Personal Research Areas

Ian Mitchell
Department of Computer Science
The University of British Columbia

research supported by
National Science and Engineering Research Council of Canada
ONR Computational Methods for Collaborative Control MURI (N00014-02-1-0729)

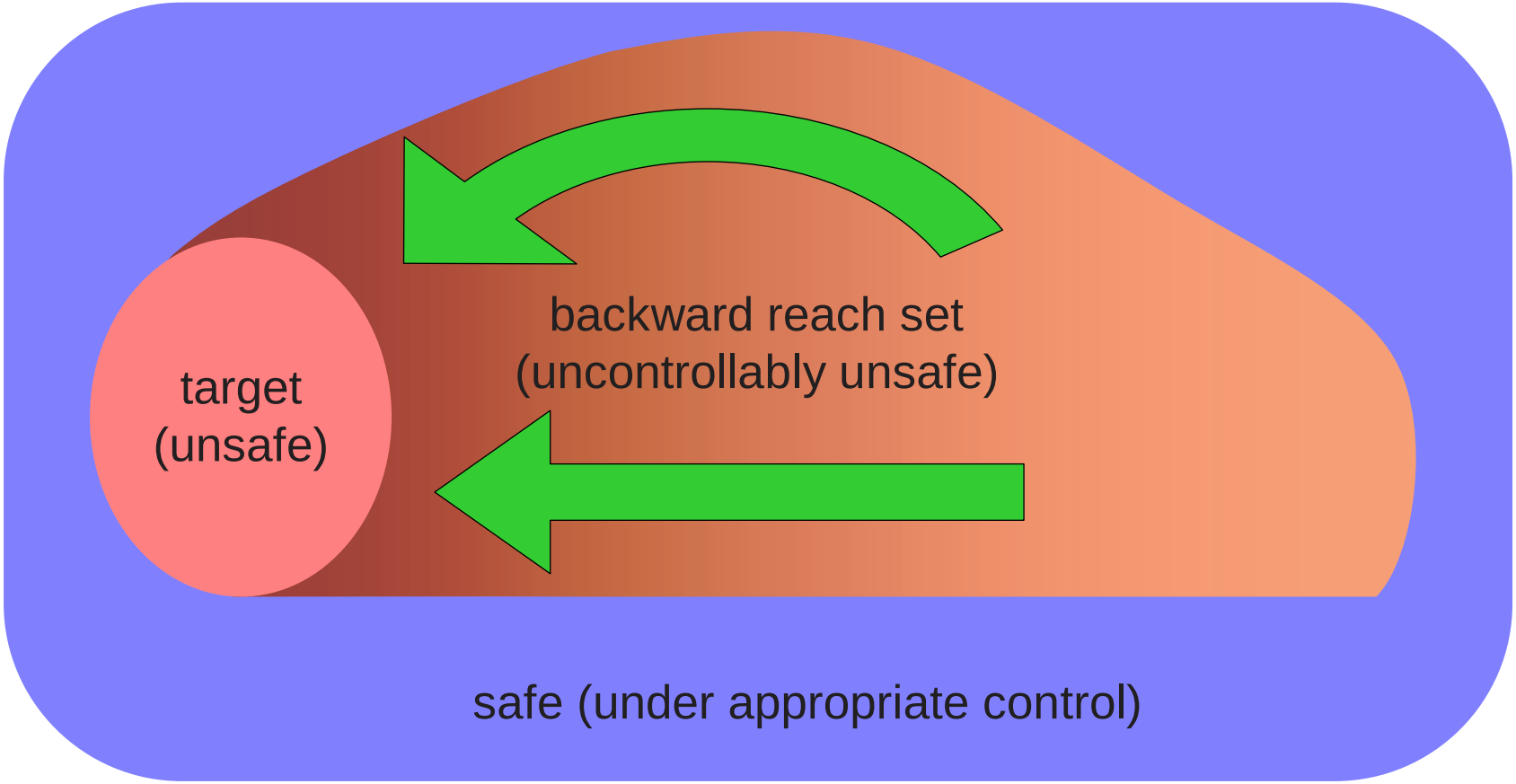
Outline

- Reach sets for control verification
  - Filtering pilot commands for safety
- Dynamic surfaces
  - Particle level set
  - Free surface fluid simulation
- Robotic path planning
  - Shortest continuous path, subject to constraints
  - Optimality for multiple vehicle and/or various norms
- Hamilton-Jacobi PDEs
  - The common link

- Just for fun: **none** of this material is from the course
  - It is graduate level and/or research material

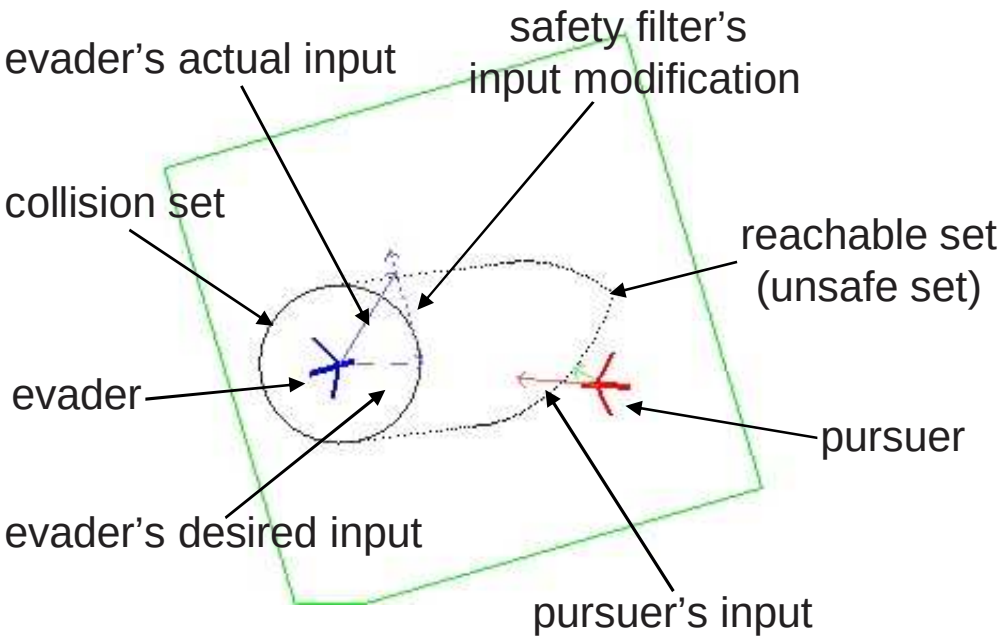
Reachable Sets: What and Why?

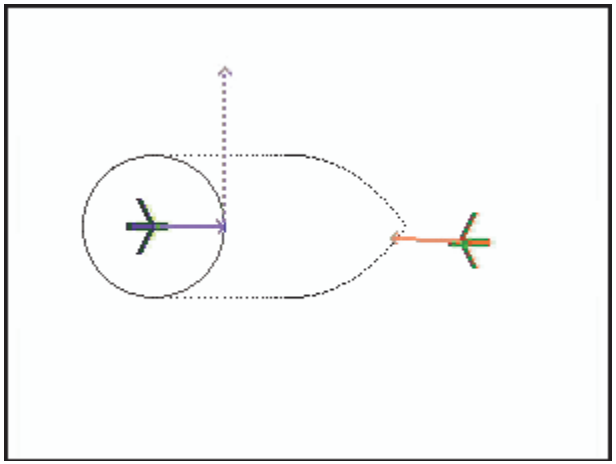
- One application: safety analysis
  - What states are doomed to become unsafe?
  - What states are safe given an appropriate control strategy?



Application: Softwalls for Aircraft Safety

- Use reachable sets to guarantee safety
- Basic Rules
  - Pursuer: turn to head toward evader
  - Evader: turn to head east
- Evader's input is filtered to guarantee that pursuer does not enter the reachable set

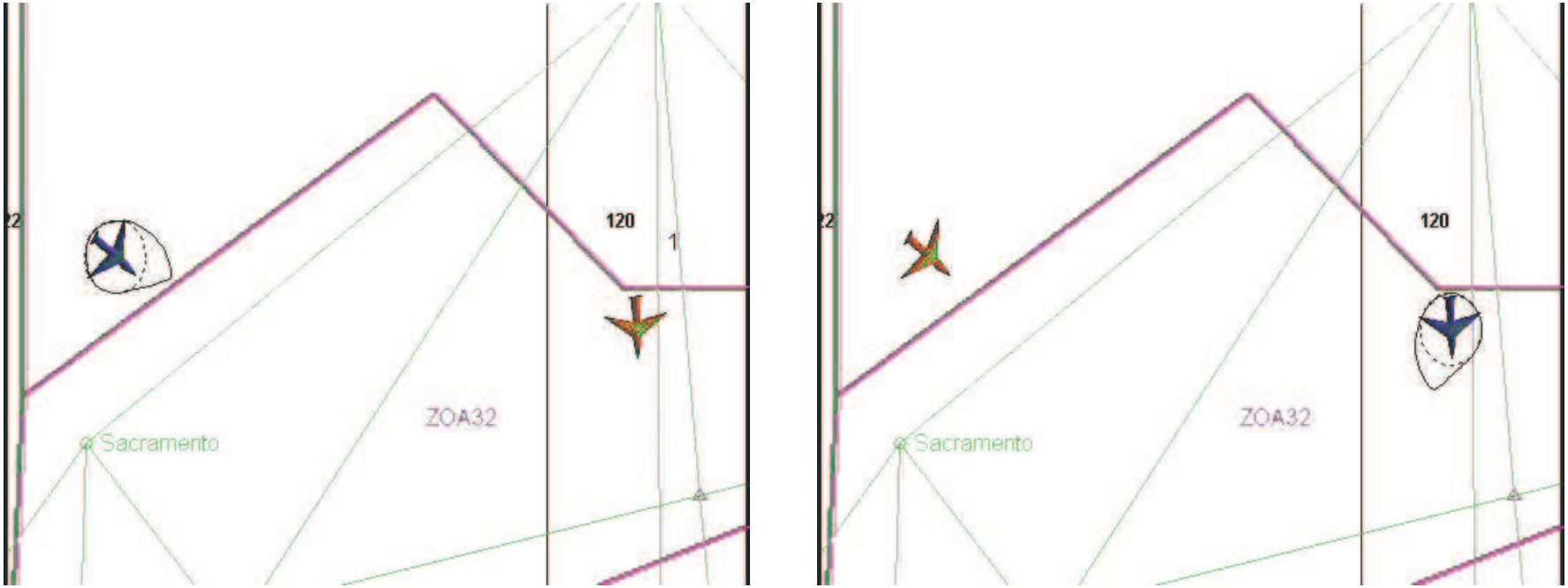




joint work with Edward Lee & Adam Cataldo

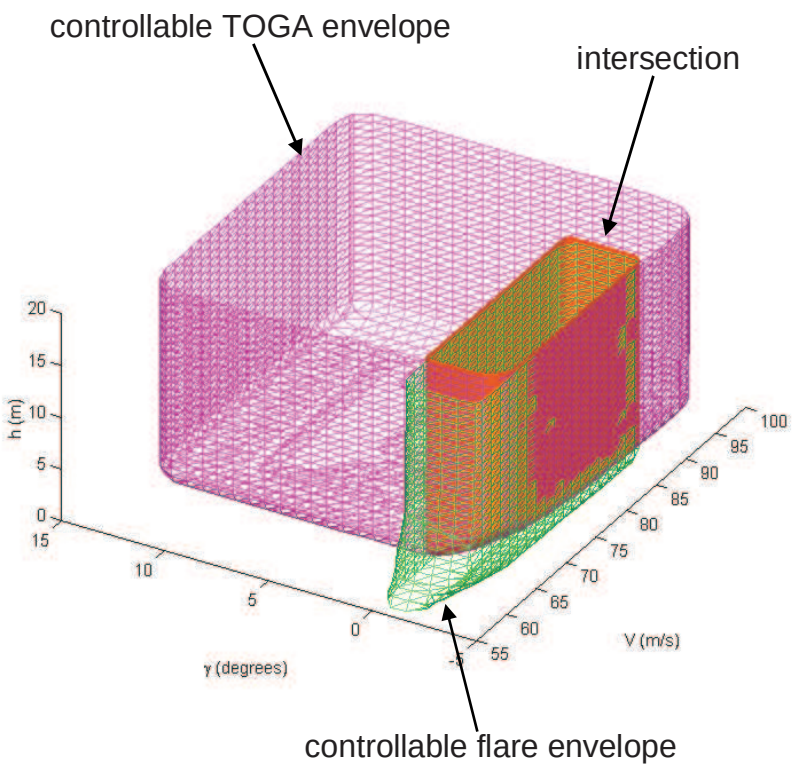
Application: Collision Alert for ATC

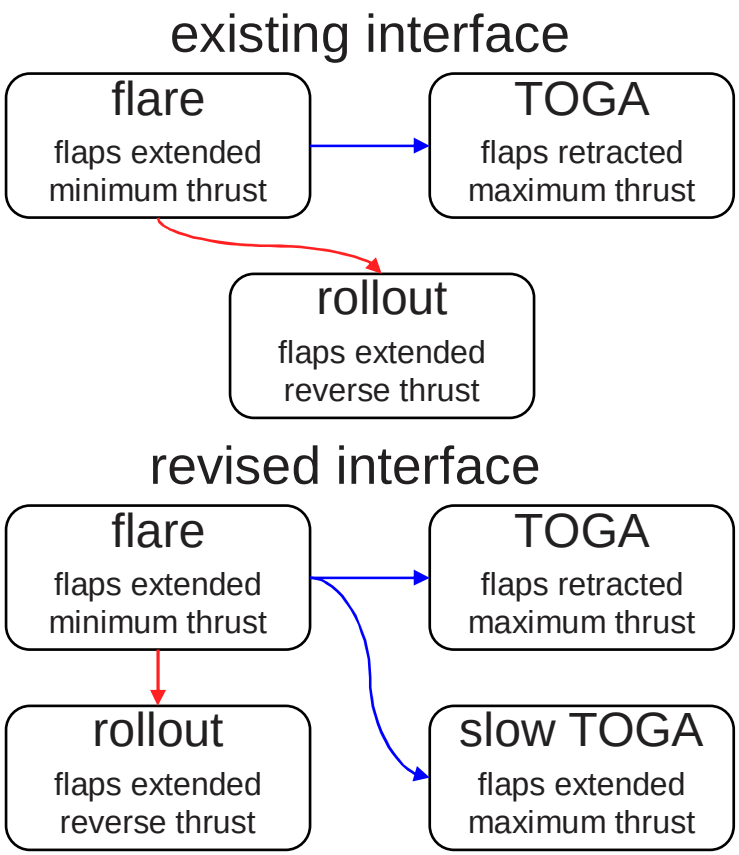
- Use reachable set to detect potential collisions and warn Air Traffic Control (ATC)
  - Find aircraft pairs in ETMS database whose flight plans intersect
  - Check whether either aircraft is in the other's collision region
  - If so, examine ETMS data to see if aircraft path is deviated
  - One hour sample in Oakland center's airspace—
    - 1590 pairs, 1555 no conflict, 25 detected conflicts, 2 false alerts



Application: Cockpit Display Analysis

- Controllable flight envelopes for landing and Take Off / Go Around (TOGA) maneuvers may not be the same
- Pilot's cockpit display may not contain sufficient information to distinguish whether TOGA can be initiated

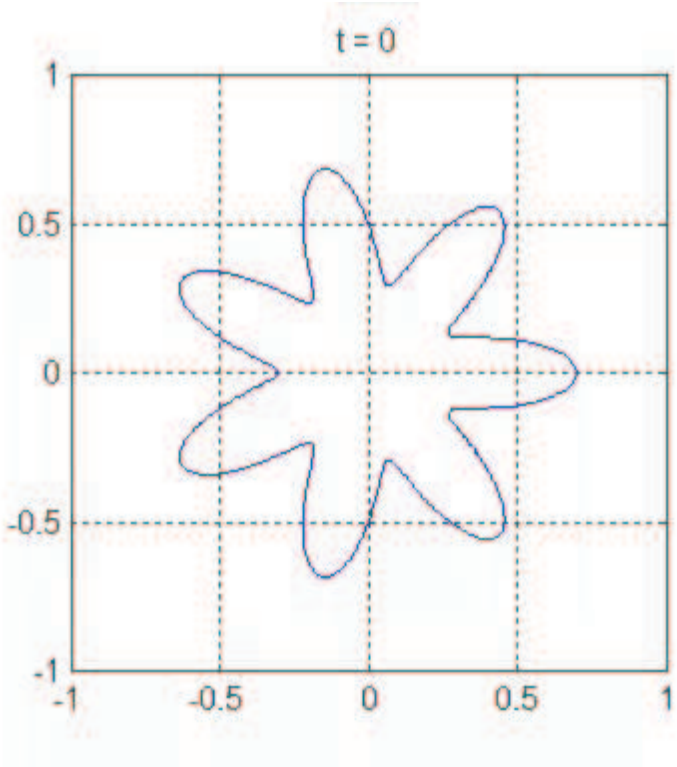


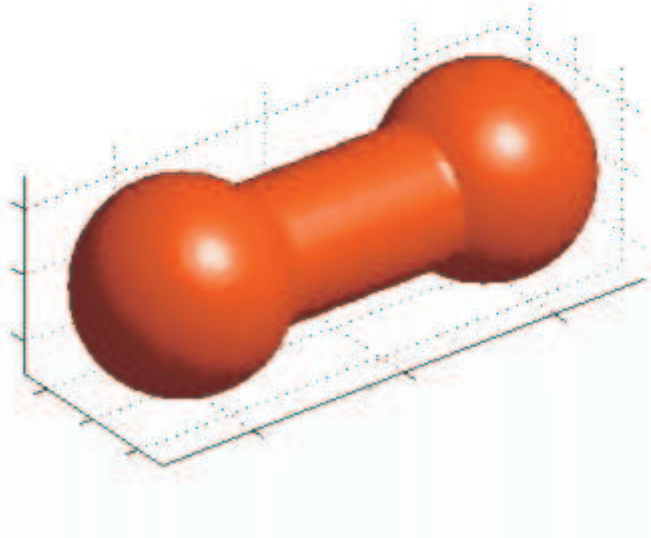


Outline

- Reach sets for control verification
  - Filtering pilot commands for safety
- Dynamic surfaces
  - Particle level set
  - Free surface fluid simulation
- Robotic path planning
  - Shortest continuous path, subject to constraints
  - Optimality for multiple vehicle and/or various norms
- Hamilton-Jacobi PDEs
  - The common link

Dynamic Interfaces





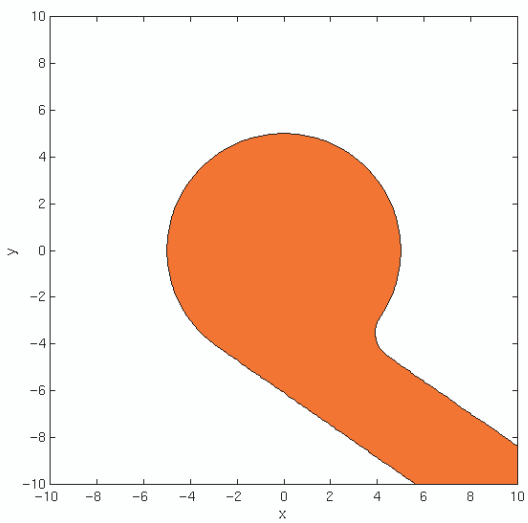
- How do you represent an evolving interface?
  - Size may grow or shrink
  - Regions may combine or pinch off
  - Interface is a curve in 2D, a surface in 3D
- One solution: dynamic implicit surfaces

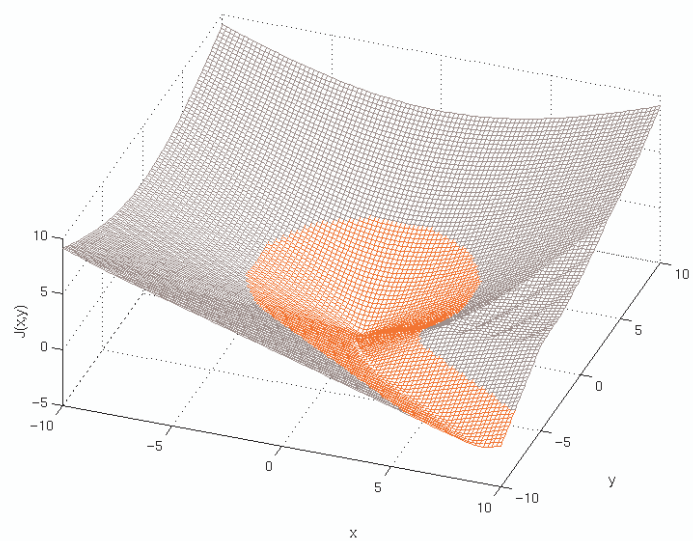
Implicit Surface Functions

- Surface $S(t)$ is defined implicitly by zero isosurface of a scalar function $\phi(x,t)$, with several benefits
  - Surfaces automatically merge and/or separate
  - Geometric quantities (normal, curvature) are easy to calculate
  - Most common surface motions become PDE in ϕ

$$\phi : \mathbb{R}^n \times \mathbb{R} \rightarrow \mathbb{R}$$

$$S(t) = \{x \in \mathbb{R}^n \mid \phi(x,t) = 0\}$$

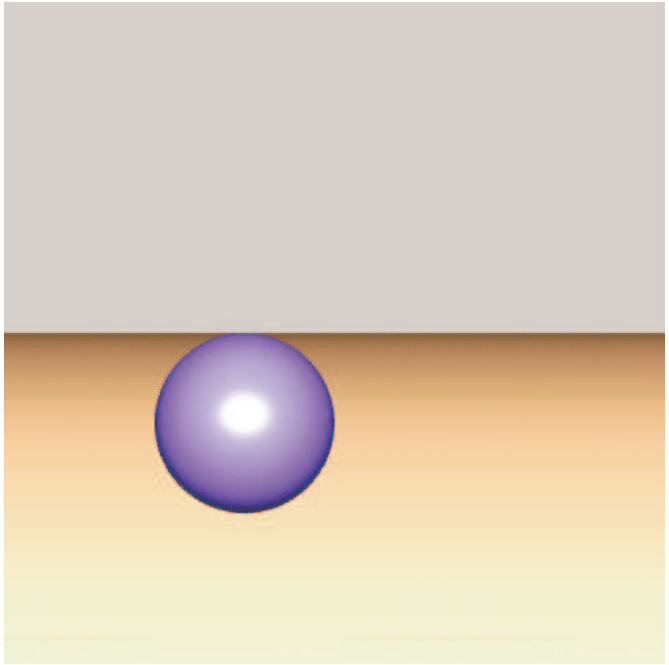




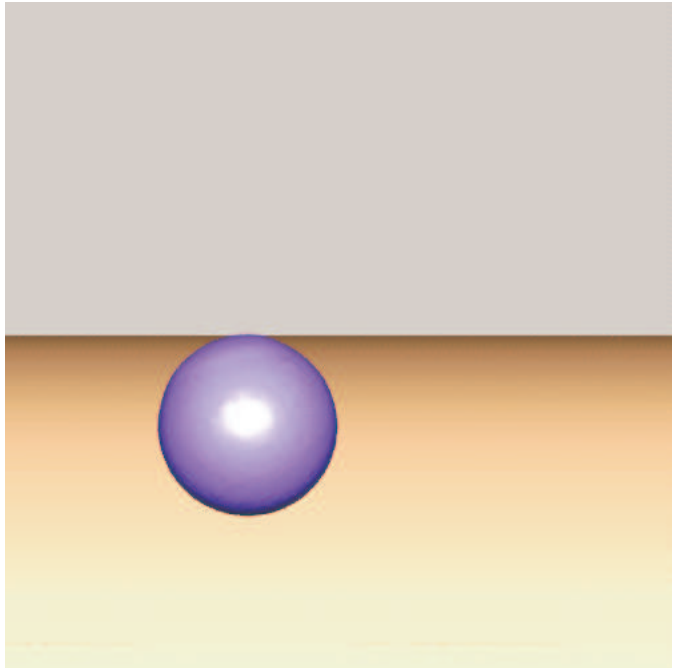
Level Set Methods

- Challenge: how can we conserve volume despite small features and numerical error?

Level Set Only



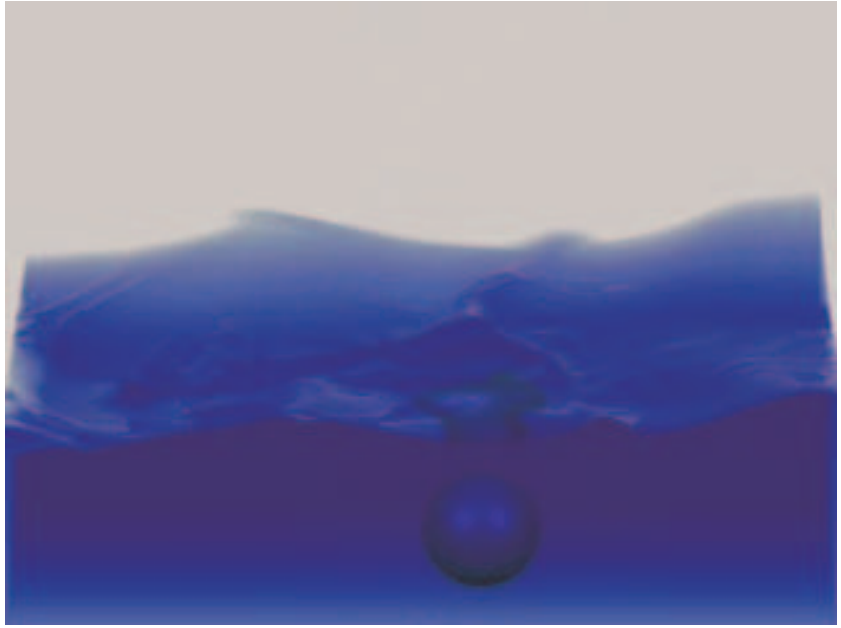
Particle Level Set



Rendering by Sou Cheng Choi

Application: Animating Fluids

- State of the art evolving interface
  - Merging and separating surfaces
  - Smooth simulation and rendering of fluid and container
  - Plausible water motion
- Requires fluid simulation as well (not my research)

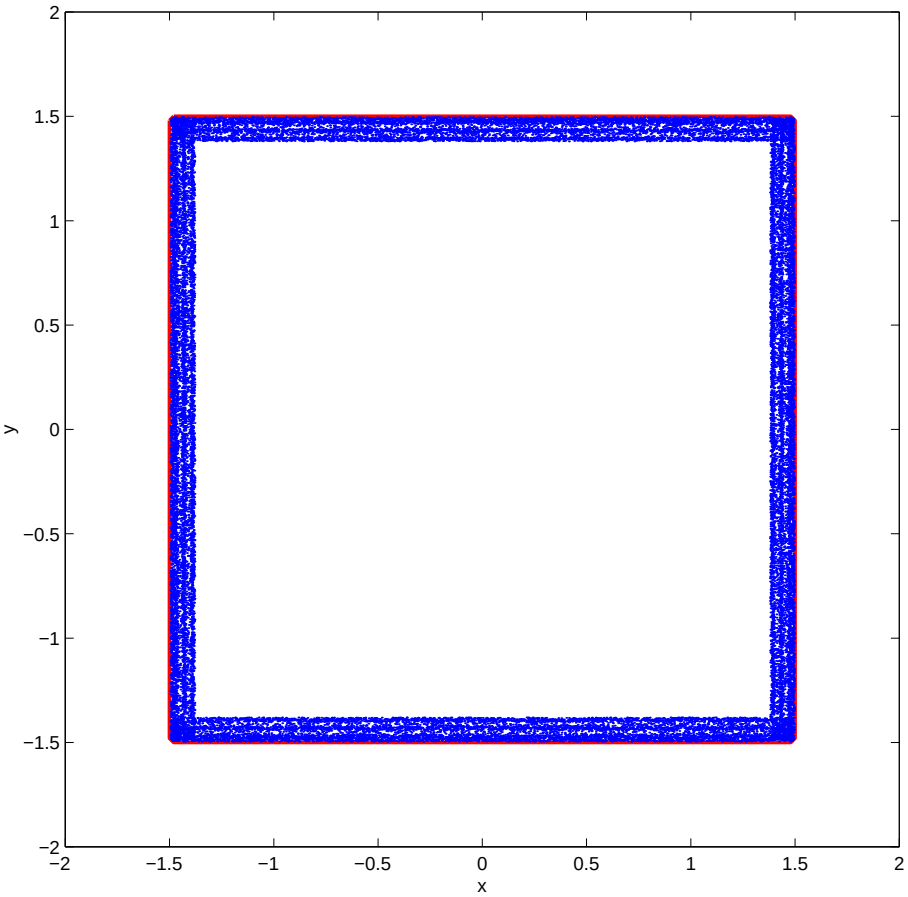


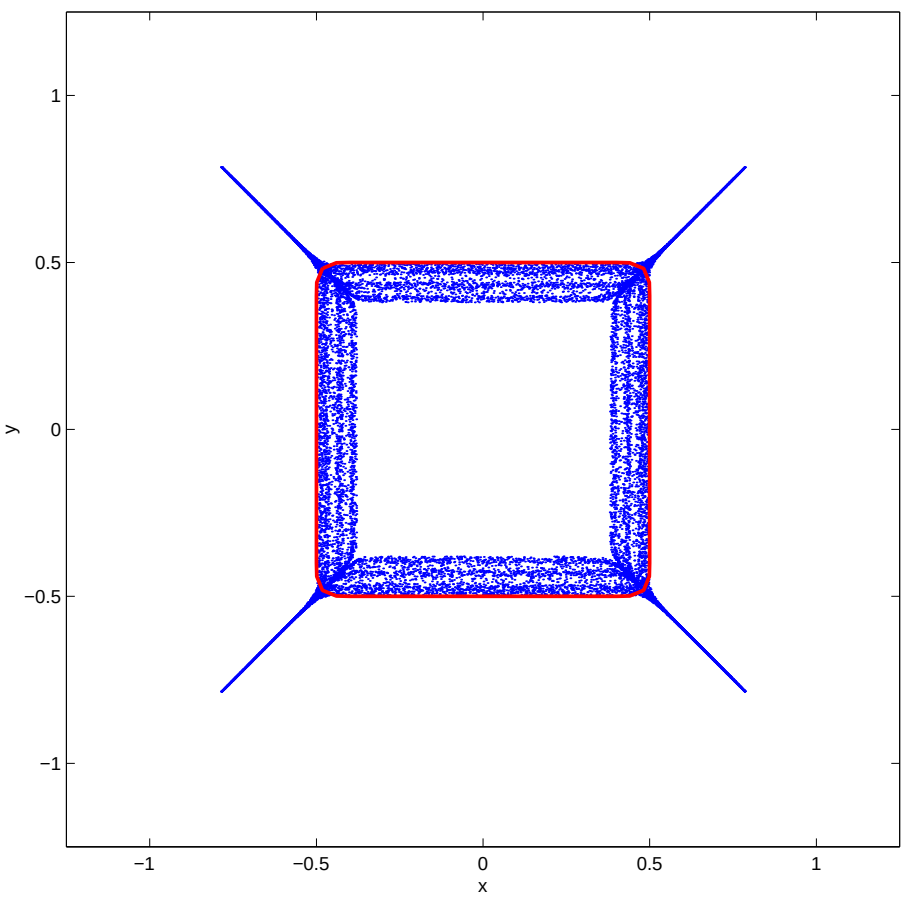
Foster & Fedkiw (2001)

Enright, Marschner & Fedkiw (2002)

Not Finished Yet

- Reports of dubious repeatability.
- What about shocks? Particle methods fail.



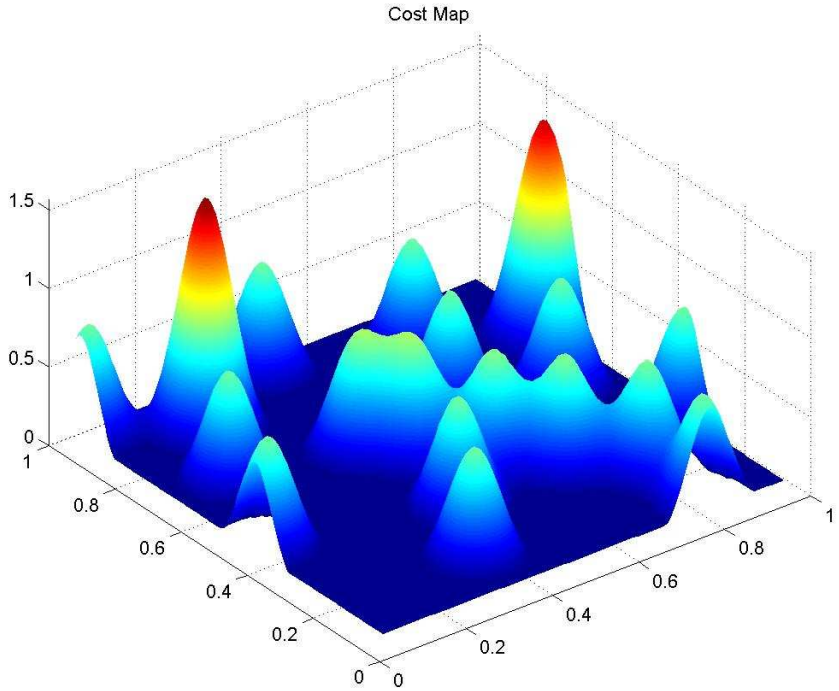


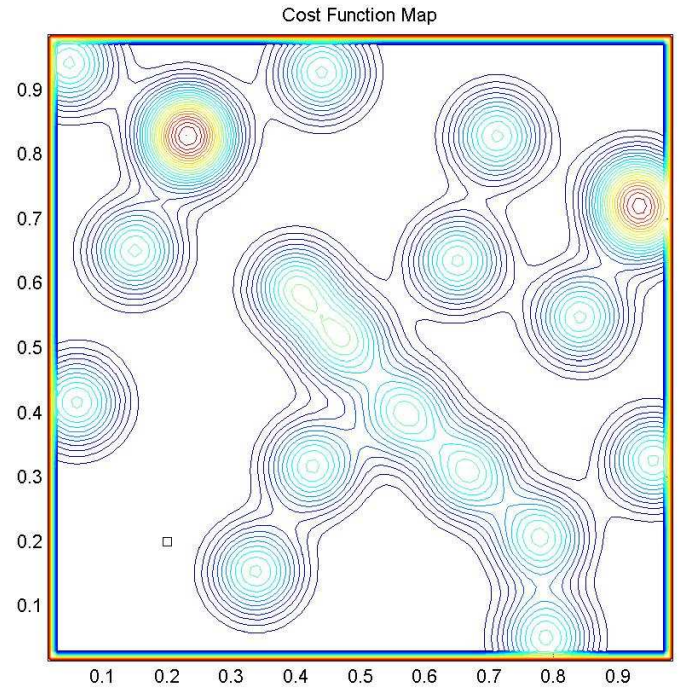
Outline

- Reach sets for control verification
  - Filtering pilot commands for safety
- Dynamic surfaces
  - Particle level set
  - Free surface fluid simulation
- Robotic path planning
  - Shortest continuous path, subject to constraints
  - Optimality for multiple vehicle and/or various norms
- Hamilton-Jacobi PDEs
  - The common link

Basic Path Planning

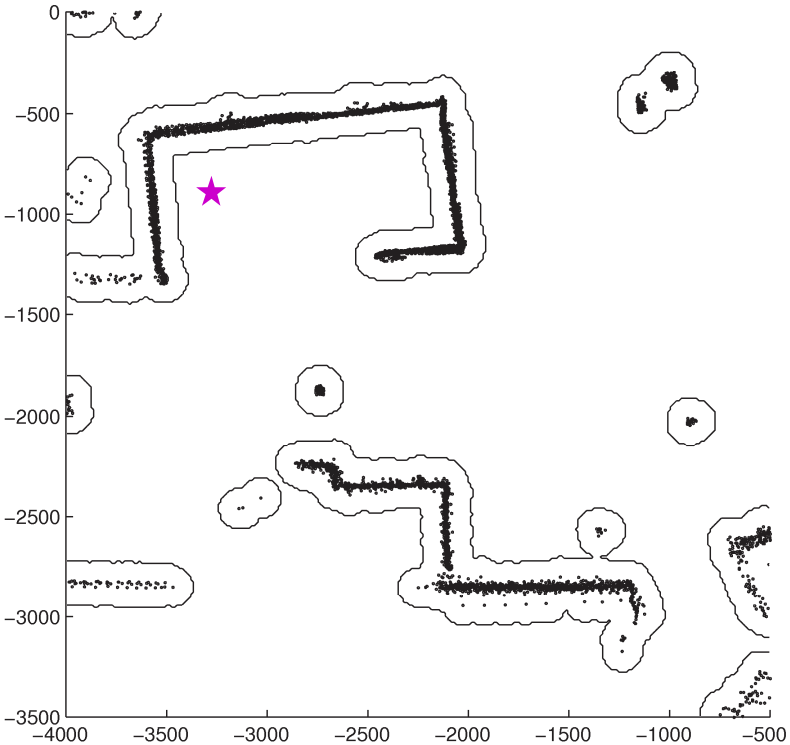
- Find the optimal path $p(s)$ to a target (or from a source)
- Inputs
  - Cost to pass through each state in the state space
  - Set of targets or sources (provides boundary conditions)



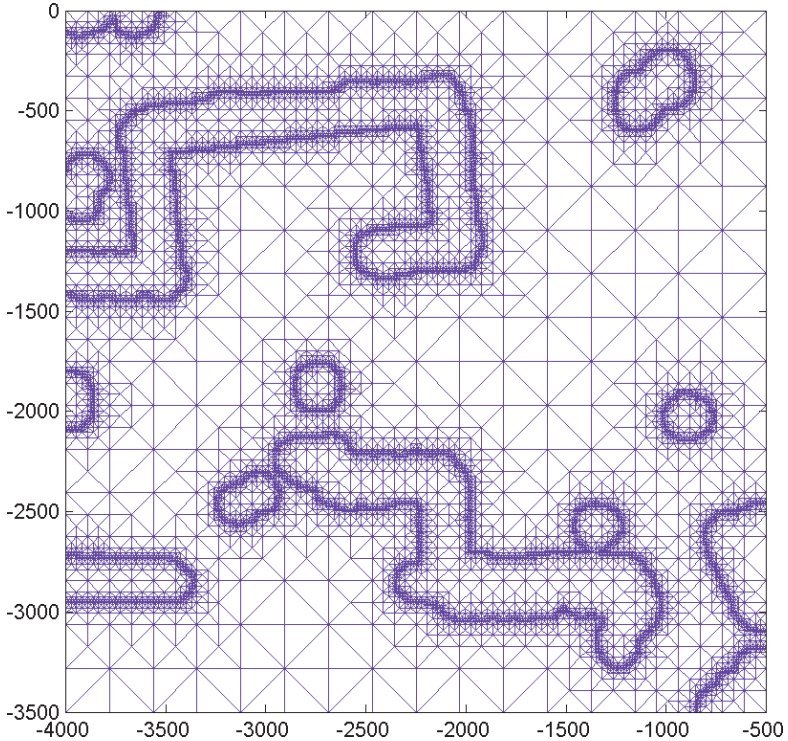


Robot Path Planning

- Find shortest path to objective while avoiding obstacles
  - Obstacle maps from laser scanner
  - Configuration space accounts for robot shape
  - Cost function essentially binary



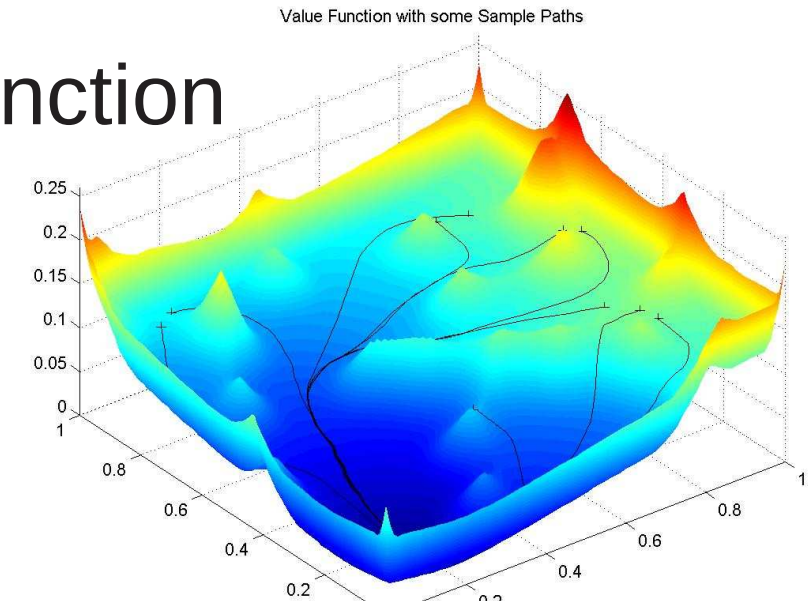
typical laser scan with configuration space obstacles

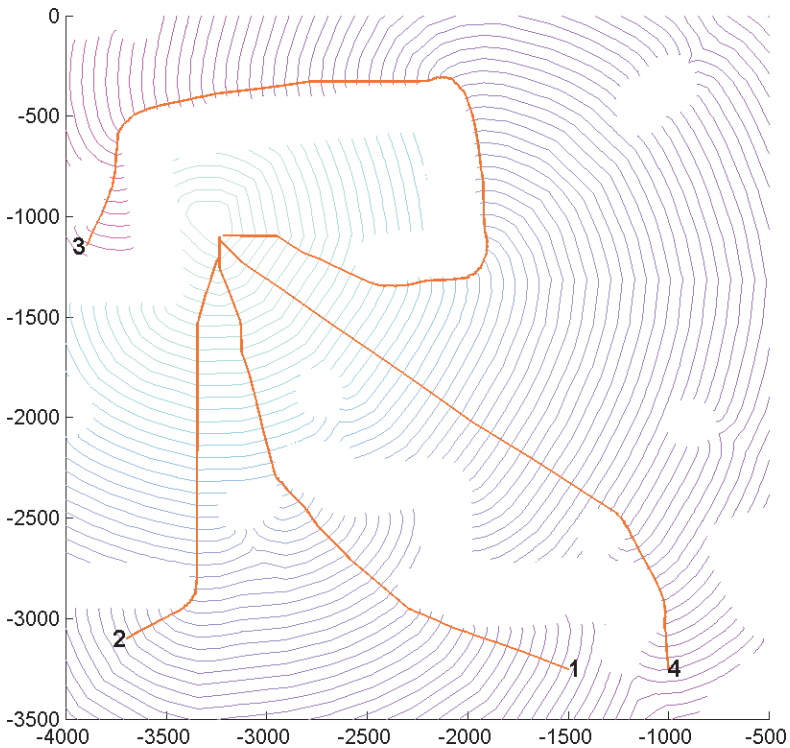


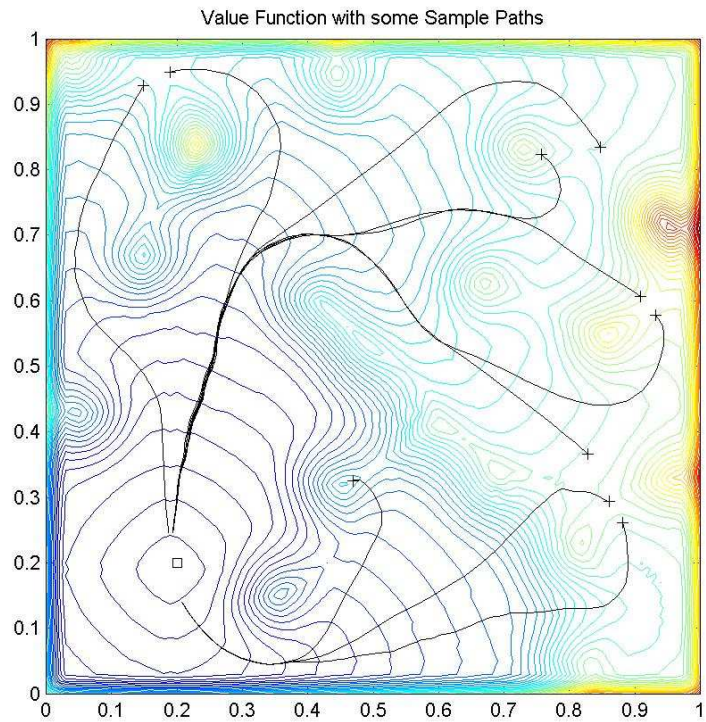
adaptive grid

Value Function

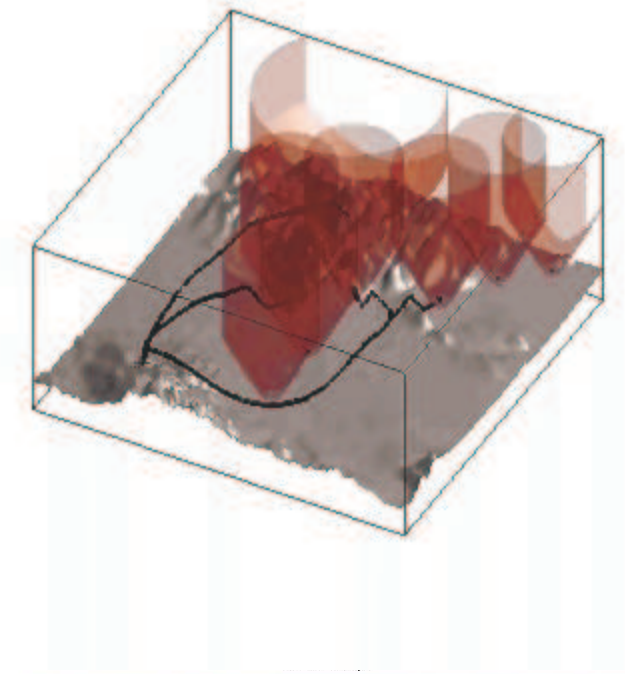
- Specifies cost of optimal path to target from any point
- Steepest descent finds optimal path

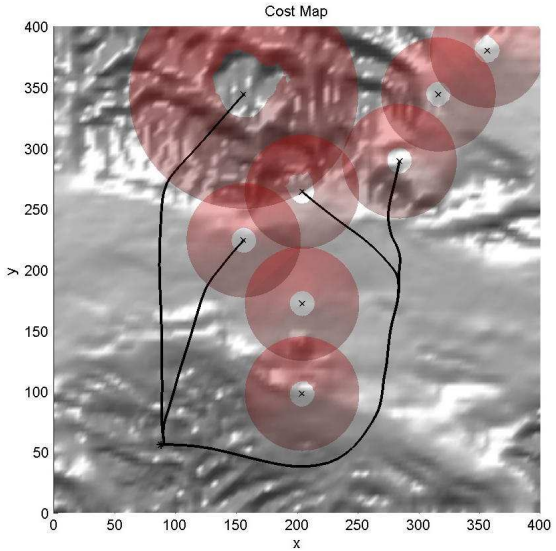


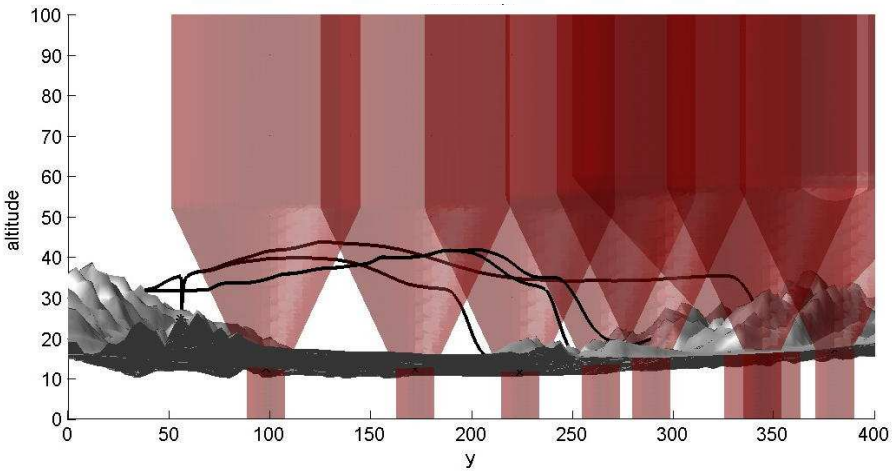


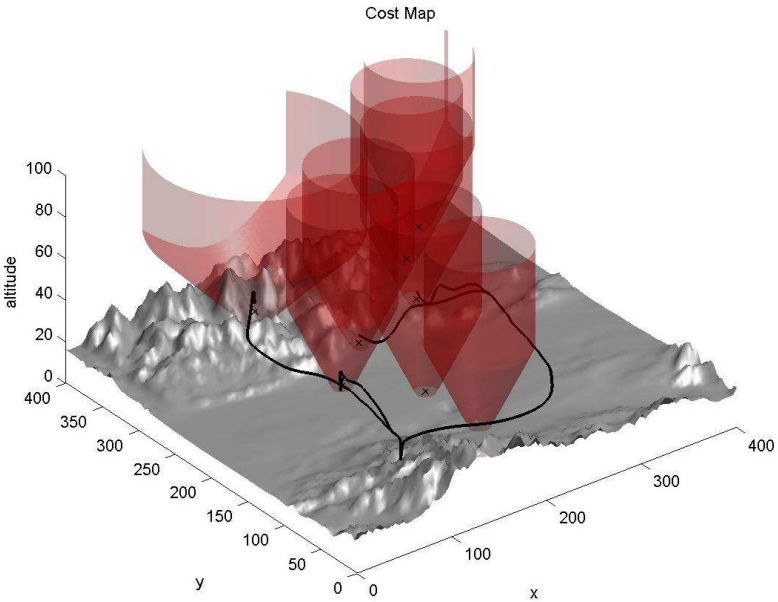


Demanding Example? No!





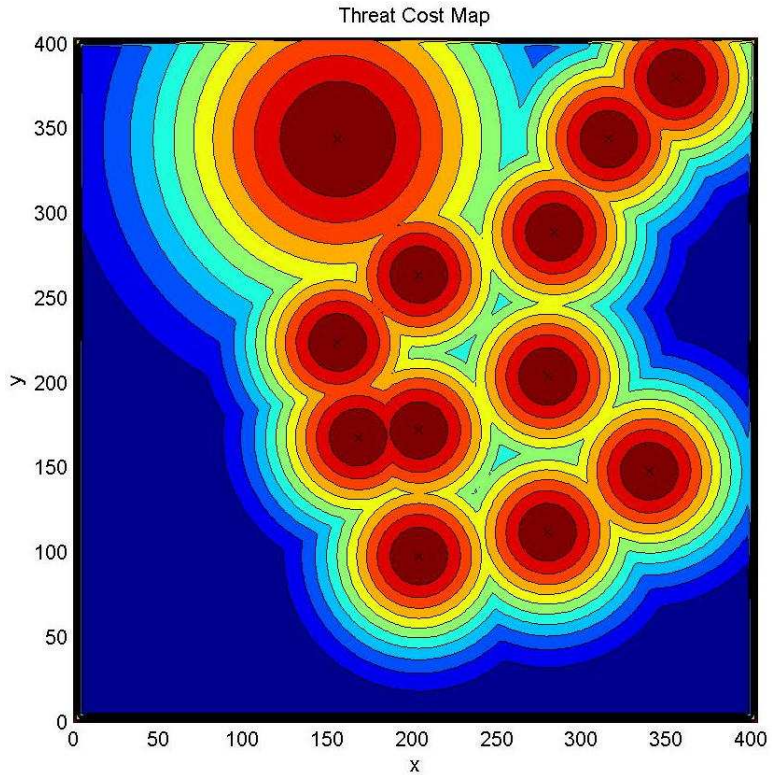




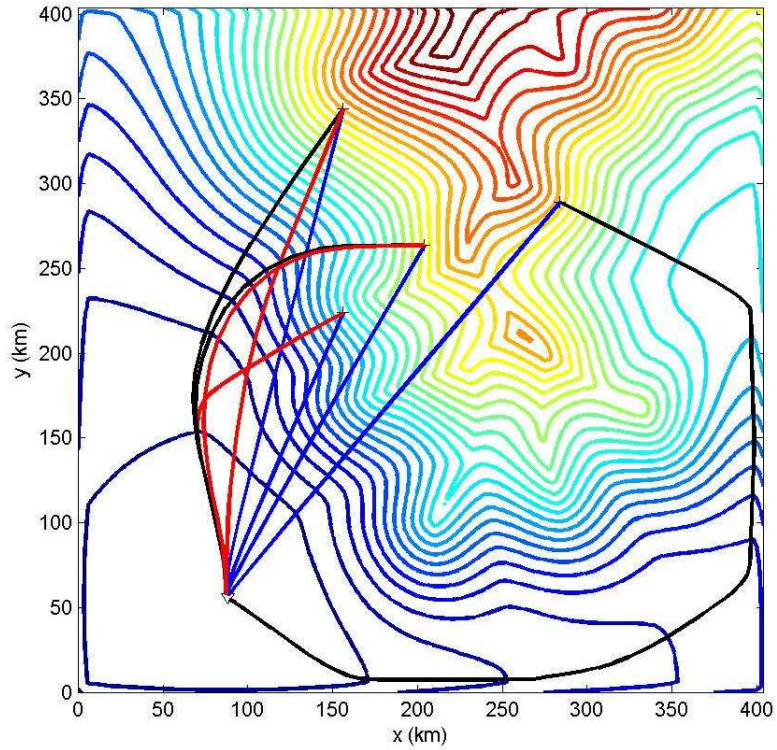
C Computer

Constrained Example

- Plan path to selected sites
  - Threat cost function is maximum of individual threats
- For each target, plan 3 paths
  - minimum threat, minimum fuel, minimum threat (with fuel ≤ 300)



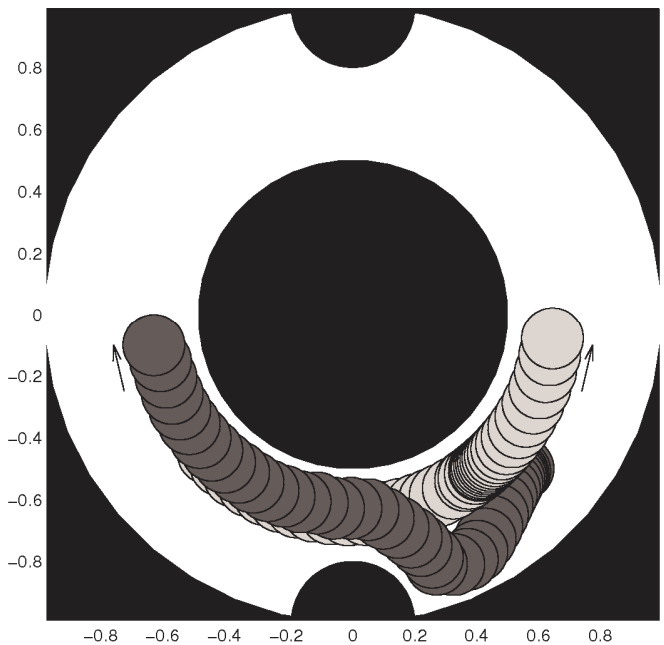
threat cost

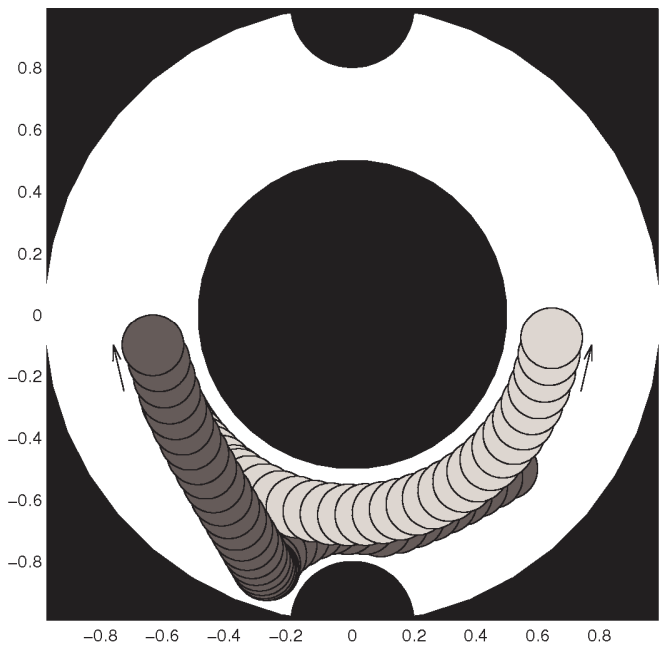


Paths (on value function)

Mixtures of Norms

- May even be situations where action norm bounds are mixed
  - Dark shaded robot starts on right, may move any direction in 2D
  - Light shaded robot starts on left, constrained to 1D circular path
  - Cost encodes black obstacles and collision states
  - 2D robot action constrained in $||\cdot||_2$ and combined action in $||\cdot||_\infty$





Outline

- Reach sets for control verification
  - Filtering pilot commands for safety
- Dynamic surfaces
  - Particle level set
  - Free surface fluid simulation
- Robotic path planning
  - Shortest continuous path, subject to constraints
  - Optimality for multiple vehicle and/or various norms
- Hamilton-Jacobi PDEs
  - The common link

Evolving Reachable Sets

- Modified Hamilton-Jacobi partial differential equation

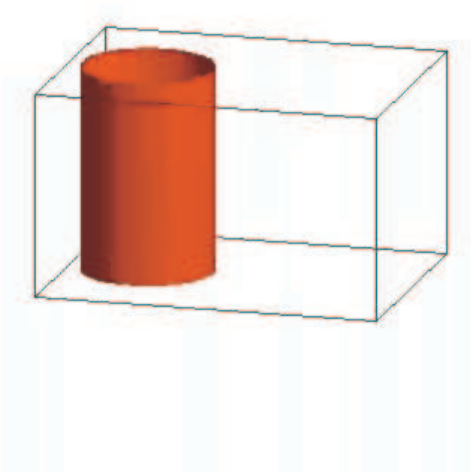
$$D_t\phi(x,t) + \min\left[0, H(x, D_x\phi(x,t))\right] = 0$$

$$\text{with Hamiltonian}: \; H(x,p) = \max_{a\in\mathcal{A}} \min_{b\in\mathcal{B}} f(x,a,b)\cdot p$$

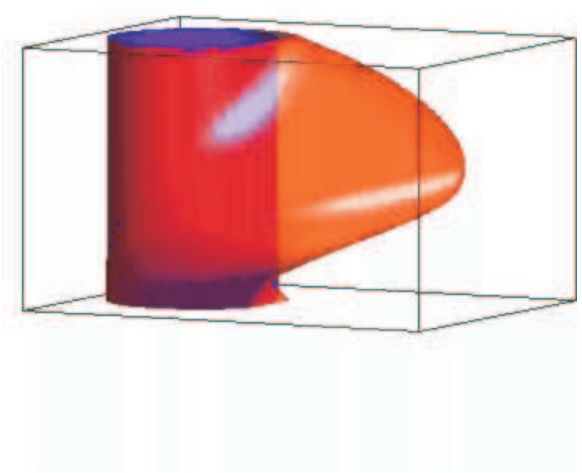
$$\text{and terminal conditions}: \; \phi(x,0) = h(x)$$

$$\text{where} \quad G(0) = \{x \in \mathbb{R}^n \mid h(x) \leq 0\}$$

$$\text{and} \quad \dot{x} = f(x,a,b)$$

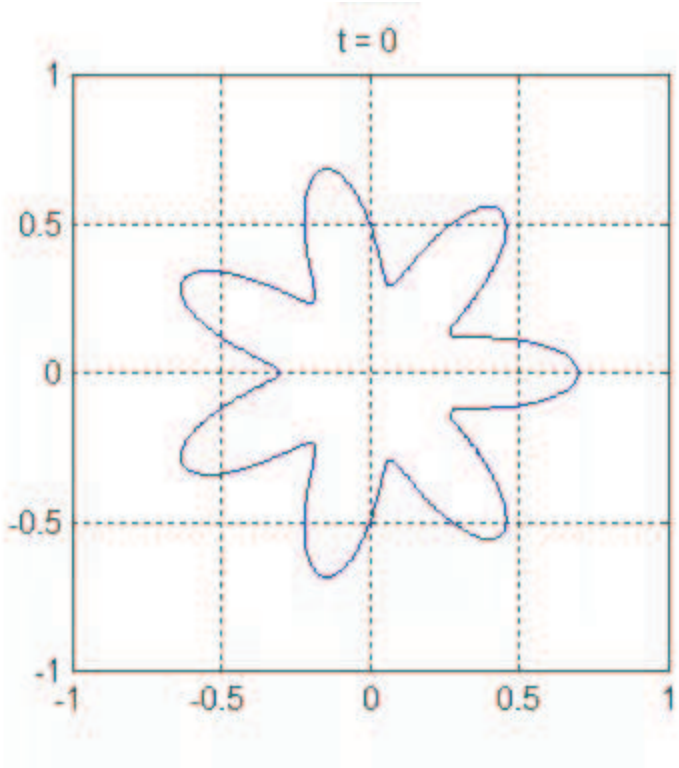


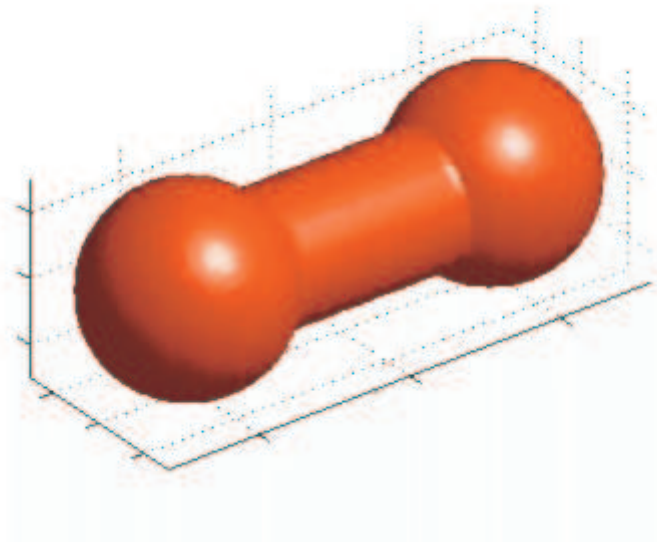
growth of reachable set



final reachable set

Dynamic Implicit Surfaces





$$D_t\phi(x,t) + a(x,t)\|\nabla\phi(x,t)\| + v(x,t)\cdot\nabla\phi(x,t) = 0$$

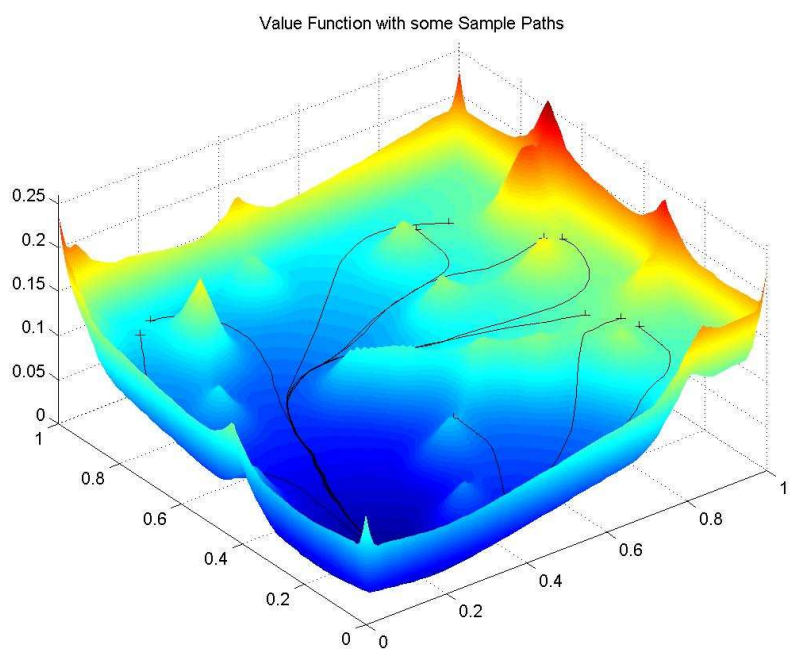
$$D_t\phi(x,t) - \kappa(x,t)\|\nabla\phi(x,t)\| = 0$$

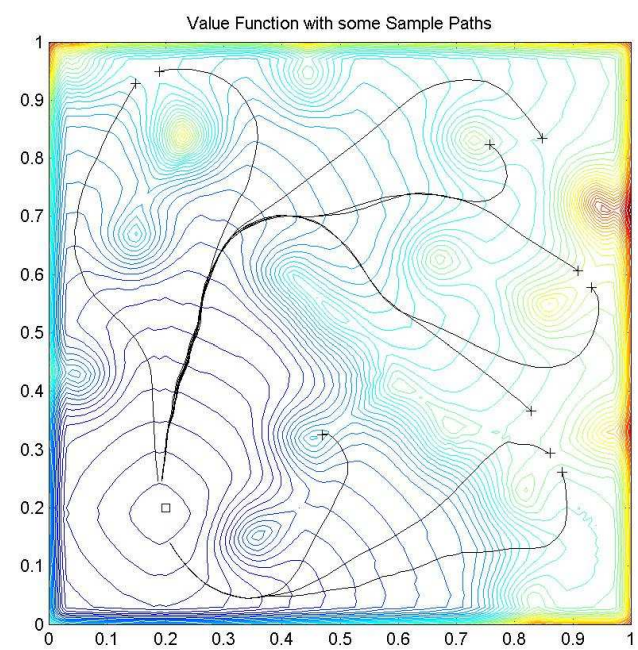
Path Generation

- Value function $V(x)$ is solution of Eikonal equation
  - Solution may not be differentiable
- Optimal path $p(s)$ is found by gradient descent
  - Value function $V(x)$ has no local minima, so paths will always terminate at a target

$$\|\nabla V(x)\| = c(x)$$

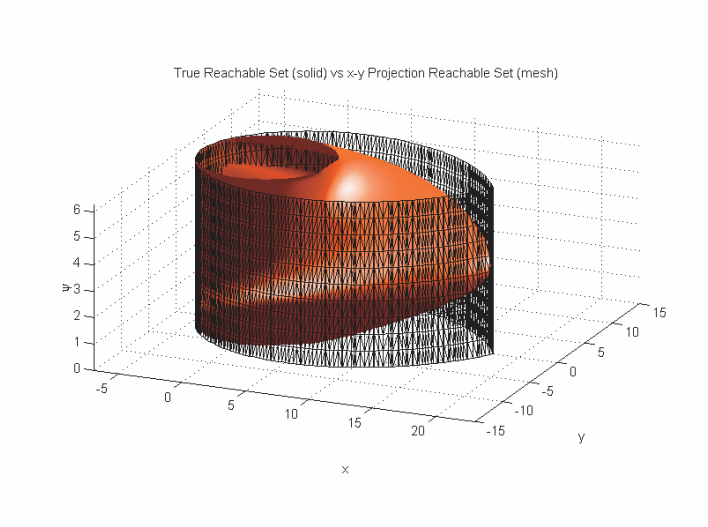
$$\frac{dp}{ds} = \frac{\nabla V(x)}{\|\nabla V(x)\|}$$

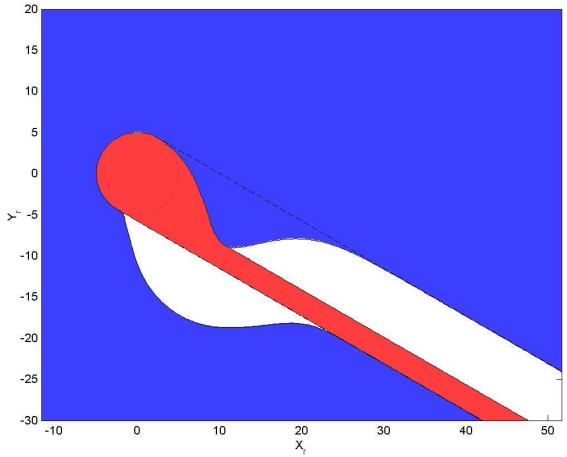


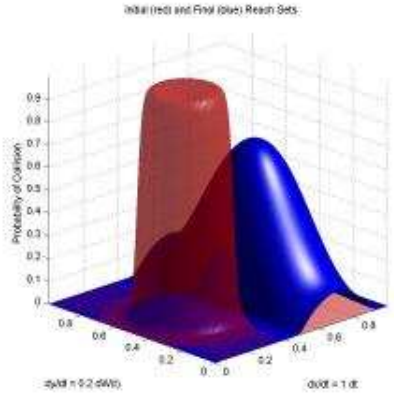


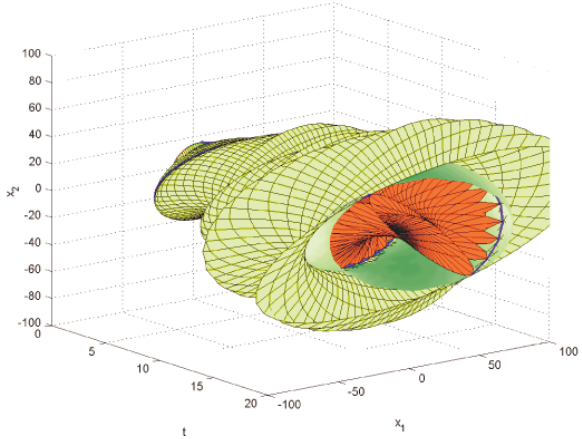
Not Discussed

- Mathematical Finance & Ecology
- Image Processing
- Embedded and real-time platforms
- Methods for higher dimensions
- Hybrid and Stochastic Reach sets









mitchell@cs.ubc.ca

http://www.cs.ubc.ca/~mitchell