of near-Earth objects (proving they will not hit the Earth),
  - Bounds on paths of particles in actual and planned particle accelerators (proving feasibility before expensive accelerators are built).
- Berz and Makino are recipients of the 2008 Moore Prize for best paper on applications of interval analysis.

Other Verified ODE Solutions

Chemical Engineering and Biological Models

- Mark Stadtherr and his students (originally Youdong Lin) have produced success in using interval techniques (with Taylor models) to find the range of responses to systems, subject to initial conditions that vary.

Outline

Computer-Aided Proofs

Proofs of Some Famous Mathematical Conjectures

Chaos in the Lorenz equations Warwick Tucker used intervals to prove that the Lorenz equations (a simple model of atmospheric circulation) have a strange attractor (and hence behave chaotically). Warwick received the 2002 Moore Prize for this work.

Proof of the Kepler Conjecture Exhibit the way to arrange spheres in space to maximize the ratio of filled to unfilled space, and prove it is optimal. This was done by Thomas Hales, who received the 2004 Moore prize for his success.

Computer-Aided Proofs

Proofs of Some Famous Mathematical Conjectures

Chaos in the Lorenz equations Warwick Tucker used intervals to prove that the Lorenz equations (a simple model of atmospheric circulation) have a strange attractor (and hence behave chaotically). Warwick received the 2002 Moore Prize for this work.

Proof of the Kepler Conjecture Exhibit the way to arrange spheres in space to maximize the ratio of filled to unfilled space, and prove it is optimal. This was done by Thomas Hales, who received the 2004 Moore prize for his success.

Outline

Use in Diverse Applications

(with the INTLAB toolbox)

- Siegfried Rump has maintained a high-quality MATLAB toolbox INTLAB for interval computations.
- The widespread availability of this toolbox has stimulated its use in diverse applications, both within and outside the area of interval computations research.
- Siegfried has compiled a list of about 200 salient references that use INTLAB to obtain results, with many science and engineering fields represented. See `http://www.ti3.tu-harburg.de/rump/intlab/INTLABref.pdf`

Use in Diverse Applications

(with the INTLAB toolbox)

- Siegfried Rump has maintained a high-quality MATLAB toolbox INTLAB for interval computations.
- The widespread availability of this toolbox has stimulated its use in diverse applications, both within and outside the area of interval computations research.
- Siegfried has compiled a list of about 200 salient references that use INTLAB to obtain results, with many science and engineering fields represented. See `http://www.ti3.tu-harburg.de/rump/intlab/INTLABref.pdf`

Use in Diverse Applications

(with the INTLAB toolbox)

- Siegfried Rump has maintained a high-quality MATLAB toolbox INTLAB for interval computations.
- The widespread availability of this toolbox has stimulated its use in diverse applications, both within and outside the area of interval computations research.
- Siegfried has compiled a list of about 200 salient references that use INTLAB to obtain results, with many science and engineering fields represented. See `http://www.ti3.tu-harburg.de/rump/intlab/INTLABref.pdf`

Outline

Molecular Properties

Minimum-energy spatial conformations

- Mark Stadtherr and his students have used interval arithmetic with global optimization techniques to obtain minimum-energy atomic configurations and prove that they are correct.
- The work has resulted in significant revision of tables of molecular properties.
- This has won Mark a prize from a chemical engineering society.

Molecular Properties

Minimum-energy spatial conformations

- Mark Stadtherr and his students have used interval arithmetic with global optimization techniques to obtain minimum-energy atomic configurations and prove that they are correct.
- The work has resulted in significant revision of tables of molecular properties.
- This has won Mark a prize from a chemical engineering society.

Molecular Properties

Minimum-energy spatial conformations

- Mark Stadtherr and his students have used interval arithmetic with global optimization techniques to obtain minimum-energy atomic configurations and prove that they are correct.
- The work has resulted in significant revision of tables of molecular properties.
- This has won Mark a prize from a chemical engineering society.

Outline

Robotics

Design and Kinematics

- The nonlinear systems of equations arising in the geometric design of robot manipulators have in many cases been successfully solved with interval-based branch and bound techniques.
- The book *Applied Interval Analysis* by Jaulin, Kieffer, Didrit, and Walter explains some of these systems in a chapter.
- Jean-Pierre Merlet has achieved particular success in this area.

Robotics

Design and Kinematics

- The nonlinear systems of equations arising in the geometric design of robot manipulators have in many cases been successfully solved with interval-based branch and bound techniques.
- The book *Applied Interval Analysis* by Jaulin, Kieffer, Didrit, and Walter explains some of these systems in a chapter.
- Jean-Pierre Merlet has achieved particular success in this area.

Robotics

Design and Kinematics

- The nonlinear systems of equations arising in the geometric design of robot manipulators have in many cases been successfully solved with interval-based branch and bound techniques.
- The book *Applied Interval Analysis* by Jaulin, Kieffer, Didrit, and Walter explains some of these systems in a chapter.
- Jean-Pierre Merlet has achieved particular success in this area.

Outline

Finite Element Analysis

- Rafi Muhanna and Robert Mullen have put forward very promising methods for structural analysis that use interval methods to take account of measurement uncertainties and tolerances in components.
- They have founded the Center for Reliable Engineering Computing, where work on techniques and applications continues.
- A focus is the analysis of truss structures.
- Various others have also contributed to this area.

Finite Element Analysis

- Rafi Muhanna and Robert Mullen have put forward very promising methods for structural analysis that use interval methods to take account of measurement uncertainties and tolerances in components.
- They have founded the Center for Reliable Engineering Computing, where work on techniques and applications continues.
- A focus is the analysis of truss structures.
- Various others have also contributed to this area.

Finite Element Analysis

- Rafi Muhanna and Robert Mullen have put forward very promising methods for structural analysis that use interval methods to take account of measurement uncertainties and tolerances in components.
- They have founded the Center for Reliable Engineering Computing, where work on techniques and applications continues.
- A focus is the analysis of truss structures.
- Various others have also contributed to this area.

Finite Element Analysis

- Rafi Muhanna and Robert Mullen have put forward very promising methods for structural analysis that use interval methods to take account of measurement uncertainties and tolerances in components.
- They have founded the Center for Reliable Engineering Computing, where work on techniques and applications continues.
- A focus is the analysis of truss structures.
- Various others have also contributed to this area.

Various PDE Problems

3D Photonic Crystals Michael Plum, "A Computer-Assisted Band-Gap Proof for 3D Photonic Crystals," and various other work.

Other PDE applications Hashimoto, Kimura, Kinoshita, Nakao, Nagatou, Tomura, Watanabe, and Yamamoto have worked on rigorous error bounding in elliptic and other PDEs.

Various PDE Problems

3D Photonic Crystals Michael Plum, "A Computer-Assisted Band-Gap Proof for 3D Photonic Crystals," and various other work.

Other PDE applications Hashimoto, Kimura, Kinoshita, Nakao, Nagatou, Tomura, Watanabe, and Yamamoto have worked on rigorous error bounding in elliptic and other PDEs.

Outline

1. The Promise of Interval Arithmetic
   - Things it Might Do
   - Early Motivations
2. How Does Interval Arithmetic Work?
3. Advantages and Pitfalls
4. Areas Currently Impacted
   - Nonlinear Programming
   - Verified ODE's
   - Automatic Theorem Proving
   - General Numerical Work
   - Chemical Engineering
   - Robotics
5. Areas With Strong Promise
   - Some PDE Problems
   - Unusual Uses of Intervals

Finance and Decision Making

- Chenyi Hu and Ling T. He have proposed a singular value analysis based on interval matrices that compares favorably to existing methods for modeling stock market indices. However, the idea is not related to rigorous enclosure of errors.

Finance and Decision Making

- Chenyi Hu and Ling T. He have proposed a singular value analysis based on interval matrices that compares favorably to existing methods for modeling stock market indices. However, the idea is not related to rigorous enclosure of errors.

(slide 76)