Interval Analysis Grading of On-Line Homework

John L. Orr

jorr@math.unl.edu

University of Nebraska–Lincoln

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU).

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU). Lecture plan:

- Describe practical problem from mathematical software development

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU). Lecture plan:

- Describe practical problem from mathematical software development
- Introduce interval analysis concepts

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU). Lecture plan:

- Describe practical problem from mathematical software development
- Introduce interval analysis concepts
- Describe our solution to the problem

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU). Lecture plan:

- Describe practical problem from mathematical software development
- Introduce interval analysis concepts
- Describe our solution to the problem
- Describe a solution to a related mathematical problem

Introduction

Joint work with Stephen Scott (UNL) and Travis Fisher (UNL & PSU). Lecture plan:

- Describe practical problem from mathematical software development
- Introduce interval analysis concepts
- Describe our solution to the problem
- Describe a solution to a related mathematical problem
- Evaluation and results

Statement of the Problem

Computer grading of students' answers to mathematical questions.

Example: In response to "Differentiate $y = xe^x$" the student enters

$$xe^x + e^x$$

but the stored answer in the question bank is

$$(1 + x)e^x$$

Are the two answers equivalent?

Context of the Problem

We were developing software for on-line delivery of student assessment.

- Support multiple choice, fill-in-blank, interactive Flash questions, etc
- Multiple, reworkable assignments with different algorithmically-generated parameters
- Vital to be able to grade mathematical questions on *content*

Context of the Problem

Our algorithms are now used in

- Brownstone's EDU,
- Wiley's eGrade,
- Prentice Hall's PHGA,
- McGraw-Hill's Netgrade and MHHM,
- Freeman's iSolve,
- Maplesoft's Maple T.A.

The Zero-Equivalence Problem

Problem: Given two functions f and g, determine whether $f(x) = g(x)\ \forall x \in \mathbb{R}$.

Equivalently, given an expression f, determine whether $f(x) = 0\ \forall x \in \mathbb{R}$.

The Zero-Equivalence Problem

Problem: Given two functions f and g, determine whether $f(x) = g(x)\ \forall x \in \mathbb{R}$.

Equivalently, given an expression f, determine whether $f(x) = 0\ \forall x \in \mathbb{R}$.

Possible solutions:

- Symbolic manipulation

The Zero-Equivalence Problem

Problem: Given two functions f and g, determine whether $f(x) = g(x)\ \forall x \in \mathbb{R}$.

Equivalently, given an expression f, determine whether $f(x) = 0\ \forall x \in \mathbb{R}$.

Possible solutions:

- Symbolic manipulation
- Numerical evaluation

The Zero-Equivalence Problem

Problem: Given two functions f and g, determine whether $f(x) = g(x)\ \forall x \in \mathbb{R}$.

Equivalently, given an expression f, determine whether $f(x) = 0\ \forall x \in \mathbb{R}$.

Possible solutions:

- Symbolic manipulation
- Numerical evaluation
- Caviness, 1970: Undecidable for functions built from $1, \pi, +, -, \times, \div, x, \sin(x), |x|$.

Monte-Carlo Methods

Method: Evaluate $f(x)$ and $g(x)$ at a set of random points and compare.

- Not always correct if $f \neq g$
- Always correct if $f = g$

Monte-Carlo Methods

Method: Evaluate $f(x)$ and $g(x)$ at a set of random points and compare.

- Not always correct if $f \neq g$
- Always correct if $f = g \ldots$ or is it?

Monte-Carlo Methods

Method: Evaluate $f(x)$ and $g(x)$ at a set of random points and compare.

- Not always correct if $f \neq g$
- Always correct if $f = g$... or is it?
- Rounding error problems

Monte-Carlo Methods

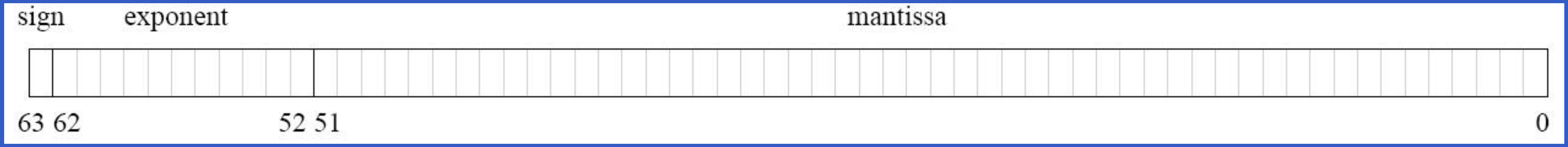
Method: Evaluate $f(x)$ and $g(x)$ at a set of random points and compare.

- Not always correct if $f \neq g$
- Always correct if $f = g \ldots$ or is it?
- Rounding error problems

One-sided error would be acceptable in this application, so we wanted to overcome rounding errors

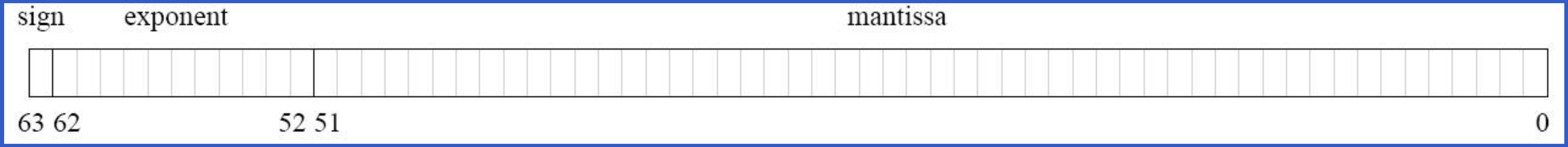
Review Floating Point Arithmetic

IEEE-754 64 bit floating point number



Review Floating Point Arithmetic

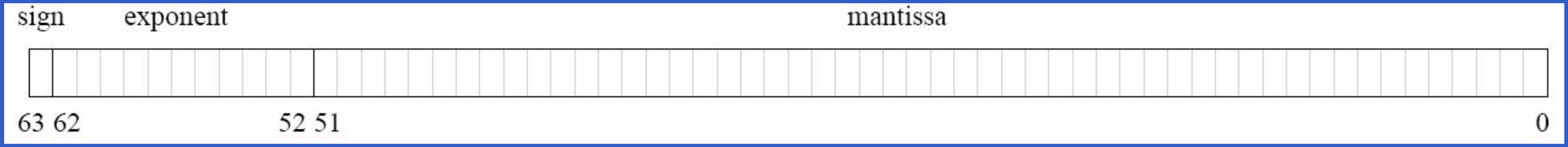
IEEE-754 64 bit floating point number



- Sign bit, $S = \{0, 1\}$

Review Floating Point Arithmetic

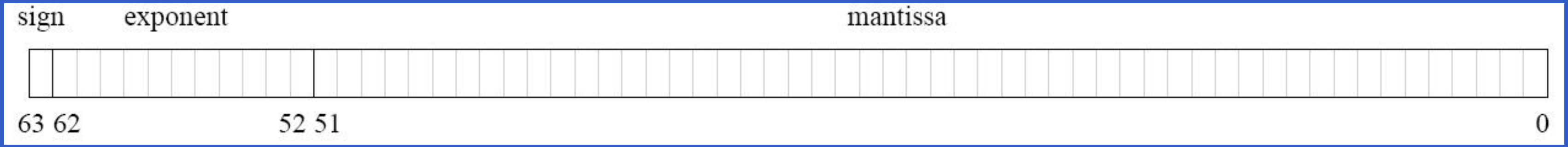
IEEE-754 64 bit floating point number



- Sign bit, $S = \{0, 1\}$
- Exponent, $0 \leq E \leq 2^{11} - 1 = 2047$

Review Floating Point Arithmetic

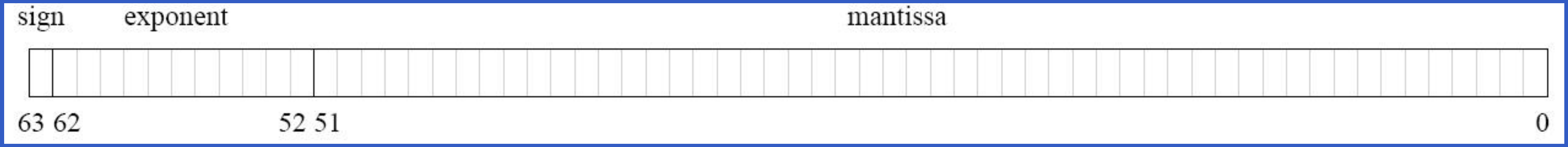
IEEE-754 64 bit floating point number



- Sign bit, $S = \{0, 1\}$
- Exponent, $0 \leq E \leq 2^{11} - 1 = 2047$
- Mantissa, $0 \leq M \leq 2^{52} - 1$

Review Floating Point Arithmetic

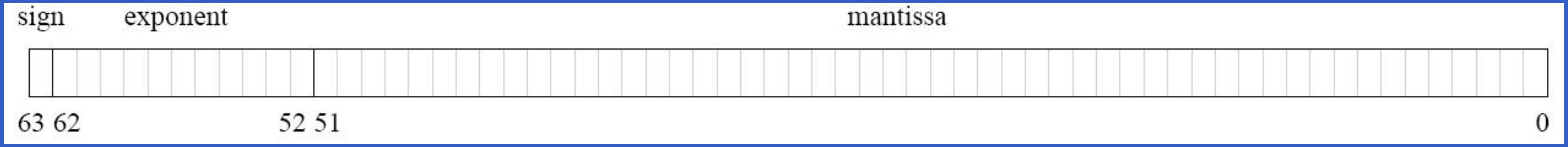
IEEE-754 64 bit floating point number



- Sign bit, $S = \{0, 1\}$
- Exponent, $0 \leq E \leq 2^{11} - 1 = 2047$
- Mantissa, $0 \leq M \leq 2^{52} - 1$
- $I = 0$ if $E = 0$; $I = 1$ otherwise

Review Floating Point Arithmetic

IEEE-754 64 bit floating point number



- Sign bit, $S = \{0, 1\}$
- Exponent, $0 \leq E \leq 2^{11} - 1 = 2047$
- Mantissa, $0 \leq M \leq 2^{52} - 1$
- $I = 0$ if $E = 0$; $I = 1$ otherwise

$$x = (-1)^S (I + 2^{-52} M) \times 2^{E-1022-I}$$

Review Floating Point Arithmetic

For example,

$$
\begin{aligned}
24.5 &= 49 \times 2^{-1} \\
&= 110001 \times 10^{-1} \\
&= 1.10001 \times 10^{5} \\
&= (-1)^{S}(1 + 2^{-52}M) \times 2^{E-1023} \\
&\mapsto \begin{array}{c|c|c} \textit{sign} & \textit{exponent} & \textit{mantissa} \\ \hline 0 & 10000000100 & 1000100000000000000000000000000000000000000000000000 \end{array}
\end{aligned}
$$

Review Floating Point Arithmetic

Rounding errors, e.g. $0.1 + 0.2 \neq 0.3$

		sign	*exponent*	*mantissa*	
0.1	$\mapsto$	0	01111111011	1001100110011001100110011001100110011001100110011010	
	$=$	$0.00011001100110011001100110011001100110011001100110011010$			(base 2)

		sign	*exponent*	*mantissa*	
0.2	$\mapsto$	0	01111111100	1001100110011001100110011001100110011001100110011010	
	$=$	$0.0011001100110011001100110011001100110011001100110011010$			(base 2)

So, adding

$$
\begin{array}{rl}
 & 0.0001\underline{1001100110011001100110011001100110011001100110011010} \\
+ & 0.001\underline{1001100110011001100110011001100110011001100110011010} \\
\hline
= & 0.01\underline{0011001100110011001100110011001100110011001100110011}10
\end{array}
$$

which is rounded to

$$0.01\underline{0011001100110011001100110011001100110011001100110100}$$

Review Floating Point Arithmetic

Rounding errors, e.g. $0.1 + 0.2 \neq 0.3$

		sign	exponent	mantissa	
$(0.1+0.2)$	$\mapsto$	0	01111111101	0011001100110011001100110011001100110011001100110100	
	$=$	$0.010011001100110011001100110011001100110011001100110100$			(base 2)

		sign	exponent	mantissa	
0.3	$\mapsto$	0	01111111101	0011001100110011001100110011001100110011001100110011	
	$=$	$0.010011001100110011001100110011001100110011001100110011$			(base 2)

These differ by one ULP.

Interval Analysis

Moore, 1966: Replace numbers with intervals

Interval Analysis

Moore, 1966: Replace numbers with intervals

$$\mathbf{x} = [x^-, y^+]$$
$$\mathbf{y} = [y^-, y^+]$$

Interval Analysis

Moore, 1966: Replace numbers with intervals

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [x^- + y^-, x^+ + y^+]$$

Interval Analysis

Moore, 1966: Replace numbers with intervals

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [x^- + y^-, x^+ + y^+]$$

$$\mathbf{x} \times \mathbf{y} = [x^- y^-, x^+ y^+] \quad (x^-, y^- \geq 0)$$

Interval Analysis

Moore, 1966: Replace numbers with intervals

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [x^- + y^-, x^+ + y^+]$$

$$\mathbf{x} \times \mathbf{y} = [x^- y^-, x^+ y^+] \quad (x^-, y^- \geq 0)$$

$$f(\mathbf{x}_1, \ldots, x_n) = \{f(x_1, \ldots, x_n) \ : \ x_i \in \mathbf{x}_i\}$$

Interval Analysis

In practice, use *rounded machine arithmetic*

Interval Analysis

In practice, use *rounded machine arithmetic*

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

Interval Analysis

In practice, use *rounded machine arithmetic*

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [\underline{x^- +_M y^-}, \overline{x^+ +_M y^+}]$$

Interval Analysis

In practice, use *rounded machine arithmetic*

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [\underline{x^- +_M y^-}, \overline{x^+ +_M y^+}]$$

$$\mathbf{x} \times \mathbf{y} = [\underline{x^- \times_M y^-}, \overline{x^+ \times_M y^+}] \quad (x^-, y^- \geq 0)$$

Interval Analysis

In practice, use *rounded machine arithmetic*

$$\mathbf{x} = [x^-, y^+]$$

$$\mathbf{y} = [y^-, y^+]$$

$$\mathbf{x} + \mathbf{y} = [\underline{x^- +_M y^-}, \overline{x^+ +_M y^+}]$$

$$\mathbf{x} \times \mathbf{y} = [\underline{x^- \times_M y^-}, \overline{x^+ \times_M y^+}] \quad (x^-, y^- \geq 0)$$

etc

Interval Analysis

Moral: Using rounded machine interval arithmetic, the true result is *always* contained in the computed interval.

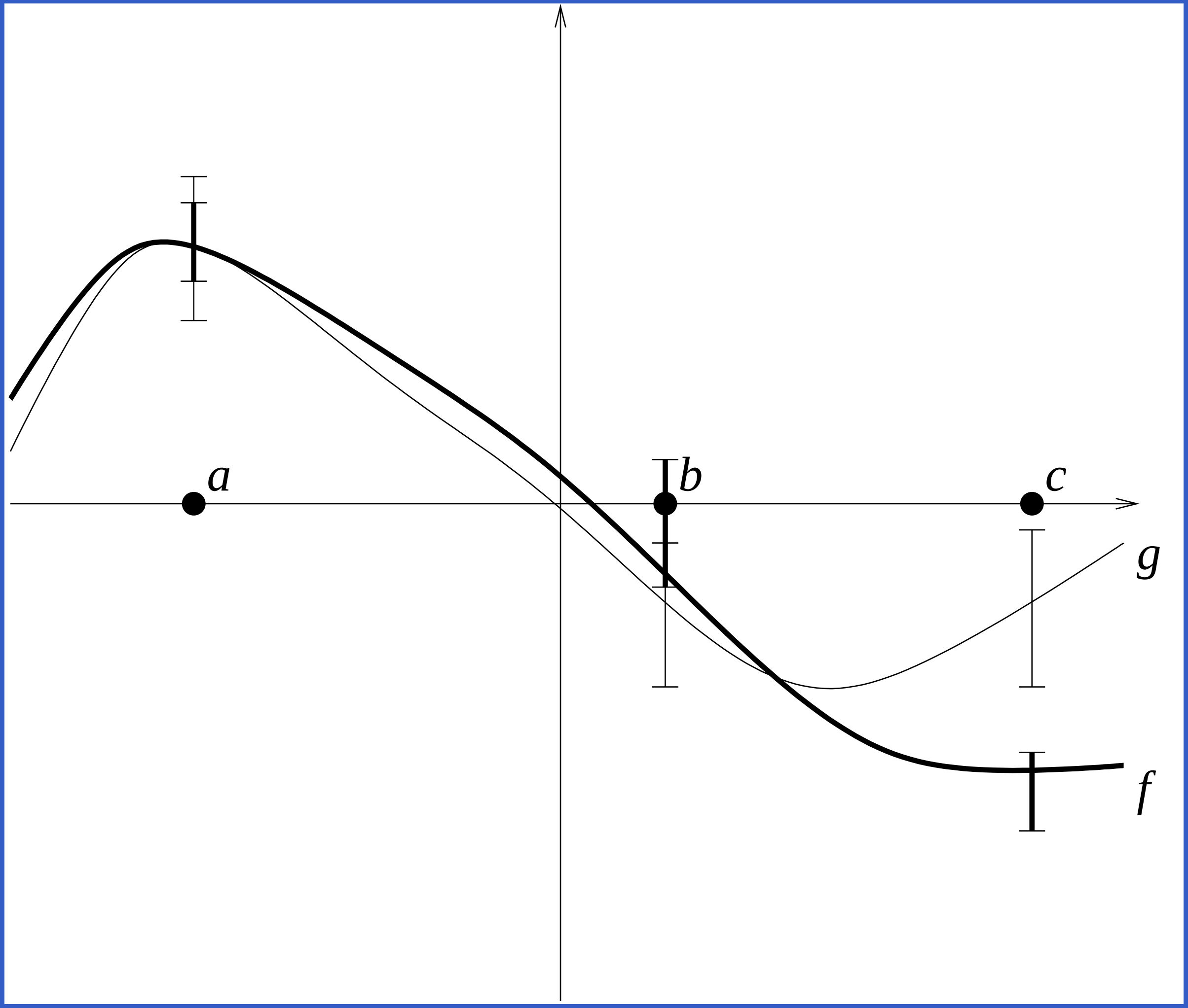
So, if the *intervals* $U = f(\mathbf{x})$ and $V = f(\mathbf{y})$ are disjoint, then f and g are *guaranteed* different.

Interval Arithmetic Solution

Algorithm 1:

```
start with TRIALS equal to 0
repeat until TRIALS > MAXTRIALS
    assign random values to each variable in f and g
    let U be the rounded interval evaluation of f under those assignments
    let V be the rounded interval evaluation of g under those assignments
    if U ∩ V = ∅
        return FALSE (the functions cannot be equal)
    increment TRIALS
return TRUE (if cannot demonstrate that f and g differ, assume they are equal)
```

Using Interval Arithmetic



A Related Problem

Determine whether $f(x)$ and $g(x)$ differ by a constant.

E.g. The student is asked to integrate $\sin(2x)$.

Integrate $\sin(2x)$

One possible route is:

$$\int \sin(2x)\,dx = -\frac{1}{2}\cos(2x) + C$$

Integrate $\sin(2x)$

One possible route is:

$$\int \sin(2x)\,dx = -\frac{1}{2}\cos(2x) + C$$

but another valid approach is

$$\begin{aligned}\int \sin(2x)\,dx &= \int 2\sin(x)\cos(x)\,dx \\ &= \int 2s\,ds = \sin^2(x) + C\end{aligned}$$

Integrate $\sin(2x)$

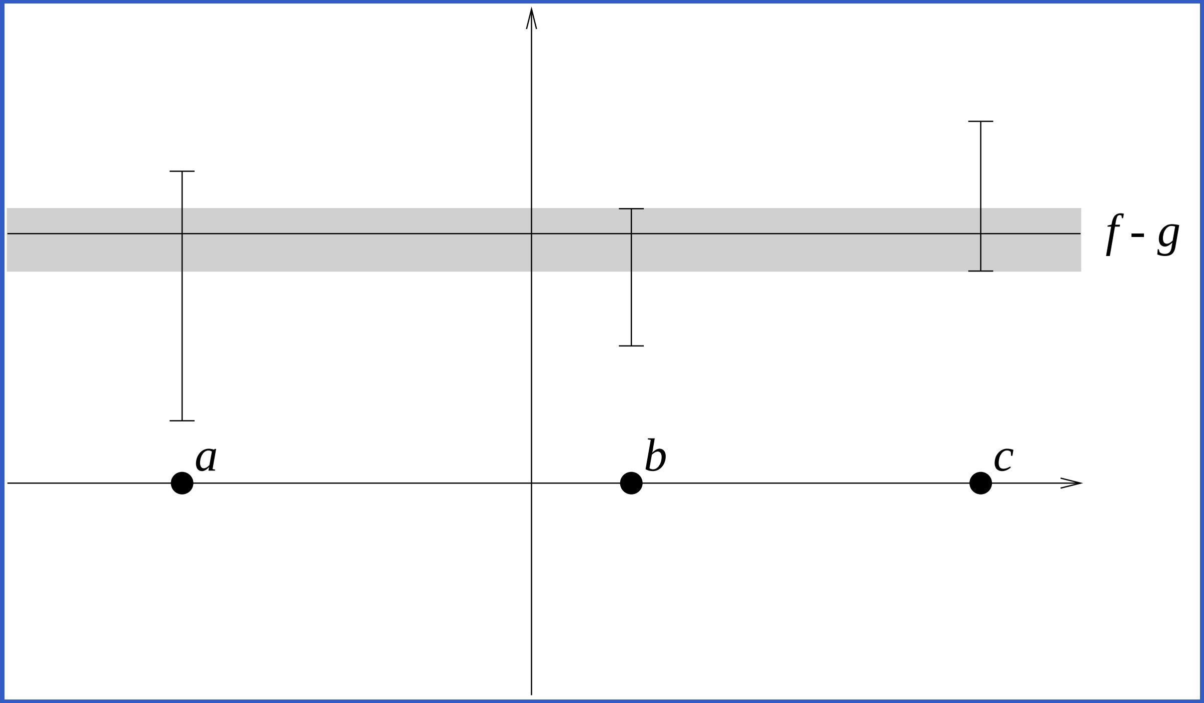
Moral: Even neglecting the constant of integration, two different approaches to integration can give answers that differ by a constant (1, in this case).

Interval Arithmetic Solution

Algorithm 2:

start with TRIALS equal to 0 and INTERSECTION equal to $[-\infty, \infty]$
repeat until TRIALS $>$ MAXTRIALS
    assign random values to each variable in f and g
    let U be the rounded interval evaluation of f under those assignments
    let V be the rounded interval evaluation of g under those assignments
    let INTERSECTION equal INTERSECTION $\cap\, (U -_M V)$
    if INTERSECTION $= \emptyset$
        return FALSE (we have found a miss)
    increment TRIALS
return TRUE (there is still a range of constants by which f and g might differ)

Interval Arithmetic Solution



Evaluation

- Evaluated using 8,000 responses to Gateway Exam questions.
- Compared student responses with “correct” responses.
- Performed same comparison using Maple’s

```
evalb(simplify(f-g)=0)
```

Evaluation

	Maple `simplify`	IA Monte-Carlo Algorithm
$f = g$	can be wrong	**always right**
$f \neq g$	**always right**	can be wrong

Evaluation

	Maple `simplify`	IA Monte-Carlo Algorithm
$f = g$	can be wrong	**always right**
$f \neq g$	**always right**	can be wrong

Moral: If the two checks agree, then they must be right!

Evaluation

- We found only 4 discrepancies out of 8,000
- Hand-verified these and found IA Monte-Carlo method was correct in all cases

Summary

IA Monte-Carlo solution to Zero Equivalence problem is:

Summary

IA Monte-Carlo solution to Zero Equivalence problem is:

- very fast

Summary

IA Monte-Carlo solution to Zero Equivalence problem is:

- very fast
- very accurate

Summary

IA Monte-Carlo solution to Zero Equivalence problem is:

- very fast
- very accurate
- with guaranteed one-sided error