On Soft Foundations for Geometric Computation

Chee Yap

Courant Institute, NYU

26th Fall Workshop on Computational Geometry
Oct 27-28, 2016

Overview

Next...

[Start] [End]

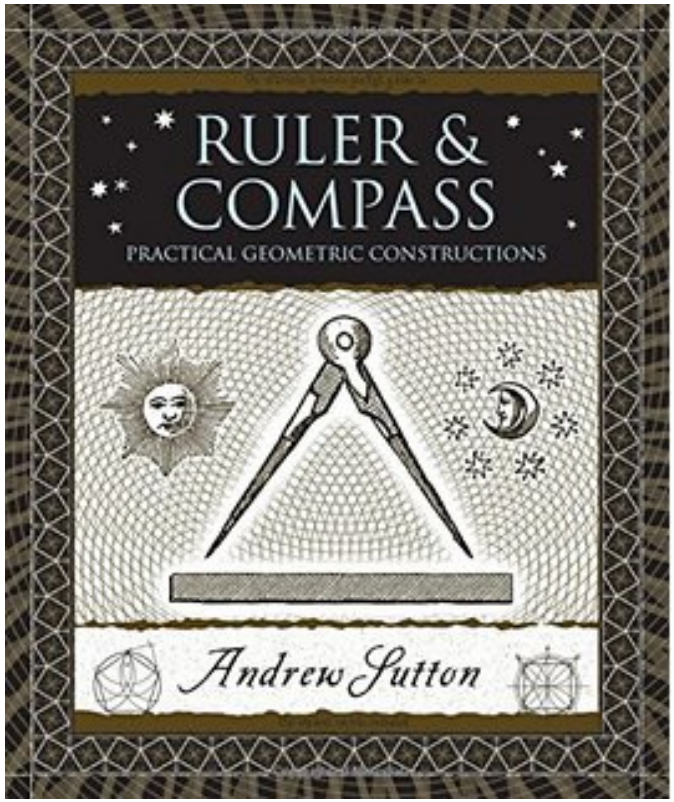
I. Introduction

Trouble with Computational Models

- Ancient Greek Geometry

Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model

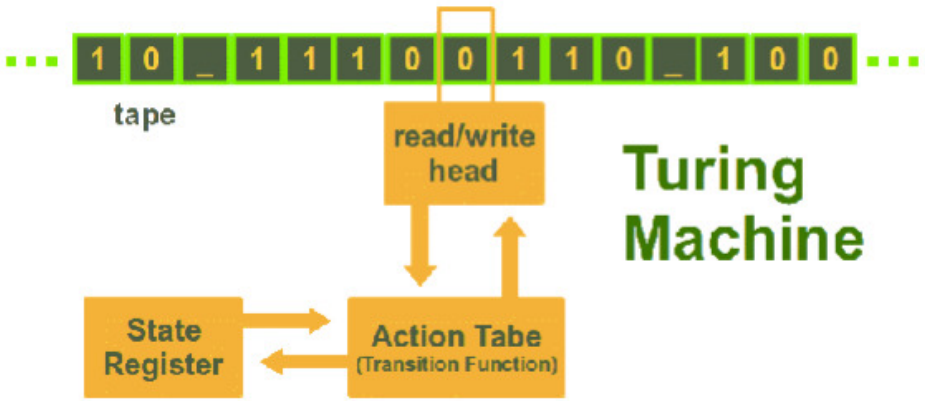


Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model
- Logic of Computing

Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model
- Logic of Computing
  - Turing Machine Model (Church's Thesis)

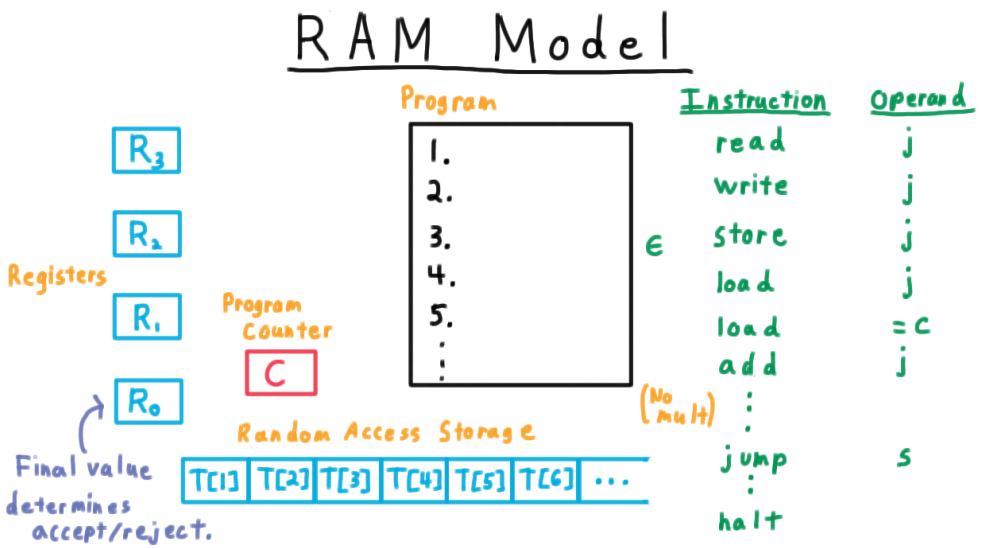


Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model
- Logic of Computing
  - Turing Machine Model (Church's Thesis)
- Geometric Computing

Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model
- Logic of Computing
  - Turing Machine Model (Church's Thesis)
- Geometric Computing
  - Real RAM model



Trouble with Computational Models

- Ancient Greek Geometry
  - Ruler and Compass Model
- Logic of Computing
  - Turing Machine Model (Church's Thesis)
- Geometric Computing
  - Real RAM model (not Church Equivalent!)
- ... the trouble begins

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
  - "Pitfalls of Numerical F.P. Computation"

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
  - "Pitfalls of Numerical F.P. Computation"
- The trouble according to Computational Geometers

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
- The trouble according to Computational Geometers
  - (crash, loop, inconsistency, etc)
  - Geometric/topological roots
  - “Bugbear” [Sedgewick], “Unsolvable problem” [Forrest]

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
- The trouble according to Computational Geometers
- Computational Geometry attacks (1980-2000)

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
- The trouble according to Computational Geometers
- Computational Geometry attacks (1980-2000)
  - Many approaches: E.g., what is a Line?

    pixelSet [graphics], polyLine [Hobby,Yao],

    fatLine [Guibas], consistency [Fortune,Hopcroft],

    bounded equations [Suigahara]

The Numerical Nonrobustness Phenomenon

- The trouble according to Numerical Analysts
- The trouble according to Computational Geometers
- Computational Geometry attacks (1980-2000)
- ... but what about Exact Computation?

Exact Geometric Computation (EGC)

- The EGC prescription

Exact Geometric Computation (EGC)

- The EGC prescription
  - Ensure all branches are error-free $_{Rx}$

    (The "Take Home Message")
  - Be exact, but only where needed!

Exact Geometric Computation (EGC)

- The EGC prescription
  - Ensure all branches are error-free $_{Rx}$

Exact Geometric Computation (EGC)

- The EGC prescription
  - Ensure all branches are error-free $_{Rx}$
- Most general/successful solution
  - So numerical approximations are allowed
  - suffices to have an EGC number type
  - Libraries: Core Library, LEDA, CGAL

Exact Geometric Computation (EGC)

- The EGC prescription
  - Ensure all branches are error-free $_{Rx}$
- Most general/successful solution
- ... therein lies the seed of our next challenge

Barriers to EGC

- EGC algorithms may not be Turing computable

Barriers to EGC

- EGC algorithms may not be Turing computable
  - E.g., transcendental functions ($\log, \sin, \exp, \ldots$)
  - The Zero Problem (Numerical Halting Problem!)

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
  - E.g., Euclidean shortest path is exponential in bit-complexity

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
- EGC requires full degeneracy analysis

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
- EGC requires full degeneracy analysis
  - Vor diagram of polyhedral objects (Open Problem!)

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
- EGC requires full degeneracy analysis
- Exact computation is unnecessary/inappropriate

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
- EGC requires full degeneracy analysis
- Exact computation is unnecessary/inappropriate
  - E.g., robot motion planning, Wireless problems
  - No physical system is accurate to $>$ 8 digits !

Barriers to EGC

- EGC algorithms may not be Turing computable
- EGC may be too inefficient
- EGC requires full degeneracy analysis
- Exact computation is unnecessary/inappropriate
- ...beyond EGC?

Towards Soft Alternative Models

- ...but which?

Towards Soft Alternative Models

- ...but which? – let us look at examples!

Towards Soft Alternative Models

- ...but which? – let us look at examples!
- Subdivision Algorithms!!

  A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

Towards Soft Alternative Models

- ...but which? – let us look at examples!
- Subdivision Algorithms!!

  A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

  A.1 Root isolation and clustering

  – [ISSAC'06,'09,'11,'12,'16, SNC'11, CiE'13]

  A.2 Meshing (Isotopic approximation of surfaces)

  – [ISSAC'08,SoCG'09,'12, SPM'12, ICMS'14]

Towards Soft Alternative Models

- ...but which? – let us look at examples!
- Subdivision Algorithms!!

  A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

  B. Combinatorial Problems – polyhededral set $\Omega \subseteq \mathrm{R}^d$

Towards Soft Alternative Models

- ...but which? – let us look at examples!
- Subdivision Algorithms!!

  A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

  B. Combinatorial Problems – polyhededral set $\Omega \subseteq \mathrm{R}^d$

  B.1 Robot motion planning

  – [SoCG'13, WAFR'14, FAW'15]

  B.2 Voronoi diagrams

  – [ISVD'13, SGP'16]

Towards Soft Alternative Models

- ...but which? – let us look at examples!
- Subdivision Algorithms!!

  A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

  B. Combinatorial Problems – polyhededral set $\Omega \subseteq \mathrm{R}^d$
- WHAT TO LOOK OUT FOR:

  – Avoiding the Zero Problem (What is this?)

  – Replacing Algebraic Techniques by Numerical Ones

  – Soft Concepts (What is this?) (i.e., numerical, but relative to "hard" or Algebraic notions)

Next...

[Start] [End]

II. Roots

"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)

"Eventually, the topic [...of proving non-zeroness...] takes over the whole subject [...of Transcendental Number Theory...]"

— DAVID MASSER (2000)

Introduction

Zero-dimensional geometry:

Fundamental Theorem of Algebra (FTA)

Every complex polynomial of degree n

has exactly n complex roots, counted with multiplicity

Introduction

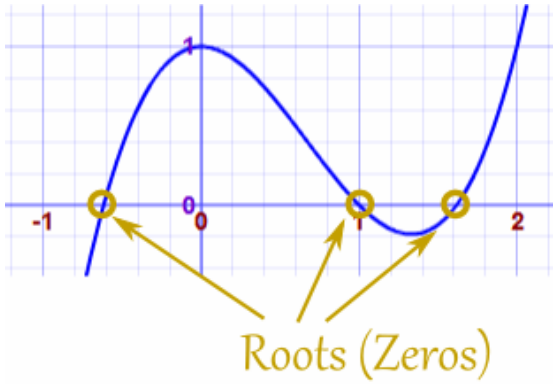
Zero-dimensional geometry:

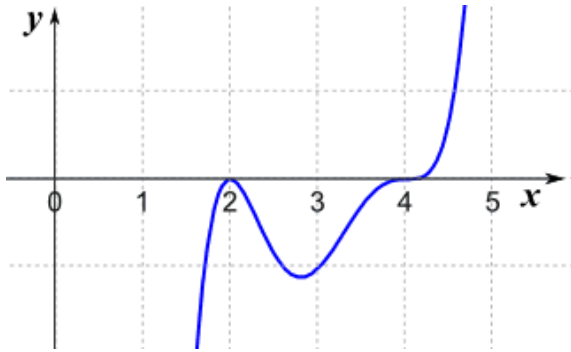
Fundamental Theorem of Algebra (FTA)

Every complex polynomial of degree n

has exactly n complex roots, counted with multiplicity

Real roots:





Introduction

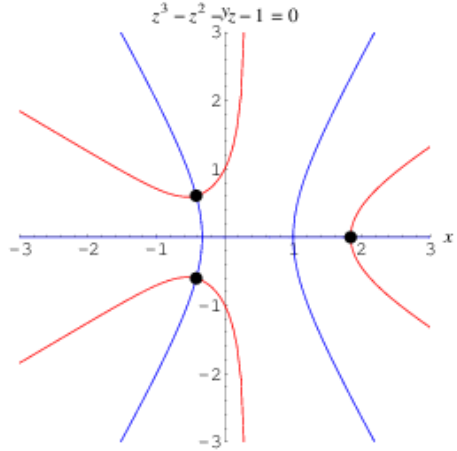
Zero-dimensional geometry:

Fundamental Theorem of Algebra (FTA)

Every complex polynomial of degree n

has exactly n complex roots, counted with multiplicity

Complex roots:



Introduction

- Numerical Analyst's view:
  - compute an ε-disc for each root
- Computer Algebra's view: (exact computation)
  - Root Isolation: first compute disjoint discs
  - Root Refinement: then make discs ε-small

Introduction

- Numerical Analyst's view:
  - compute an ε-disc for each root
- Computer Algebra's view: (exact computation)
  - Root Isolation: first compute disjoint discs
  - Root Refinement: then make discs ε-small


(focus)

Classical History of FTA

Root Finding

has roots (no pun intended) in antiquity (Babylonians)

Associated with most of the “pantheon” of mathematicians

- Descartes, Newton, d’Alembert, Euler, Lagrange,
- Laplace, Gauss, Fourier, Sturm, Weierstrass,
- Vincent, Obreshkorff, Ostrowski, Brouwer, Weyl, Henrici, ...

  – Weierstrass (1891) (basis of Durand-Kerner-Aberth)

  – Weyl (1924) (subdivision approach)

Modern History of FTA

- Modern FTA research can be dated to 1981

Modern History of FTA

- Modern FTA research can be dated to 1981

  * Arnold Schönhage (1982):

    FTA in Terms of Computational Complexity

    – Unpublished

  * Steve Smale (1981):

    FTA and Complexity Theory

    – Bulletin (N.S.) of the AMS.

Modern History of FTA

- Modern FTA research can be dated to 1981
  - Smale and Schönhage
- Each paper initiated a line of research, active to this day!

Modern History of FTA

- Modern FTA research can be dated to 1981
  - Smale and Schönhage
- Each paper initiated a line of research, active to this day!
- Terminology:
  - Benchmark Problem: isolate all roots of integer polynomial
  - Near-Optimal Bound: $\widetilde{O}(n^2 L)$ for Benchmark problem

    (where n=degree, L=coefficient bitsize)
  - Global vs. Local root isolation

Modern History of FTA

- Modern FTA research can be dated to 1981
  - Smale and Schönhage
- Each paper initiated a line of research, active to this day!
- Terminology:
- The Schönhage-Pan Near-Optimal Bound was known for 25 years
  - But never implemented.

  "Our algorithms are quite involved, ... implementation would require a non-trivial work, incorporating numerous known implementation techniques and tricks." – [Pan 2002]

  - Basically, we cannot implement a Real RAM algorithm

    (like many algorithms of CG)

Modern History of FTA

- So, WHAT is implemented in computer algebra systems?
  - Subdivision algorithms! (Sturm, Descartes, Eval, ...)
  - lagged behind the theoretical best bounds
  - recent progress in complexity of subdivision

    (tree-size & bit complexity)

EVAL for Real Roots

- Algorithm Real Roots of $f(x)$ in $I = [a, b]$

```
Start with Q = {I}
While Q ≠ ∅
      Remove J from Q
      If 0 ∉ □f(J) discard J
      Elif 0 ∉ □f'(J) output J
      Elif bisect J and put into Q
```

EVAL for Real Roots

– I call this the EVAL algorithm

– Simple and easy to implement

– Applicable even for analytic f

– Catch: simple roots only (HINT: it is numerical)

EVAL for Real Roots

- How good is it?
  - Depends on how you implement the predicates!

    Exclusion, $C_0(J) : 0 \notin \square f(J)$

    Inclusion, $C_1(J) : 0 \notin \square f'(J)$
  - Remark: numerical heuristics might skip Inclusion test

    (e.g., Yakoubsohn 2005)

EVAL for Real Roots

- How good is it?
  - Depends on how you implement the predicates!

    Exclusion, $C_0(J) : 0 \notin \square f(J)$

    Inclusion, $C_1(J) : 0 \notin \square f'(J)$
  - Remark: numerical heuristics might skip Inclusion test

    (e.g., Yakoubsohn 2005)
- THEOREM [Sharma-Y, ISSAC 2012]

  If $\square f(J)$ *is implemented in "centered form" then*

  $\boxed{\text{EVAL tree size is } O(n(L + r + \log n))}$ *for Benchmark Problem*

EVAL for Real Roots

- THEOREM [Sharma-Y, ISSAC 2012]

  If $\square f(J)$ is implemented in "centered form" then

  EVAL tree size is $O(n(L + r + \log n))$ for Benchmark Problem

- Optimal for $L \geq n$ (matching Sturm!)
  - Else, incomparable with [Burr-Krahmer 2012] or [Sagraloff-Y 2011]
  - Analysis based on Continuous Amortization
- TAKE AWAY: Numerical Methods may be surprisingly good if done right

EVAL for Real Roots

- Bit complexity was tougher story:
  - lagged behind by factor of $\widetilde{O}(n)$
  - breakthrough in Descartes method [Sagraloff 2012]

Near-Optimal Subdivision for Complex Roots

- THEOREM [Ruben-Sagraloff-Sharma-Y, 2015]

  There is a subdivision algorithm with near-optimal bit complexity $\widetilde{O}(n^2(L+n))$ *for Benchmark Problem.*

  – independent of Schönhage's Circle Splitting method

  – local method

  – implementable (explicit error bounds)

Near-Optimal Subdivision for Complex Roots

- SOFT INGREDIENTS:

Near-Optimal Subdivision for Complex Roots

- SOFT INGREDIENTS:
  - Soft comparison $E : F$

    Outcomes: $\boxed{E > F}$ or $\boxed{E < F}$ or $\boxed{\frac{1}{2} < |E/F| < 2}$

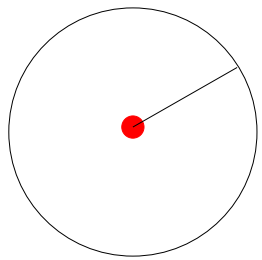
Near-Optimal Subdivision for Complex Roots

- SOFT INGREDIENTS:
  - Soft comparison $E : F$
  - Pellet Test (1881)

Let $\Delta = \mathit{Disc}(r, m)$ and $k = 0, \ldots, n$.

$$T_k(\Delta): \quad \Big|f_k(m)\Big| r^m > \sum_{i \neq k} \Big|f_i(m)\Big| r^i$$

implies $\#(\Delta) = k$.



WHAT conclusion if the test fails?

Near-Optimal Subdivision for Complex Roots

- SOFT INGREDIENTS:
  - Soft comparison $E : F$
  - Soft Converse to Pellet:

    THEOREM [Ruben-Sagraloff-Sharma-Y]

    If $k = \#(\Delta = \#(Kn^4\Delta)$ *then* $T_k(\Delta)$ *holds*

Near-Optimal Subdivision for Complex Roots

- SOFT INGREDIENTS:
  - Soft comparison $E : F$
  - Soft Converse to Pellet:

    THEOREM [Ruben-Sagraloff-Sharma-Y]

    If $k = \#(\Delta = \#(Kn^4\Delta)$ *then* $T_k(\Delta)$ *holds*

    – Tools: Graeffe and Newton iterations

    – Newton-Bisection technique (Sagraloff-Abbot)

    – Planned implementation

Payoffs of Soft Approach

- Adaptive complexity bounds

  based on **local geometric parameters**

  instead of **global synthetic parameters**

Payoffs of Soft Approach

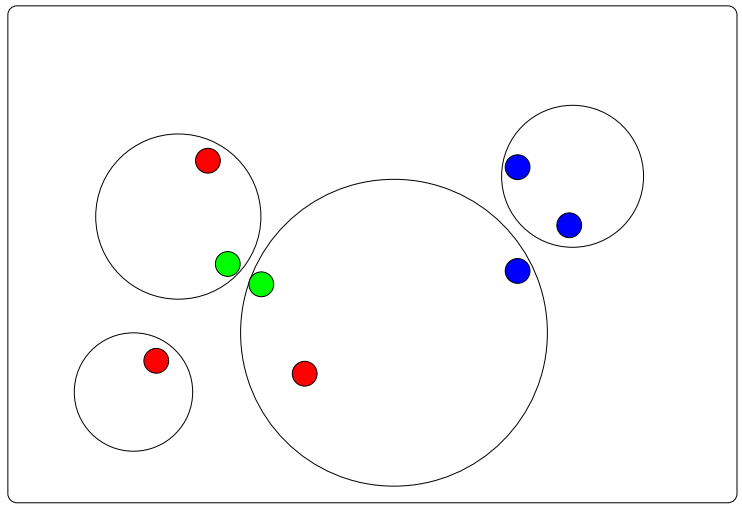
- Beyond the Benchmark Problem
  - Analytic Roots [CiE 2013]
  - Polynomials in $\mathrm{C}[z]$ [ISSAC 2016]

- Key Issue:

  how to avoid the Zero Problem?

Payoffs of Soft Approach

- The ε-Root Clustering Problem



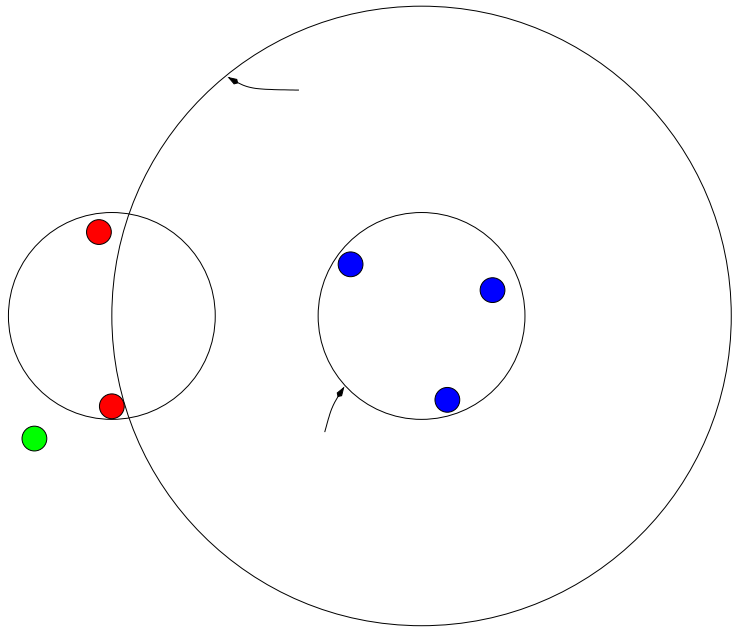
Must output multiplicity per disc

This avoids the Zero problem,

but arbitrariness of clusters?

Payoffs of Soft Approach

- On Natural Clusters



- Natural clusters are disjoint or has inclusion relation
  - at most $2n$ natural clusters, forming a tree!
  - NOW: we can address analytic roots, bit-stream polynomials, etc

Next...

I. Introduction

II. Roots

III. Motion planning

IV. Conclusion

[Start] [End]

III. Motion planning

Motion Planning

- Youtube video from SoCG'2016.

Motion Planning

- Youtube video from SoCG'2016.

Modern Robots

Sci-Fi robot (Karel Capek, 1921)

Industrial robot (1980s)

Domestic robot (2000s):

Fun robots (Drones – 2010s):

Basic Motion Planning Problem:

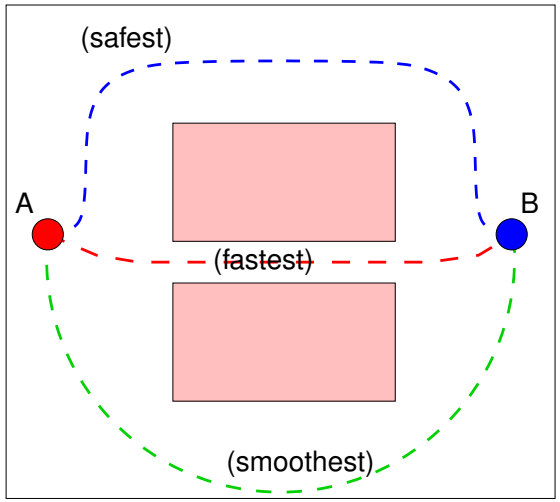
How do you move from A to B, avoiding obstacles?

Assuming you have a map and know your location

Basic Motion Planning Problem:

How do you move from A to B, avoiding obstacles?

Assuming you have a map and know your location



Robot R_0 is fixed; input is (A, B, map)

Kinematics only! (a.k.a. “path planning”)

What we leave out:

Dynamics, non-holonomic constraints, optimality, SLAM

Why? Partly because of lack of softness

3 Approaches to Path Planning

Basic Problem in Robotics for over 40 years

We mention 3 basic approaches:

(I) Subdivision Approach

"A subdivision algorithm in configuration space for findpath with rotation"

— Brooks and Lozano-Perez (8th IJCAI, 1983)

(I) Subdivision Approach

(II) Exact Approach

"On the piano mover's problem: II.
General techniques for computing topological
properties of real algebraic manifolds"

— Schwartz and Sharir (Advances Appl.Math., 1983)

(I) Subdivision Approach

(II) Exact Approach

(III) Sampling Approach (PRM)

"Probabilistic roadmaps for path planning in high-dimensional configuration spaces"

— Kavraki, Svestka, Latombe, and Overmars

(IEEE Trans. Robotics and Automation, 1996)

State-of-Art today (challenges):

(I) Subdivision Approach

- still popular
- Cannot scale beyond medium DOF's (4 or 5 DOF's)

(II) Exact Approach

- connectivity of semi-algebraic sets
- Not practical beyond 3DOF

(III) Sampling Approach (PRM)

- dominated the field in the last 2 decades
- Very weak guarantees
- Grapples with the narrow passage problem

Our goal:

reconsider the foundations for subdivision method

Subdivision Approach

- What do we subdivide?

  Configuration space $\mathcal{C}space = \mathcal{C}space(R_0)$

  E.g.,

  $\mathcal{C}space = \mathrm{R}^2$ (Disc)

  $\mathcal{C}space = SE(3) = \mathrm{R}^3 \times SO(2)$ (Rigid planar robot)

  $\mathcal{C}space = \mathrm{R}^3 \times SO(3)$ (Rigid spatial robot)

  $\mathcal{C}space = \mathrm{R}^2 \times \mathcal{T}^2$ (2-link robot)

- Classify every box $B \subseteq \mathcal{C}space$ as

  FREE (GREEN),

  STUCK (RED),

  MIXED (YELLOW),

  ε-small (GREY).

What is B when $\mathcal{C}space$ is non-Euclidean?

- Classify every box $B \subseteq \mathcal{C}space$ as

  FREE (GREEN),

  STUCK (RED),

  MIXED (YELLOW),

  ε-small (GREY).

  What is B when $\mathcal{C}space$ is non-Euclidean?

- Youtube video from SoCG'2016.

3 Notions of Correctness:

- Exact Approach:

(Path) If there is a path, must output one

(NoPath) If there is no path, must output `NO-PATH`

3 Notions of Correctness:

- Exact Approach:
- Subdivision Approach: "Resolution Complete"

(εPath) If there is a path,

must return one if resolution ε is fine enough

3 Notions of Correctness:

- Exact Approach:
- Subdivision Approach:
- Sampling Approach: "Probabilistic Complete"

(PPath) If there is a path,

must "probably" return one if enough samples N are taken

What is wrong with Exactness?

- Exactness does not make sense for robotics
  - Sensors, environments, actuators, robot dimensions are all approximate
  - Physical constants are known to 8 digits
- Zero problem, efficiency, degeneracy
- Non-adaptive

What is wrong with Resolution/Probabilistic Completeness?

- They are silent about the “NoPath” case!
- This leads to a Halting Problem

  disguised as the Narrow Passage Problem

  (the central problem of Sampling Approach for 20 years)

- Given a resolution parameter, $\varepsilon > 0$.

  How to use it?

- Conventional answer:

  (εPath) If there is a path of clearance ε,

  must return one

  (εNoPath) If there is no path of clearance ε,

  must return `NO-PATH`

No improvement over exact approach!

"exact planning with ε-fattened robot or obstacles"

"No soft enough"

Our solution [Wang-Chang-Y, SoCG'2013]:

a planner is resolution-exact

if it has a constant $K > 1$ such that

(εPath) If there is a path of clearance $K\varepsilon$,

will return one

(εNoPath) If there is no path of clearance ε/K,

will return `NO-PATH`

What does the output tell us?

Contra-positive view:

- If planner returns `NO-PATH`,

  then the optimal clearance is $< K\varepsilon$

- If planner returns a path,

  then the optimal clearance is $> \varepsilon/K$

Indeterminacy (!!) when optimal clearance lies in gap $(\varepsilon/K, K\varepsilon)$

Big Payoff: we escaped the Zero Problem

Subdivision Search

How to exploit resolution-exactness?

Subdivision (of course)!

Soft Subdivision Search (SSS):

```
While Q is non-empty:
   if (Find(Box(α)) = Find(Box(β)) return PATH.
   B ← Q.getNext()
   Expand(B) into children
   Classify new boxes as FREE, STUCK, MIXED.
        FREE-boxes are put into a union-find
           structure, and unioned with neighbors
        MIXED-boxes are put in Q
        ε-small boxes discarded
Return NO-PATH
```

Plug-and-play Framework:

3 data structures

- Subdivision tree
- Priority Queue Q
- Union-Find Data Structure D

3 subroutines

- Q.`getNext()`
- `Expand`(B)
- *Classify*(B)

Subdivision Search + Soft Predicates

The classification $C(B)$ could be done exactly

In fact, all authors seems to assume this!

This does not exploit the full power of subdivision

The new ingredient is... softness

Exact classification: $C(B) \in \{\texttt{FREE}, \texttt{STUCK}, \texttt{MIXED}\}$:

Soft Classification: $\widetilde{C}(B) \in \{\texttt{FREE}, \texttt{STUCK}, \texttt{MIXED}\}$:

(1) **Conservative**

$\widetilde{C}(B) \neq \texttt{MIXED}$ implies $\widetilde{C}(B) = C(B)$

(2) **Convergent**

$p = \lim_{i \to \infty} B_i$ implies $C(p) = \lim_{i \to \infty} \widetilde{C}(B)$

Call $\widetilde{C}$ a **soft version** of C.

Compared to “PRM Framework”:

$\widetilde{C}(B) \equiv$ Collision Detector

Adjacency $\equiv$ Connect(u,v)

- Power of SSS (shared with PRM, not Exact Approach):
  - Flexible *(plug-n-play subroutines)*
  - Easy to extend/generalize *(fattened robot/obstacles)*
  - Adaptive complexity *(cf. exact methods)*
  - One basic algorithmic framework

    (cf. Machine Learning "field with ONE algorithm")

- Design of Soft Predicates

  Can be implemented using only numerical approximations

  Our technique is based on *feature sets*

  LaValle calls it

  > *“opening up the blackbox of collision detection”*

Softness

- Search Strategy
  - Has no impact on correctness for us
  - In PRM, all the effort is done here
- Split Strategy
  - T/R Splitting
  - Critical for the success of our 2-Link Robot

Subdivision in SO(3)

ISSUE : How to do subdivision in non-Euclidean Space?

IDEA : borrow idea of charts from differential geometry

Consider $SO(3)$ or SO_3

3×3 real orthogonal matrices with determinant 1

Can view SO_3 as unit quaternions

$$q = (a, b, c, d) = a + \mathbf{i}b + \mathbf{j}c + \mathbf{k}d$$

where

$$\mathbf{i}^2 = \mathbf{j}^2 = \mathbf{k}^2 = \mathbf{ijk} = -1$$

Unit means $N(q) := a^2 + b^2 + c^2 + d^1 = 1$

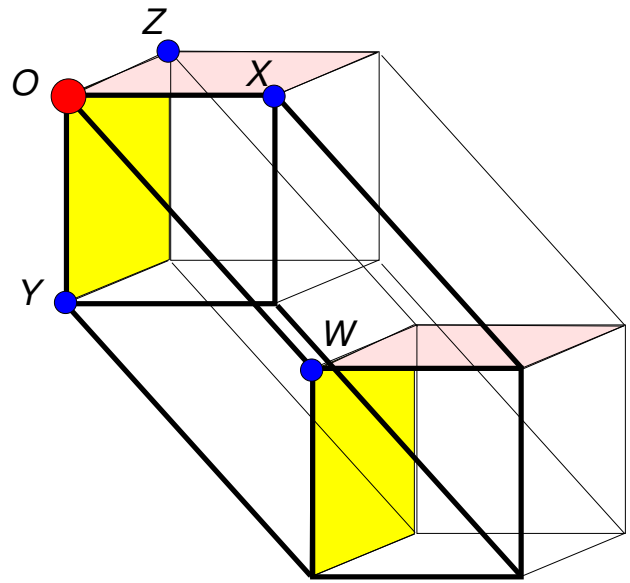
The 3-Sphere $S^3 \subseteq \mathrm{R}^4$

Thus $S^3/\sim$ is a model of SO_3

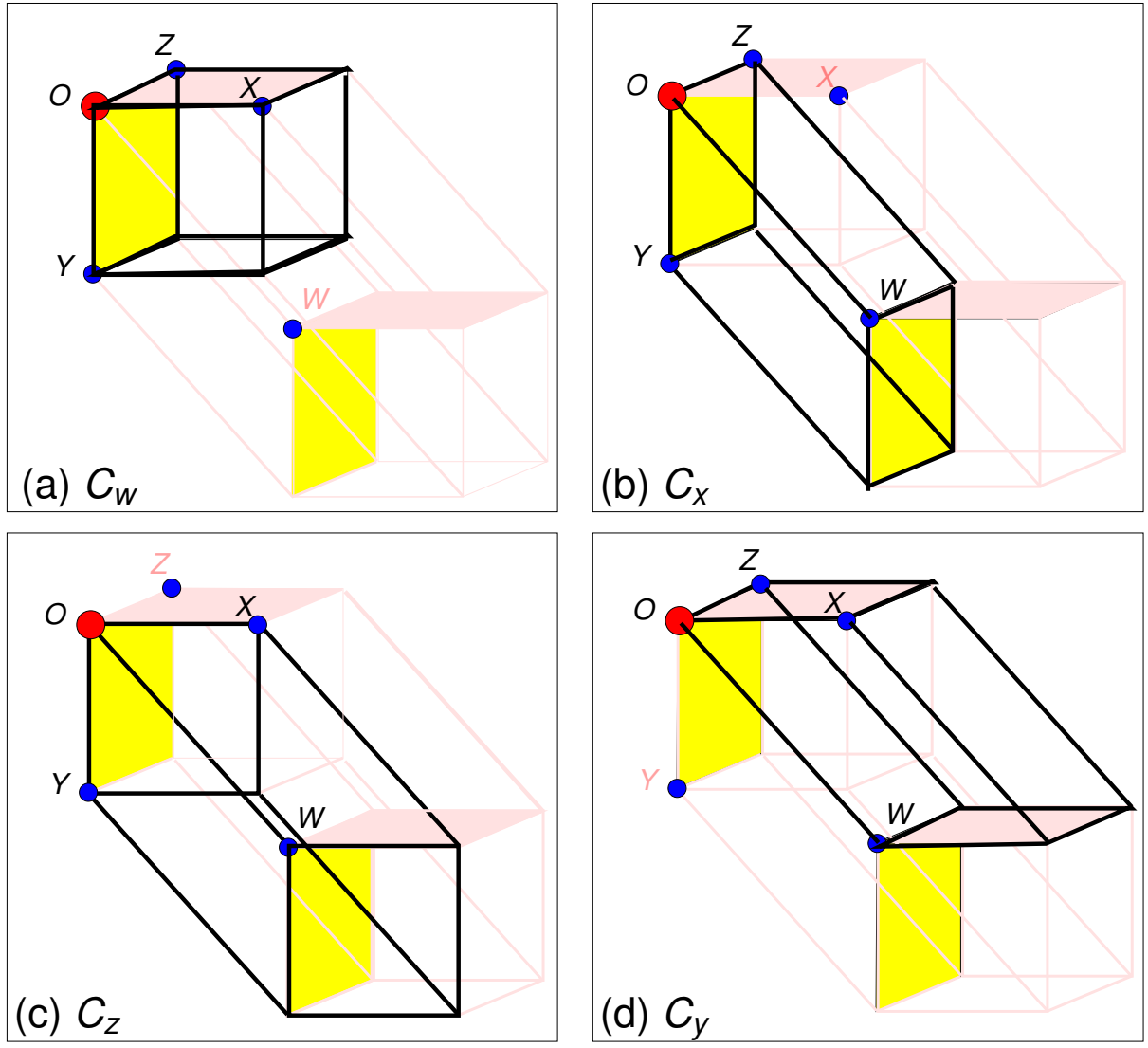
where $q \sim -q$

Consider the 4-cube $I^4 \subseteq \mathrm{R}^4$ where

where $I = [-1, 1]$



Subdivision in SO(3)



Represent SO_3 as the union of four 3-cubes:

C_w = cube defined by corners $O, X, Y, Z, -$

C_x = cube defined by corners $O, -, Y, Z, W$

C_y = cube defined by corners $O, X, -, Z, W$

C_z = cube defined by corners $O, X, Y, -, W$

Let $H^3 := C_w \cup C_x \cup C_y \cup C_z$ be the cubic hemisphere representation of SO_3, with suitable identifications

Use language of chart/atlas of differential geometry:

Each $C_i \subseteq H^3$ has a **chart**

$$\mu_i : C_i \to SO(3)$$

where $\mu_i(q) = q/\|q\|$.

The set $\{\mu_i : i = w, x, y, z\}$ is a **subdivision atlas** of $SO(3)$

The transition map between these charts are trivial.

Get global homeomorphism $\mu : H^3 \to SO(3)$ after identifications.

UPSHOT: can now do subdivision on the domain of each chart μ_i.

What Makes SSS Work? SSS Theory

The trick in axiomatization is not be too general

(to avoid general nonsense),

but enough to capture useful applications with interesting properties.

The Abstract View:

Replace $\mathcal{C}space$ by a metric space (X, d_X)

Replace $\mathcal{C}free$ by an open subset $Y \subseteq X$.

We write $X = X_T \times X_R$

where X_T is translational, Euclidean

X_R is rotational, non-Euclidean, compact

Obstacles live in **physical space** R^k ($k = 2, 3$)

ISSUE: relate $d_X(\cdot, \cdot)$ to the separation in R^k

Footprint function : $Fprint : \mathcal{C}space \rightarrow 2^{\mathrm{R}^k}$

where $Fprint(\gamma)$ = physical space occupied by the robot

Fix the obstacle set $\Omega \subseteq \mathrm{R}^k$.

Clearance function : $C\ell : \mathcal{C}space \rightarrow \mathrm{R}_{\geq 0}$

is defined as

$$\mathrm{Sep}(Fprint(\gamma), \Omega).$$

- Subdivision in X:

  test cells : full-dimensional compact convex polytopes in R^d

- Let $\square X$ be a set of test cells,

  and $\texttt{Expand}(B) : \square X \to 2^{\square X}$ return a (nondeterministic) subdivision of B.

E.g., $\square X$ is the set of simplices,

and $\texttt{Expand}(B)$ returns a bisection at some edge of B.

- Cover X_R by a **subdivision atlas**, $\{\mu_1, \ldots, \mu_t\}$

  $\mu_i : B_i \to X_i$ (homeomorphism)

  $X_1, \ldots, X_t$ is a subdivision of X_R

- Atlas is **good** if it has a **chart constant** $C_0 > 0$

  if $(\forall q, q' \in B_i)$

  $$1/C_0 \leq \frac{d_X(\mu(q),\mu(q'))}{\|q-q'\|} \leq C_0$$

- So far: no discussion of rate of convergence of $\widetilde{C}(B)$

  We need it for "resolution exactness"

- Define: $\widetilde{C}(B)$ is σ- effective ($\sigma > 1$) if

  $C(B) = \texttt{FREE}$ implies $\widetilde{C}(B/\sigma) = \texttt{FREE}$.

THEOREM: *SSS is resolution-exact under these five axioms*:

- (A0: softness) $\widetilde{C}$ is a σ-effective soft version of C
- (A1: expansion) There is $D_0 > 2$ such that `Expand` returns a subdivision of size $\leq D_0$, each cell with at most D_0 vertices and ratio $\ell(B)/w(B) \leq D_0$.
- (A2: Lipschitz) There is $L_0 > 0$ such that for all $\gamma, \gamma' \in Y$, $|C\ell(\gamma) - C\ell(\gamma')| < L_0 d_X(\gamma, \gamma')$.
- (A3: Good Atlas) The subdivision atlas has an atlas constant $C_0 \geq 1$.
- (A4: Translational Cell) There is a constant $K_0 > 0$ such that if $B \in \square X$ is free, then the inner center $c_0 = c_0(B)$ has clearance $C\ell(c_0) \geq K_0 r_0(B)$.

Remarks:

Constants in these axioms:

$$\sigma, D_0, C_0, L_0, K_0.$$

The resolution-exact constant is

$$K := C_0 D_0 \varepsilon (1 + \sigma) L_0.$$

Proof is quite involved because of axiom (A4).

Next...

I. Introduction

II. Roots

III. Motion planning

IV. Conclusion

[Start] [End]

IV. Conclusion

Conclusion

- Recap:
  - we wanted an alternative to Real RAM
  - plan was to look at concrete examples

    i.e., Root Isolation and Motion Planning

    “Saw that soft solutions are superior to hard ones”

Conclusion

- Recap:
- Recall:

A. Algebraic Problems – $F(\mathbf{x}) \in \mathrm{R}[\mathbf{x}]$ or $\mathrm{C}[\mathbf{x}]$

A.1 Root isolation and clustering

– [ISSAC'06,'09,'11,'12,'16, SNC'11, CiE'13]

A.2 Meshing (Isotopic approximation of surfaces)

– [ISSAC'08,SoCG'09,'12, SPM'12, ICMS'14]

Conclusion

- Recap:
- Recall:

  A. Algebraic Problems

  B. Combinatorial Problems – polyhededral set $\Omega \subseteq \mathrm{R}^d$

  B.1 Robot motion planning

  – [SoCG'13, WAFR'14, FAW'15]

  B.2 Voronoi diagrams

  – [ISVD'13, SGP'16]

Conclusion

- Recap:
- Recall:

  A. Algebraic Problems

  B. Combinatorial Problems
- Upshot:

  – main algorithmic paradigm is subdivision/iteration

  – what emerges is a numerical computation model

Quote from Nick Trefethan in ICMS 2014...

Conclusion

- Recap:
- Recall:
- Upshot:
- Outlook:
  - Scope of computational geometry vastly broadened
  - Some unsolvable problems is now in play
  - Produces implementable and practical algorithms
  - New algorithms for old CG problems
  - Numerical CG is wide open (esp. complexity analysis)

Thanks for Listening!

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

— JEAN LE ROND D'ALEMBERT (1717-83)

"To Generalize is to be an Idiot. To Particularize is the Alone Distinction of Merit – General Knowledges are those Knowledges that Idiots possess."

— William Blake (1757 – 1827)

Annotations to Sir Joshua Reynolds's Discourses, pp. xvii – xcviii