# Tutorial: Exact Numerical Computation in Algebra and Geometry

Chee K. Yap

Courant Institute of Mathematical Sciences
New York University

and
Korea Institute of Advanced Study (KIAS)
Seoul, Korea

34th ISSAC, July 28–31, 2009

PART 1

# Exact Numeric Computation and the Zero Problem

*"The history of the zero recognition problem is somewhat confused by the fact that many people do not recognize it as a problem at all."*

— Daniel Richardson (1996)

*"Algebra is generous, she often gives more than is asked of her."*

— Jean Le Rond D'Alembert (1717-83)

# Coming Up Next

1. Introduction: What is Geometric Computation?
2. Five Examples of Geometric Computation
3. Exact Numeric Computation – A Synthesis
4. Exact Geometric Computation
5. Constructive Zero Bounds

# Introduction to Geometric Computation

- PUZZLE 1:
  Is Geometry discrete or continuous?
- PUZZLE 2:
  How come numbers do not arise in Computational Geometry?

# Introduction to Geometric Computation

- PUZZLE 1:
  Is Geometry discrete or continuous?
- PUZZLE 2:
  How come numbers do not arise in Computational Geometry?

# Introduction to Geometric Computation

- PUZZLE 1:
  Is Geometry discrete or continuous?
- PUZZLE 2:
  How come numbers do not arise in Computational Geometry?

# Introduction to Geometric Computation

- PUZZLE 1:
  Is Geometry discrete or continuous?
- PUZZLE 2:
  How come numbers do not arise in Computational Geometry?

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# What is Computational Geometry?

## Geometric Objects

- Prototype: Points, Lines, Circles (Euclidean Geometry)
- Arrangement of hyperplanes and hypersurfaces
- Zero sets and their Singularities
- Semi-algebraic sets
- Non-algebraic sets
- Geometric complexes

## Geometric Problems

- Constructing geometric objects
- Searching in geometric complexes or structures

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example:
  Cell Complexes (in the sense of algebraic topology)

# Dual Descriptions of Geometry

## Where do Geometric Objects Live?

- As Points in Parametric Space $\mathcal{P}$
  - E.g., for lines given by $L(a,b,c) := aX + bY + c = 0$, the space is $\mathcal{P} := \{(a,b,c) : a^2 + b^2 > 0\} \subseteq \mathbb{R}^3$.
- As Loci in Ambient Space $\mathcal{A}$
  - E.g., Locus of the $Line(1,-2,0)$ is the set $\{(x,y) \in \mathbb{R}^2 : x - 2y = 0\} \subseteq \mathcal{A} = \mathbb{R}^2$.
- More involved example: Cell Complexes (in the sense of algebraic topology)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# Computation: Geometric vs. Algebraic

## Where is the Computation?

- Algebraic Computation: in parameter space $\mathcal{P}$
  - E.g., Gröbner bases
  - Polynomial manipulation, Expensive (double exponential time)
- Geometric Computation: in ambient space $\mathcal{A}$
  - E.g., Finding Zeros of Polynomials in $\mathbb{R}^n$
  - Numerical, Combinatorial, Adaptive (single exponential time)

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: "BOTH"

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: “BOTH”

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: "BOTH"

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: “BOTH”

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: "BOTH"

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: “BOTH”

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: “BOTH”

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# (Contd.) Computation: Geometric vs. Algebraic

## Answer to PUZZLE 1: “BOTH”

- Geometry is discrete (in $\mathcal{P}$) (algebraic computation)
- Geometry is continuous (in $\mathcal{A}$) (analytic computation)

## Actions in the Ambient Space

- Geometric Relationships on different Object types arise in $\mathcal{A}$
  - E.g., Point is ON/LEFT/RIGHT of a Line
- Analytic properties of Objects comes from their loci
  - E.g., Proximity, Approximations, Isotopy, etc

# Mini Summary

- Geometry is discrete (algebraic view)
- Geometry is continuous (analytic view)
- Up Next: What do Computational Geometers think?

# Mini Summary

- Geometry is discrete (algebraic view)
- Geometry is continuous (analytic view)
- Up Next : What do Computational Geometers think?

# Mini Summary

- Geometry is discrete (algebraic view)
- Geometry is continuous (analytic view)
- Up Next: What do Computational Geometers think?

# Mini Summary

- Geometry is discrete (algebraic view)
- Geometry is continuous (analytic view)
- Up Next : What do Computational Geometers think?

# Mini Summary

- Geometry is discrete (algebraic view)
- Geometry is continuous (analytic view)
- Up Next : What do Computational Geometers think?

# Coming Up Next

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# Five Examples of Geometric Computation

- (I) Convex Hulls
- (II) Euclidean Shortest Path
- (III) Disc Avoiding Shortest Path
- (IV) Mesh Generation
- (V) Discrete Morse Theory

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope

# (I) Convex Hulls

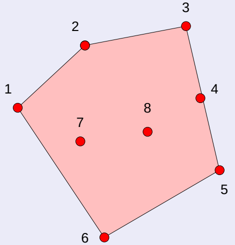
## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

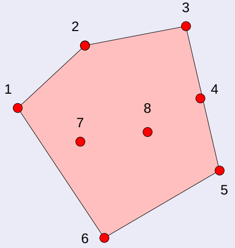
- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope



# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope

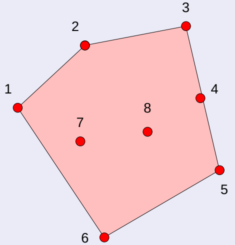



Can be reduced to a single predicate $Orientation(P_0, P_1, \ldots, P_n)$

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope



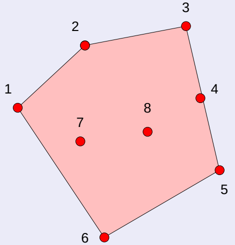

Can be reduced to a single predicate $Orientation(P_0, P_1, \ldots, P_n)$

- Main issue is combinatorial in nature

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope



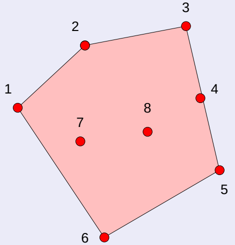

Can be reduced to a single predicate $Orientation(P_0, P_1, \ldots, P_n)$

- Main issue is combinatorial in nature

# (I) Convex Hulls

## Convex Hull of Points in $\mathbb{R}^n$

- $n = 1$: finding max and min
- $n = 2, 3$: find a convex polygon or polytope




Can be reduced to a single predicate $Orientation(P_0, P_1, \ldots, P_n)$

- Main issue is combinatorial in nature

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

# (II) Euclidean Shortest Path (ESP)

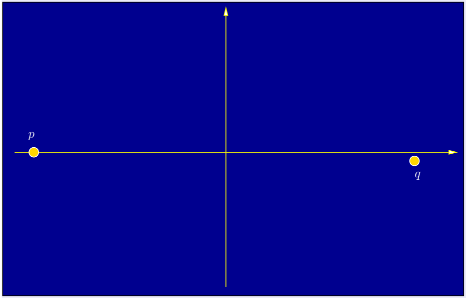
## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

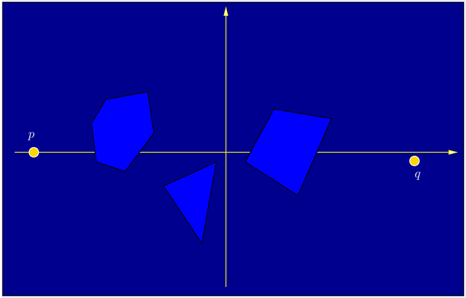
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

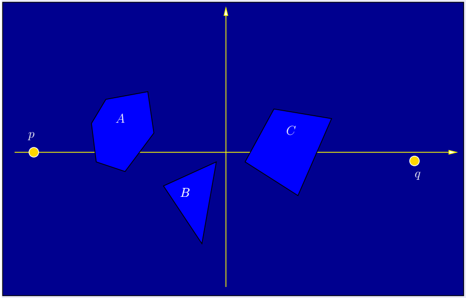
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

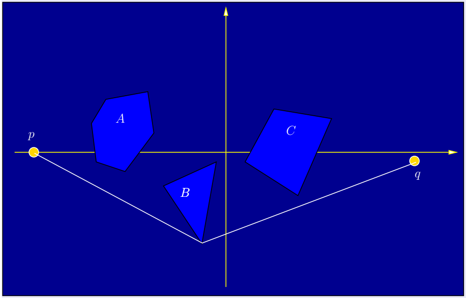
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

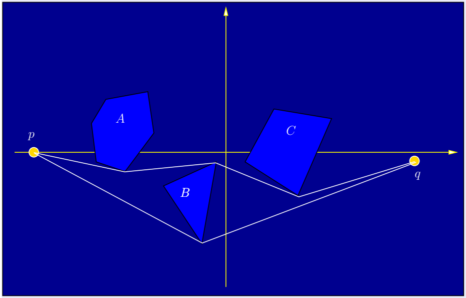
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

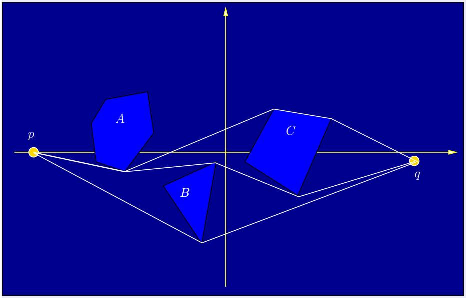
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

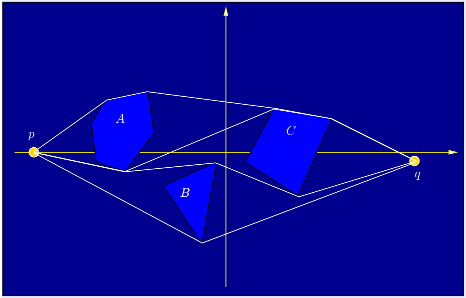
- Shortest path from $p$ to $q$ avoiding $A, B, C$



# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

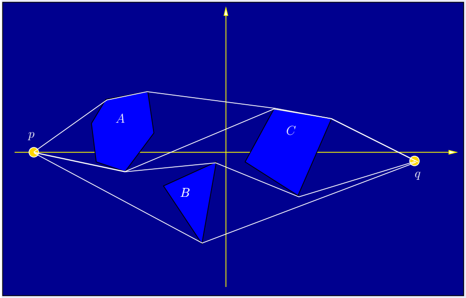



Segment length is a square-root

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$

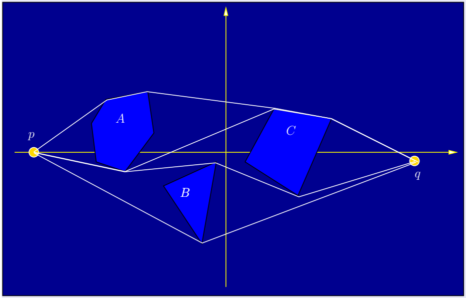



Segment length is a square-root

# (II) Euclidean Shortest Path (ESP)

## Shortest Path amidst Polygonal Obstacles

- Shortest path from $p$ to $q$ avoiding $A, B, C$




Segment length is a square-root

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- $\boxed{\text{Luck deals differently for the two approaches}}$

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- Luck deals differently for the two approaches

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- Luck deals differently for the two approaches

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- $\boxed{\text{Luck deals differently for the two approaches}}$

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- $\boxed{\text{Luck deals differently for the two approaches}}$

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0?}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- Luck deals differently for the two approaches

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0\text{?}}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- Luck deals differently for the two approaches

# ESP, contd.

## Reduction to Dijkstra's Algorithm

- Combinatorial complexity: $O(n^2 \log n)$ (negligible)
- Sum of Square-roots Problem: $\boxed{\text{Is } \sum_{i=1}^{m} a_i \sqrt{b_i} = 0\text{?}}$
- Not known to be polynomial-time!
- Algebraic Approach: Repeated Squaring Method (Nontrivial for Inequalites!)
  - $\Omega(2^m)$ (slow, unless you are lucky! (Illustrate))
- Numerical Approach: Zero Bound Method
  - $O(\log(1/|e|))$ (fast, unless you are unlucky! (Illustrate))
- $\boxed{\text{Luck deals differently for the two approaches}}$

# (III) Shortest Path Amidst Discs

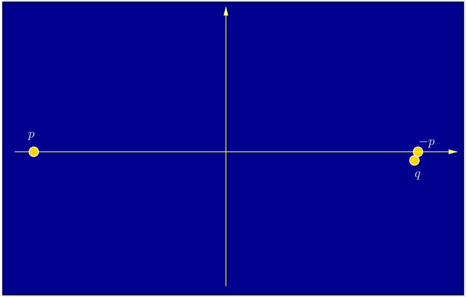
## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$

# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

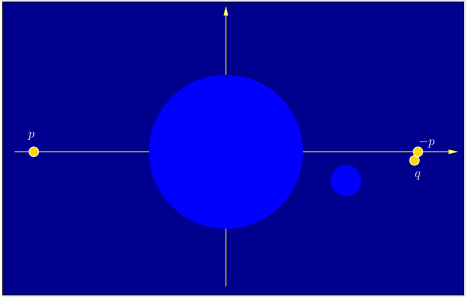
- Shortest path from $p$ to $q$ avoiding discs $A, B$



# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

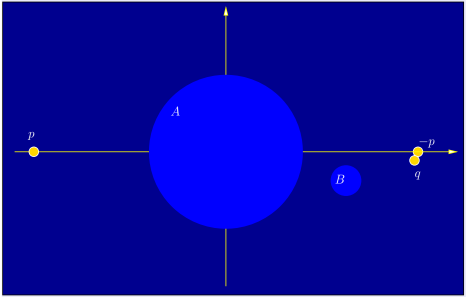
- Shortest path from $p$ to $q$ avoiding discs $A, B$



# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

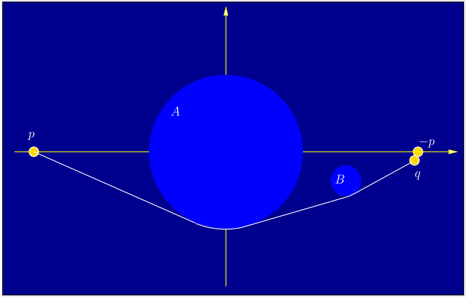
- Shortest path from $p$ to $q$ avoiding discs $A, B$



# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

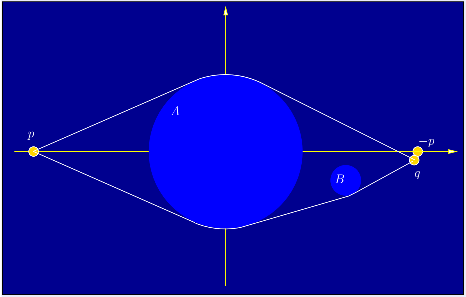
- Shortest path from $p$ to $q$ avoiding discs $A, B$



# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$

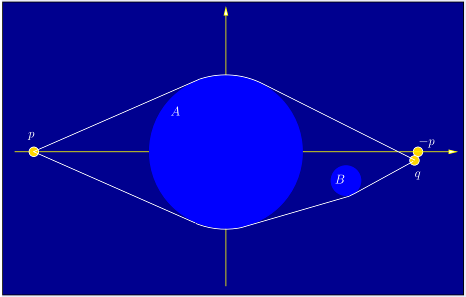



- Segment length is a square-root of an algebraic number

# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$

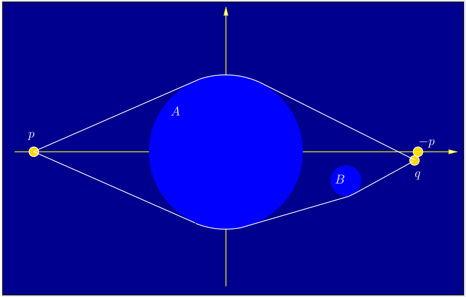



  - Segment length is a square-root of an algebraic number
  - Arc lengh is $r\,\theta$

# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$

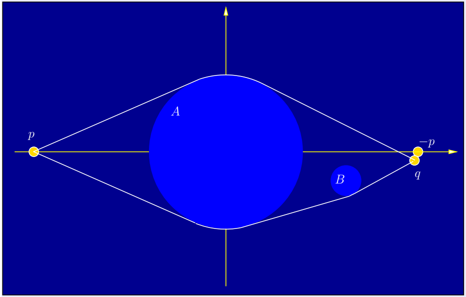



- Segment length is a square-root of an algebraic number
- Arc lengh is $r\theta$

# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$

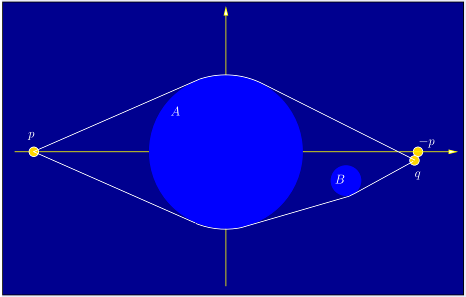



- Segment length is a square-root of an algebraic number
- Arc lengh is $r\theta$

# (III) Shortest Path Amidst Discs

## Shortest Path amidst Discs

- Shortest path from $p$ to $q$ avoiding discs $A, B$




  - Segment length is a square-root of an algebraic number
  - Arc lengh is $r\theta$

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? $\boxed{\text{Numerical Halting Problem}}$
- Analogue of the $\boxed{\text{Turing Halting Problem}}$
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? $\boxed{\text{Numerical Halting Problem}}$
- Analogue of the $\boxed{\text{Turing Halting Problem}}$
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? Numerical Halting Problem
- Analogue of the Turing Halting Problem
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Disc Obstacles, contd.

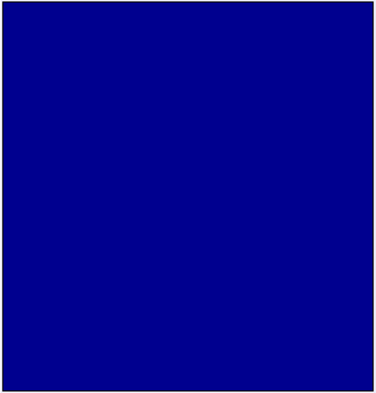
## Reduction to Dijkstra's Algorithm (Again?)

- Combinatorial complexity: $O(n^2 \log n)$ (negligible, exercise)
- Path length $= \gamma + \alpha$
  where $\gamma$ is algebraic, but $\alpha$ is transcendental
- Not even clear that we can compute this!
  - Why? $\boxed{\text{Numerical Halting Problem}}$
- Analogue of the $\boxed{\text{Turing Halting Problem}}$
  - Also semi-decidable
- Reference: my 2006 paper with E.Chang, S.W.Choi, D.Kwon, H.Park.

# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

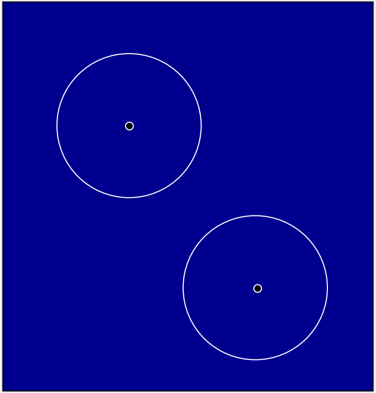
Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

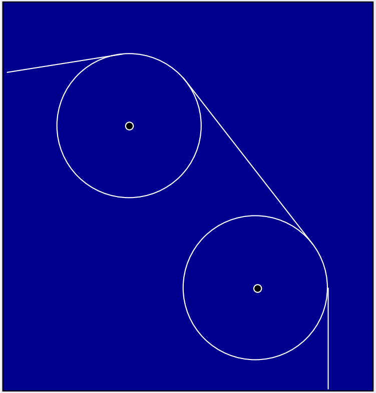
Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

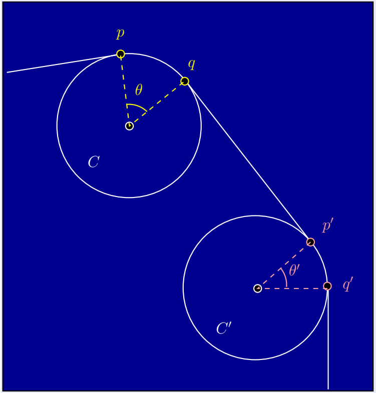
Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

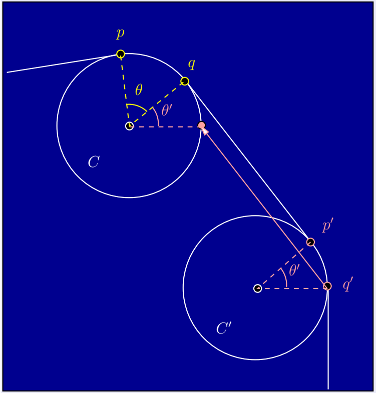
Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

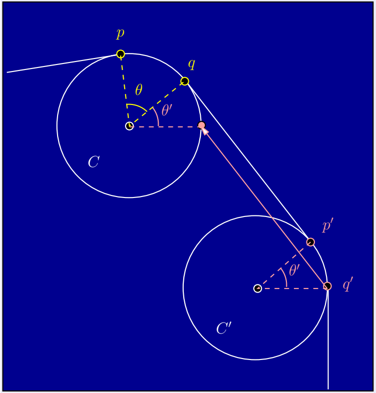
Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.

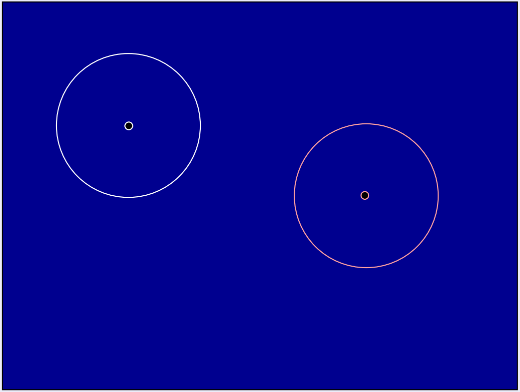


# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.

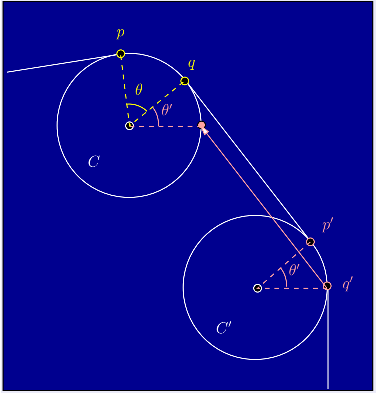


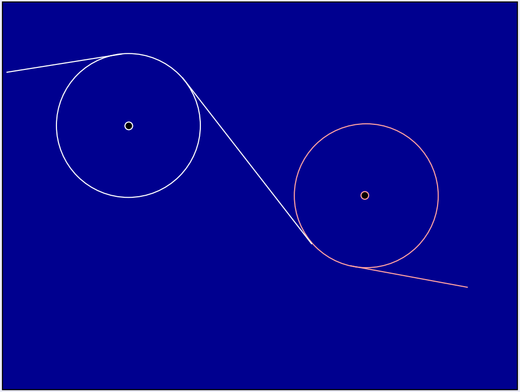



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.

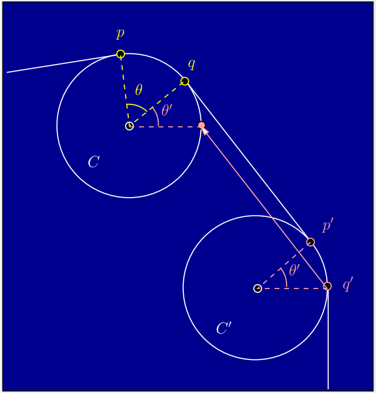


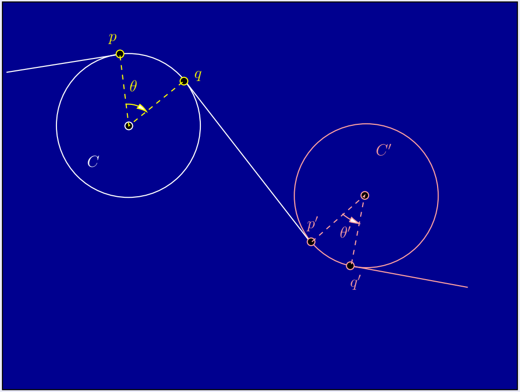



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.

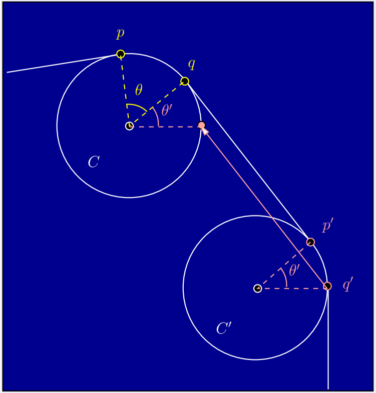


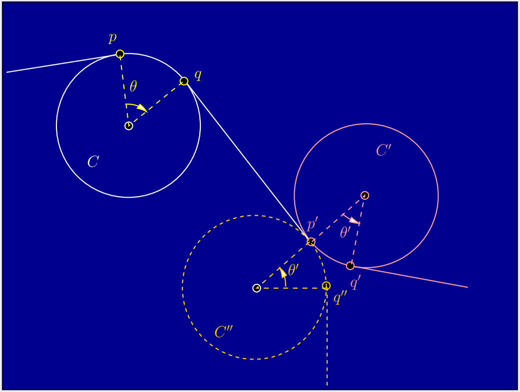



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.

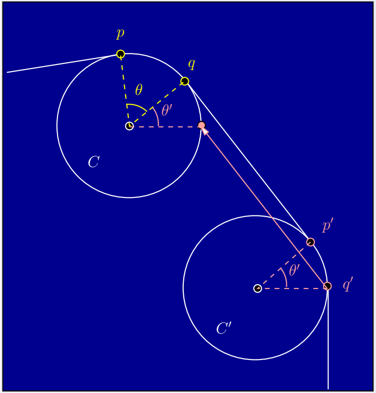


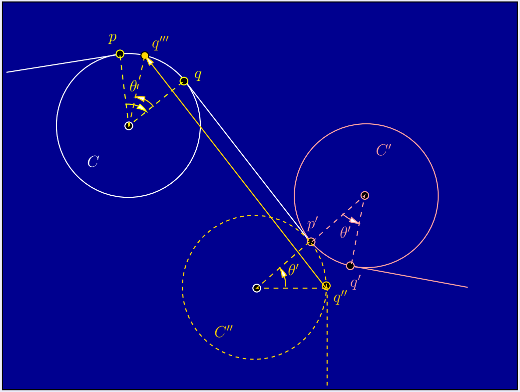



# Addition/Subtraction of Arc Lengths

## Simple Case: Unit Discs

Let $A = [C, p, q, n]$ and $A' = [C', p', q', n']$ encode two arc lengths.



# Decidability [Chang/Choi/Kwon/Park/Y. (2005)]

## Theorem (Unit Disc)

*Shortest Path for unit disc obstacles is computable.*

## Theorem (Commensurable Radii)

*Shortest Path for commensurable radii discs is computable.*

## No complexity Bounds!

Appeal to Baker's Linear Form in Logarithms: $\left|\alpha_0 + \sum_{i=1}^{n} \alpha_i \log \beta_i\right| > B$

## Theorem (Commensurable Radii Complexity)

*Shortest Paths for rational discs is in single-exponential time.*

- Rare positive result from Transcendental Number Theory
- First transcendental geometric problem shown computable

# Decidability [Chang/Choi/Kwon/Park/Y. (2005)]

## Theorem (Unit Disc)

*Shortest Path for unit disc obstacles is computable.*

## Theorem (Commensurable Radii)

*Shortest Path for commensurable radii discs is computable.*

## No complexity Bounds!

Appeal to Baker's Linear Form in Logarithms: $\left|\alpha_0 + \sum_{i=1}^{n} \alpha_i \log \beta_i\right| > B$

## Theorem (Commensurable Radii Complexity)

*Shortest Paths for rational discs is in single-exponential time.*

- Rare positive result from Transcendental Number Theory
- First transcendental geometric problem shown computable

# Decidability [Chang/Choi/Kwon/Park/Y. (2005)]

## Theorem (Unit Disc)

*Shortest Path for unit disc obstacles is computable.*

## Theorem (Commensurable Radii)

*Shortest Path for commensurable radii discs is computable.*

## No complexity Bounds!

Appeal to Baker's Linear Form in Logarithms: $\left|\alpha_0 + \sum_{i=1}^{n} \alpha_i \log \beta_i\right| > B$

## Theorem (Commensurable Radii Complexity)

*Shortest Paths for rational discs is in single-exponential time.*

- Rare positive result from Transcendental Number Theory
- First transcendental geometric problem shown computable

# Decidability [Chang/Choi/Kwon/Park/Y. (2005)]

## Theorem (Unit Disc)

*Shortest Path for unit disc obstacles is computable.*

## Theorem (Commensurable Radii)

*Shortest Path for commensurable radii discs is computable.*

## No complexity Bounds!

Appeal to Baker's Linear Form in Logarithms: $\left|\alpha_0 + \sum_{i=1}^{n} \alpha_i \log \beta_i\right| > B$

## Theorem (Commensurable Radii Complexity)

*Shortest Paths for rational discs is in single-exponential time.*

- Rare positive result from Transcendental Number Theory
- First transcendental geometric problem shown computable

# Decidability [Chang/Choi/Kwon/Park/Y. (2005)]

## Theorem (Unit Disc)

*Shortest Path for unit disc obstacles is computable.*

## Theorem (Commensurable Radii)

*Shortest Path for commensurable radii discs is computable.*

## No complexity Bounds!

Appeal to Baker's Linear Form in Logarithms: $\left|\alpha_0 + \sum_{i=1}^{n} \alpha_i \log \beta_i\right| > B$

## Theorem (Commensurable Radii Complexity)

*Shortest Paths for rational discs is in single-exponential time.*

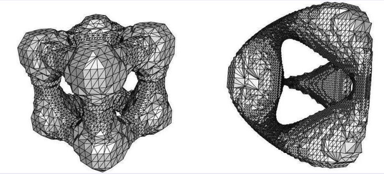
- Rare positive result from Transcendental Number Theory
- First transcendental geometric problem shown computable

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

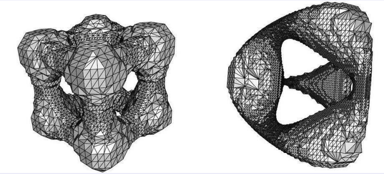
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \longrightarrow \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

“Tangled Cube”



“Chair”

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

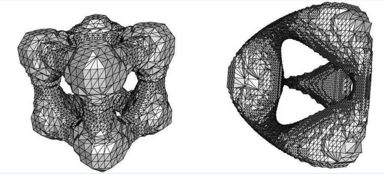
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

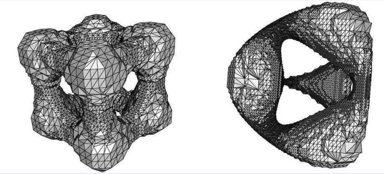
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

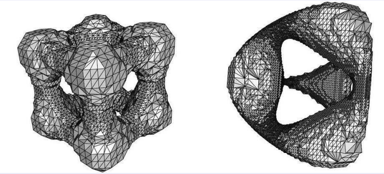
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

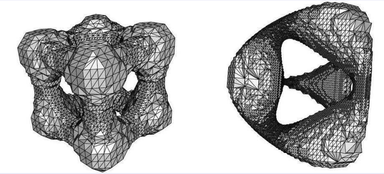
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

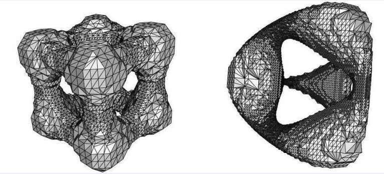
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

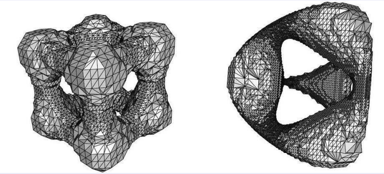
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \rightarrow \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

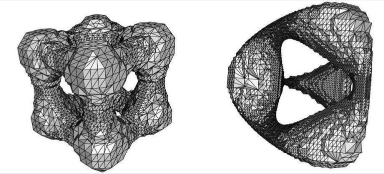
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

## Applications

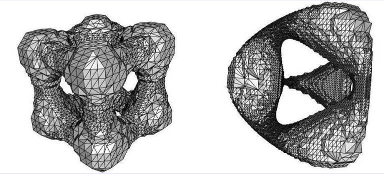
Visualization, Graphics, Simulation, Modeling: prerequisite

# (IV) Mesh Generation

## Meshing of Surfaces

- Surface $S = f^{-1}(0)$ where $f : \mathbb{R}^n \to \mathbb{R}$ ($n = 1, 2, 3$)
- Wants a triangulated surface $\widetilde{S}$ that is isotopic to $S$

"Tangled Cube"



"Chair"

- Case $n = 1$ is root isolation!
- Return to meshing in Lecture 2

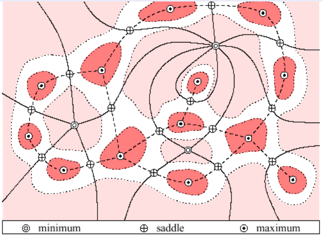
## Applications

Visualization, Graphics, Simulation, Modeling: prerequisite

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

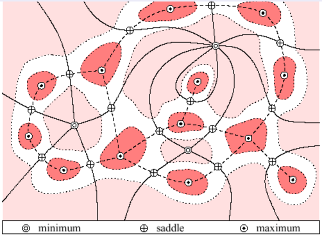



- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

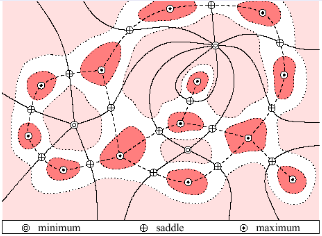



- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

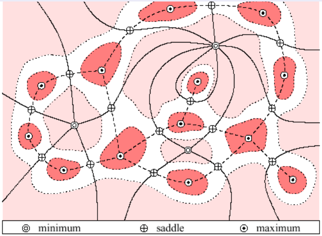



- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

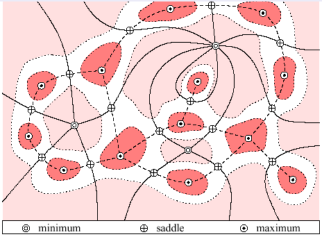



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this "Continuous-to-Discrete" transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

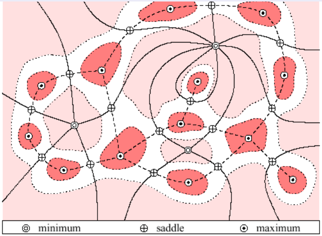



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this "Continuous-to-Discrete" transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

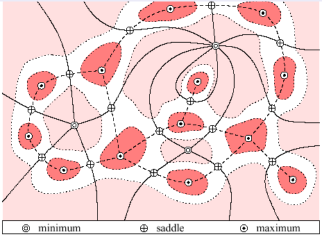



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

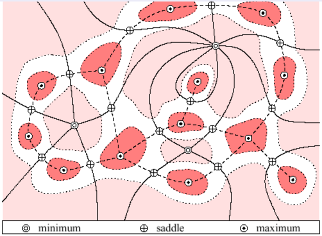



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

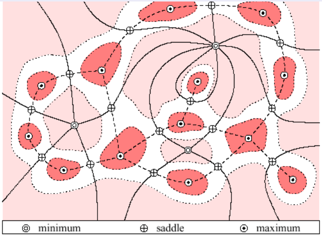



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

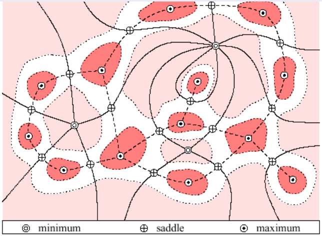



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this "Continuous-to-Discrete" transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

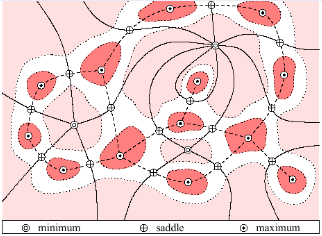



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this "Continuous-to-Discrete" transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:

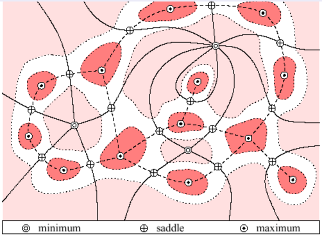



  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this “Continuous-to-Discrete” transformation

# (V) Discrete Morse Theory

## Edelsbrunner, Harer, Zomorodian (2003)

- Methodology: discrete analogues of continuous concepts
  - Differential geometry, Ricci flows, etc
- Morse-Smale Complex of a surface $S = f^{-1}(0)$:




  - Critical Points (max/min/saddle)
  - Integral Lines
  - OPEN: How to connect saddle to its maximas
- Exactness Bottleneck: this "Continuous-to-Discrete" transformation

# Mini Summary

- We saw 5 Geometric Problems:
  I=classic, II=hard, III=very hard, IV=current, V=open
- Up Next : Let us examine their underlying computational models...

# Mini Summary

- We saw 5 Geometric Problems:
  I=classic, II=hard, III=very hard, IV=current, V=open
- Up Next: Let us examine their underlying computational models...

# Mini Summary

- We saw 5 Geometric Problems:
  I=classic, II=hard, III=very hard, IV=current, V=open
- Up Next: Let us examine their underlying computational models...

# Mini Summary

- We saw 5 Geometric Problems:
  I=classic, II=hard, III=very hard, IV=current, V=open
- Up Next : Let us examine their underlying computational models...

# Coming Up Next

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Two Worlds of Computing

- (EX) Discrete, Combinatorial, *Exact*.
  - Theoretical Computer Science, Computer Algebra
- (AP) Continuous, Numerical, *Approximate*.
  - Computational Science & Engineering (CS&E) or Physics
  - Problems too hard in exact framework (e.g., 3D Ising Model)
  - Even when exact solution is possible,...
- The 2 Worlds meet in Geometry
  - Solving Linear Systems (Gaussian vs. Gauss-Seidel)
  - Linear Programming (Simplex vs. Interior-Point)
  - Solving Numerical PDE (Symbolic vs. Numeric)

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Again, What is Geometry?

## Geometry is always about zeros

- Problem (I): Is a Point on a Hyperplane?
- Problems (II),(III): Are two path lengths are equal?
- Problems (IV),(V): Continuous-to-discrete transformations, defined by zero sets
- These zero decisions are captured by geometric predicates
- View developed by CG'ers in robust geometric computation

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x+y)(1+\varepsilon)$)
  - PROBLEM: Zero is abolished

# Four Computational Models for Geometry

## How to compute in a Continuum ($\mathbb{R}^n$)?

- (EX) Algebraic Computational Model
  (e.g., Real RAM, Blum-Shub-Smale model)
  - PROBLEM: Zero is trivial
- (EX') Abstract Operational Models
  (e.g., CG, Traub, Orientation, Ray shooting, Giftwrap)
  - PROBLEM: Zero is hidden
- (AP) Analytic Computational Model (e.g., Ko, Weihrauch)
  - PROBLEM: Zero is undecidable
- (AP') Numerical Analysis Model (e.g., $x \oplus y = (x + y)(1 + \varepsilon)$)
  - PROBLEM: Zero is abolished

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a "discontinuous" phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a "discontinuous" phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Other Issues

## You cannot avoid the Zero Problem

- (EX) How do you implement $\mathbb{R}$?
- (EX') We may abstract away too much
  - cf. Problems (II) and (III)
- (AP) Only continuous functions are computable
  - Geometry is a discontinuous phenomenon
- (AP') Approximate geometry maybe harder than exact geometry
  - Exercise: Program a geometric algorithm w/o equality test

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but $0.0223$ is more useful!

- How to capture this Duality?

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but $0.0223$ is more useful!

  WHY? Want the locus of $\alpha$ in the continuum

  JOKE: a physicist and an engineer were in a hot-air balloon...

- How to capture this Duality?

  For exact computation, need algebraic representation

  For analytic properties, need an approximation process

  What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!

  WHY? Want the locus of $\alpha$ in the continuum

  JOKE: a physicist and an engineer were in a hot-air balloon...

- How to capture this Duality?

  For exact computation, need algebraic representation.

  For analytic properties, need an approximation process

  What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| ▶ $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!

  WHY? Want the locus of $\alpha$ in the continuum

  JOKE: a physicist and an engineer were in a hot-air balloon...

- How to capture this Duality?

  For exact computation, need algebraic representation.

  For analytic properties, need an approximation process

  What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $= \sqrt{15 - \sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15 - \sqrt{224}}$ is exact, but $0.0223$ is more useful!

  WHY? Want the locus of $\alpha$ in the continuum

  JOKE: a physicist and an engineer were in a hot-air balloon...

- How to capture this Duality?

  For exact computation, need algebraic representation.

  For analytic properties, need an approximation process

  What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $= \sqrt{15 - \sqrt{224}}$ | $\approx 0.0223$ |

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $= \sqrt{15 - \sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15 - \sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $= \sqrt{15 - \sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15 - \sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $= \sqrt{15 - \sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15 - \sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Duality in Numbers

- Physics Analogy:

| | Discrete | Continuous |
|---|---|---|
| Light | particle | wave |
| $\mathbb{R}$ | field | metric space |
| Numbers | algebraic | analytic |
| $\alpha$ | $=\sqrt{15-\sqrt{224}}$ | $\approx 0.0223$ |

- $\sqrt{15-\sqrt{224}}$ is exact, but 0.0223 is more useful!
  - WHY? Want the locus of $\alpha$ in the continuum
  - JOKE: a physicist and an engineer were in a hot-air balloon...
- How to capture this Duality?
  - For exact computation, need algebraic representation.
  - For analytic properties, need an approximation process
  - What about deciding zero? (Algebraic or Numeric)

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next : A General Solution

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next : A General Solution

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next : A General Solution

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next: A General Solution

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next: A General Solution

# Mini Summary

- Geometry is decided by Zeros
- Zero is a special number
- Numbers have a dual nature: need dual representation
- Up Next: A General Solution

# Coming Up Next

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# The Universal Solution (EGC)

## Key Principle of Exact Geometric Computation (EGC)

- Algorithm = Sequence of Steps
- Steps = Construction $\boxed{x := y + 2;}$ or Tests $\boxed{\text{if } x = 0 \text{ goto L}}$
- Geometric relations determined by Tests (Zero or Sign)
- THUS: if Tests are error free, the Geometry is exact
- Numerical robustness follows! Take-home message

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double→BigFloat→Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double→BigFloat→Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double→BigFloat→Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double→BigFloat→Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
    - Promotion/Demotion Rules: e.g., double→BigFloat→Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double$\longrightarrow$BigFloat$\longrightarrow$Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double$\longrightarrow$BigFloat$\longrightarrow$Expr

# Implementing the Universal Solution (Core Library)

## Any programmer can access this capability

```
#define Core_Level 3
#include "CORE.h"
.... Standard C++ Program ....
```

## Numerical Accuracy API

- Level 1: Machine Accuracy (int, long, float, double)
- Level 2: Arbitrary Accuracy (BigInt, BigRat, BigFloat)
- Level 3: Guaranteed Accuracy (Expr)
- Program should compile at every Accuracy Level
  - Promotion/Demotion Rules: e.g., double$\longrightarrow$BigFloat$\longrightarrow$Expr

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA , CGAL , Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA , CGAL , Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA, CGAL, Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA, CGAL, Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA, CGAL, Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA , CGAL , Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA, CGAL, Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA , CGAL , Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# What is Achieved?

## Features

- Removed numerical non-robustness from geometry (!)
- Algorithm-independent solution to non-robustness
- Standard (Euclidean) geometry (why important?)
- Exactness in geometry ( can use approximate numbers !)
- Implemented in LEDA , CGAL , Core Library

## Other Implications

- A new approach to do algebraic number computation
- In Euclidean Shortest Path, we need the signs of expressions like $\sum_{i=1}^{100} a_i \sqrt{b_i}$.

  Standard algebraic approach is doomed

# Coming Up Next

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
  - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Adaptive Zero Determination

## Core of Core Library

- MUST not use algebraic method!
- Numerical method based on Zero Bounds
- Let $\Omega = \{+, -, \times, \ldots\} \cup \mathbb{Z}$ be a class of operators
    - $ZERO(\Omega)$ is the corresponding zero problem
- A Zero Bound for $\Omega$ is a function $B : Expr(\Omega) \to \mathbb{R}_{\geq 0}$ such that $e \in Expr(\Omega)$ is non-zero implies

$$|e| > B(e)$$

- How to use zero bounds? Combine with approximation.
- Zero Bound is the bottleneck only in case of zero.

# Some Constructive Bounds

- Degree-Measure Bounds [Mignotte (1982)], [Sekigawa (1997)]
- Degree-Height, Degree-Length [Yap-Dubé (1994)]
- BFMS Bound [Burnikel et al (1989)]
- Eigenvalue Bounds [Scheinerman (2000)]
- Conjugate Bounds [Li-Yap (2001)]
- BFMSS Bound [Burnikel et al (2001)]
  - One of the best bounds
- k-ary Method [Pion-Yap (2002)]
  - Idea: division is bad. $k$-ary numbers are good

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
-

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
-

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
-

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
- 

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
-

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# An Example

- Consider the $e = \sqrt{x} + \sqrt{y} - \sqrt{x + y + 2\sqrt{xy}}$.
- Assume $x = a/b$ and $y = c/d$ where $a, b, c, d$ are $L$-bit integers.
- Then Li-Yap Bound is $28L + 60$ bits, BFMSS is $96L + 30$ and Degree-Measure is $80L + 56$.
-

| $L$ | 50 | 100 | 500 | 5000 |
|---|---|---|---|---|
| BFMS | 0.637 | 9.12 | 101.9 | 202.9 |
| Measure | 0.063 | 0.07 | 1.93 | 15.26 |
| BFMSS | 0.073 | 0.61 | 1.95 | 15.41 |
| Li-Yap | 0.013 | 0.07 | 1.88 | 1.89 |

# Mini Summary

- There is a “Universal Solution” for synthesizing the Algebraic and the Geometric viewpoints
- Slogan: $\boxed{\text{Algebraic computation without Algebra}}$

  (Use approximations & zero bounds)
- PUZZLE 3: What was the answer to PUZZLE 2?

# Mini Summary

- There is a “Universal Solution” for synthesizing the Algebraic and the Geometric viewpoints
- Slogan: $\boxed{\text{Algebraic computation without Algebra}}$
  (Use approximations & zero bounds)
- PUZZLE 3: What was the answer to PUZZLE 2?

# Mini Summary

- There is a “Universal Solution” for synthesizing the Algebraic and the Geometric viewpoints
- Slogan: $\boxed{\text{Algebraic computation without Algebra}}$
  (Use approximations & zero bounds)
- PUZZLE 3: What was the answer to PUZZLE 2?

# Mini Summary

- There is a “Universal Solution” for synthesizing the Algebraic and the Geometric viewpoints
- Slogan: $\boxed{\text{Algebraic computation without Algebra}}$
  (Use approximations & zero bounds)
- PUZZLE 3: What was the answer to PUZZLE 2?

# Mini Summary

- There is a “Universal Solution” for synthesizing the Algebraic and the Geometric viewpoints
- Slogan: $\boxed{\text{Algebraic computation without Algebra}}$
  (Use approximations & zero bounds)
- PUZZLE 3: What was the answer to PUZZLE 2?

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms

# Summary of Lecture 1

- Nature of Geometric Computation:
  - Discrete as well as Continuous
  - Algebraic as well as Analytic
- It is possible to provide a fairly general solution (ENC) that combines the dual nature of numbers
- Up Next : Directly design ENC algorithms