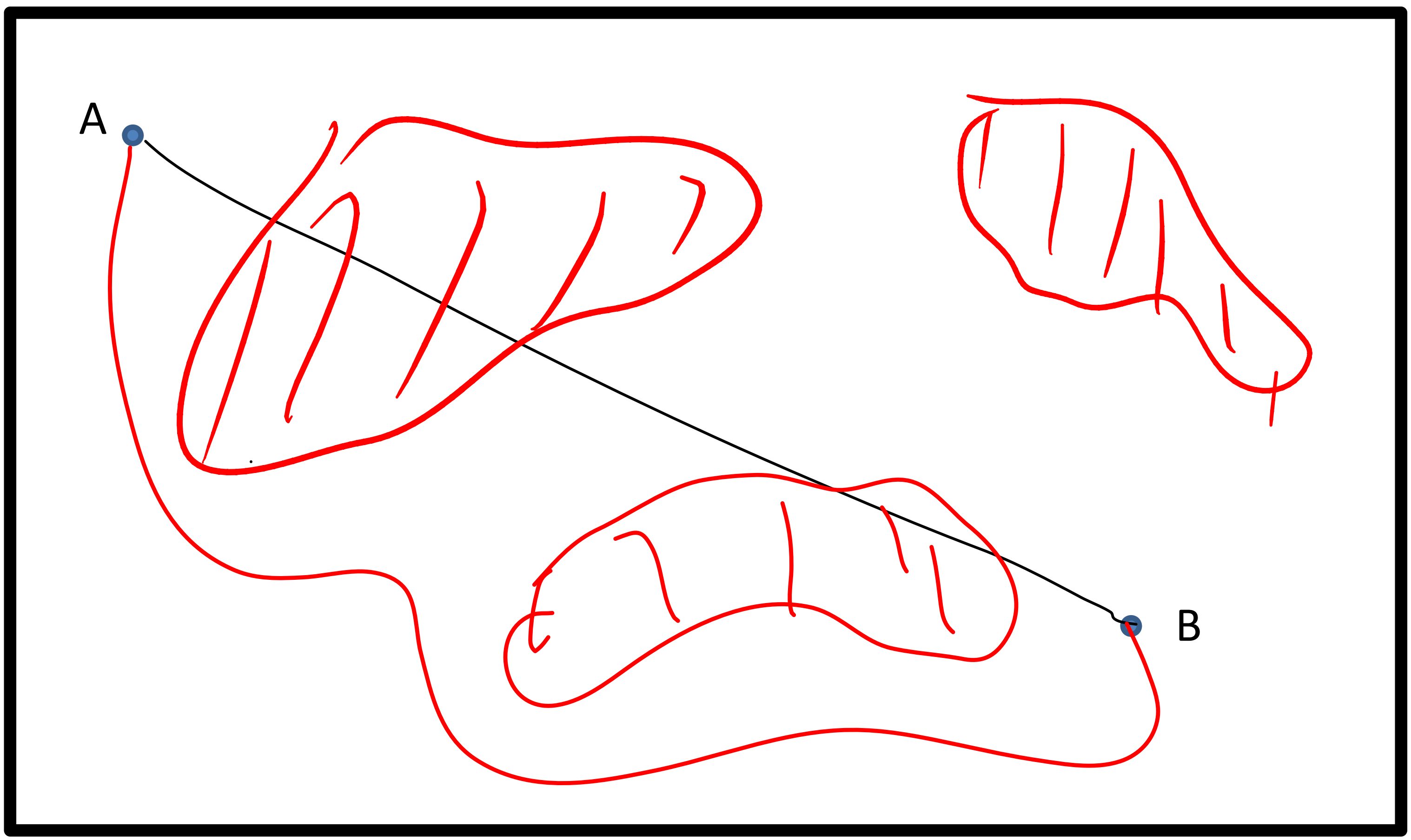
Motion and path planning in a nutshell

Prof. Hadas Kress-Gazit
MAE
Guest lecture: CS 4758/6758
March 15, 2012

easy



not as easy

“How do I get to point B?”

- Motion planning
  - Bug algorithms
  - Roadmaps, cell decomposition
  - Potential functions
  - Sampling-based methods

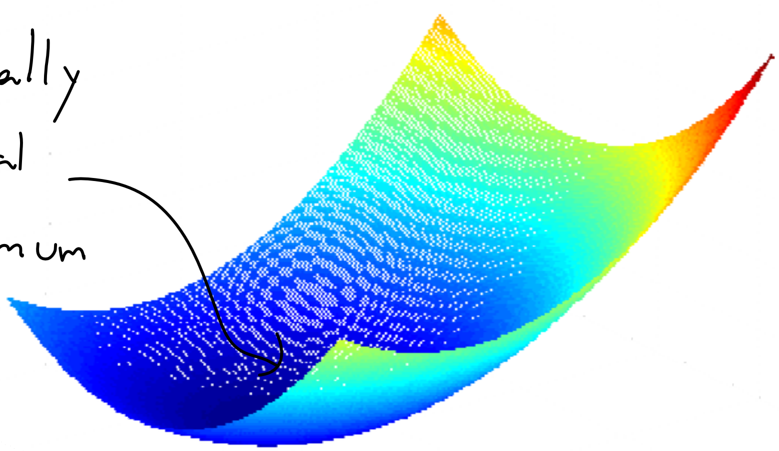
“How do I get to point B?”

- Motion planning
  - Bug algorithms
  - Roadmaps, cell decomposition
  - **Potential functions (“vanilla” potential functions)**
  - **Sampling-based methods (RRT)**

Potential functions – basic idea

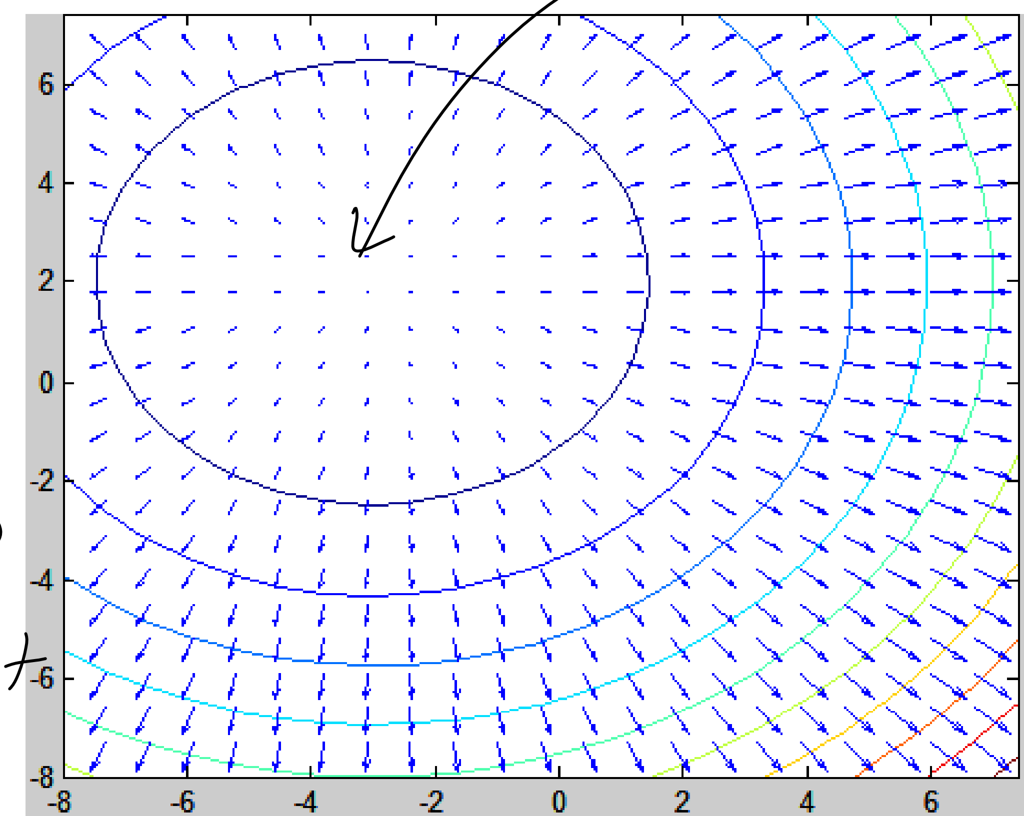
energy function over C_{free} (obstacle free configuration space)

Ideally global minimum at goal



goal

gradient



Definitions

$q \in \mathbb{R}^n$
configuration

- Potential function $U : \mathbb{R}^n \longrightarrow \mathbb{R}$

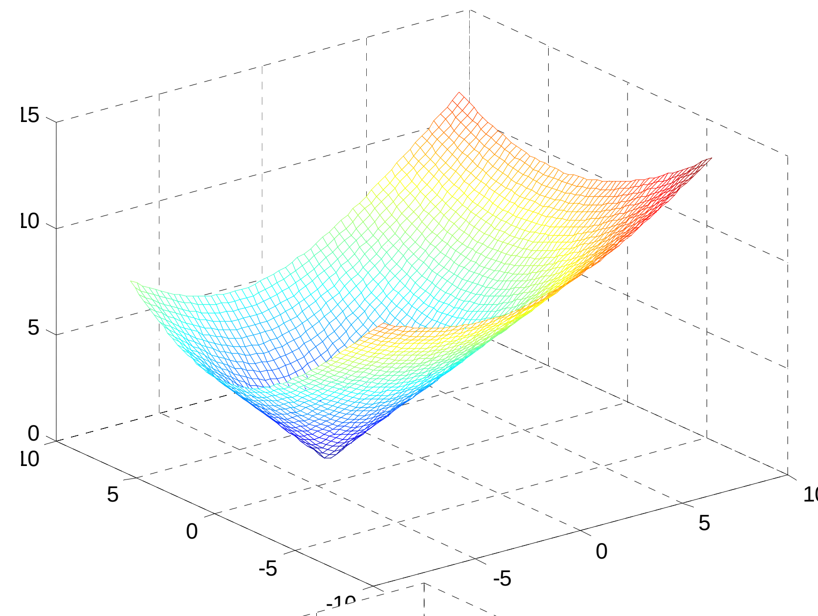
- Gradient $\nabla U(q) = \begin{bmatrix} \frac{\partial U}{\partial q_1}(q) \\ \vdots \\ \frac{\partial U}{\partial q_n}(q) \end{bmatrix}$

- Control

$$\dot{q} = -\nabla U(q)$$

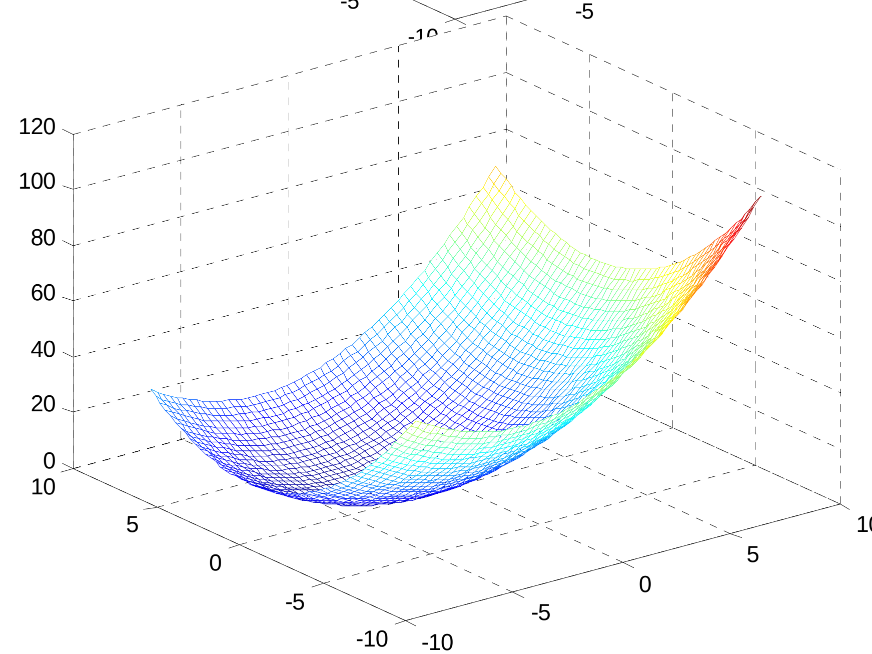
Attractive force = go to goal

$d(q, q_{goal}) = \text{norm}(q - q_{goal})$



$U_{att} = C \cdot d(q, q_{goal})$

$\nabla U_{att} = \frac{C}{d(q, q_{goal})} \cdot (q - q_{goal})$



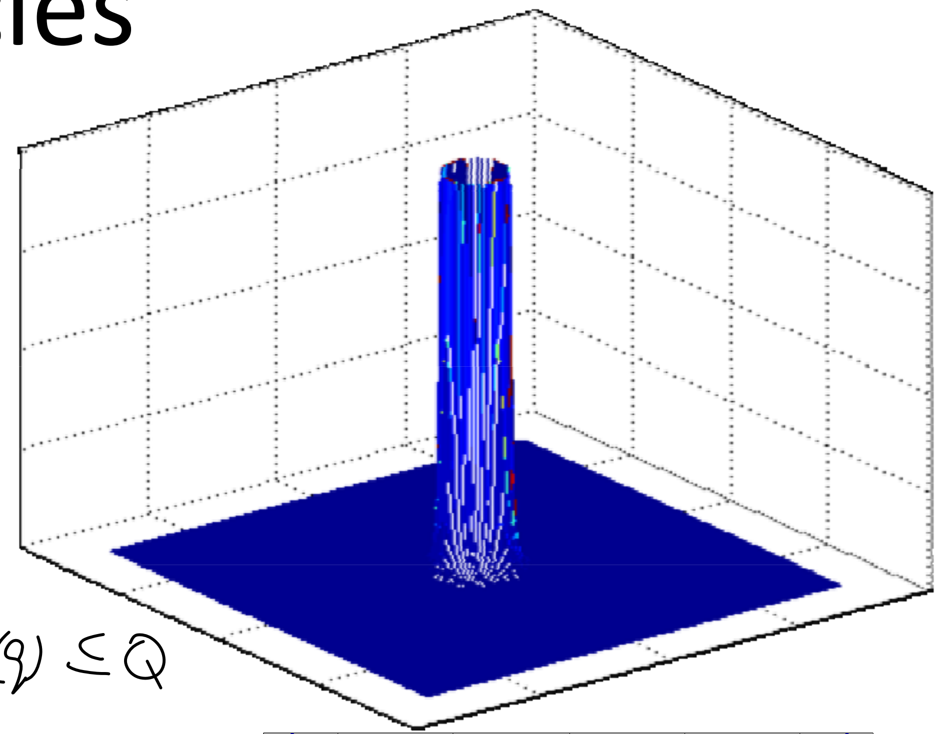
$U_{att} = \frac{1}{2} C \cdot d(q, q_{goal})^2$

$\nabla U_{att} = C \cdot (q - q_{goal})$

Repulsive force = keep away from obstacles

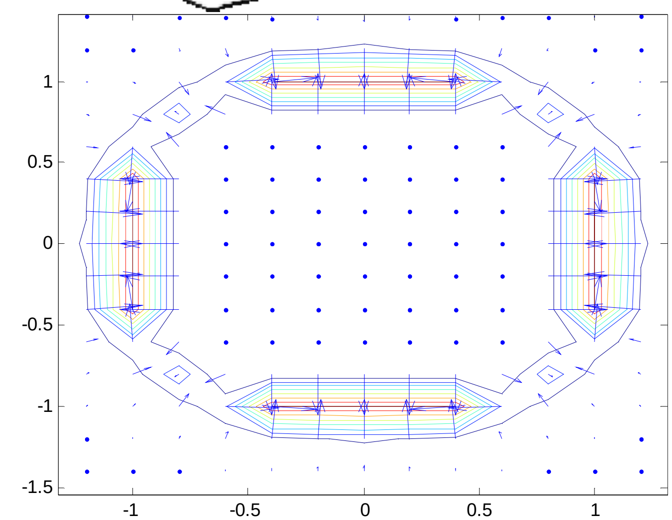
Distance from obstacle

$$d_i(q) = \min_{q^* \in obs_i} d(q, q^*)$$



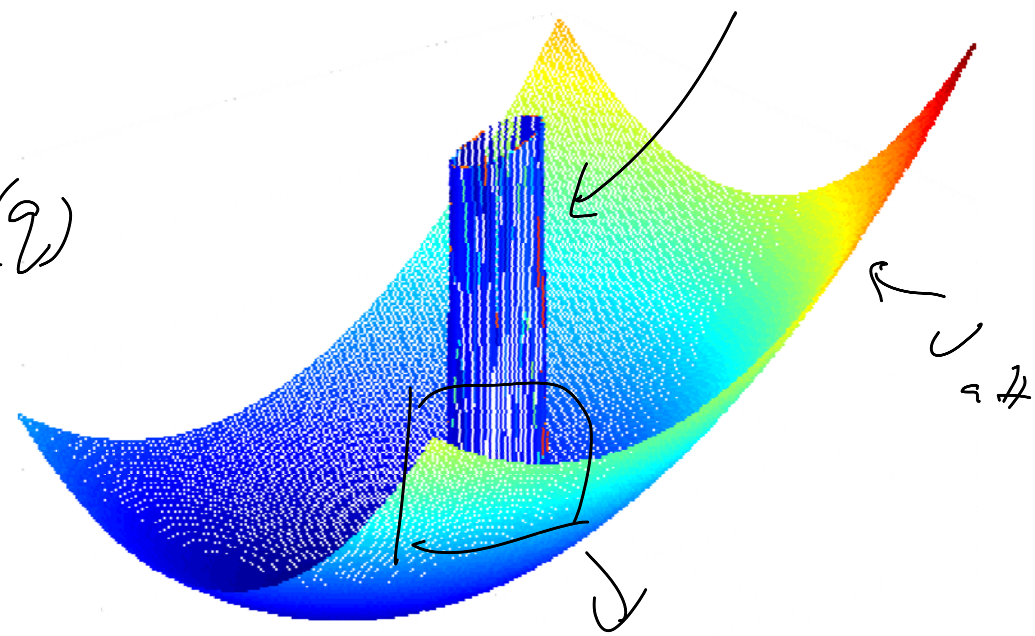
$$U_{rep_i} = \begin{cases} \frac{1}{2} c \left(\frac{1}{d_i(q)} - \frac{1}{Q} \right)^2 & d_i(q) \le Q \\ 0 & d_i(q) > Q \end{cases}$$

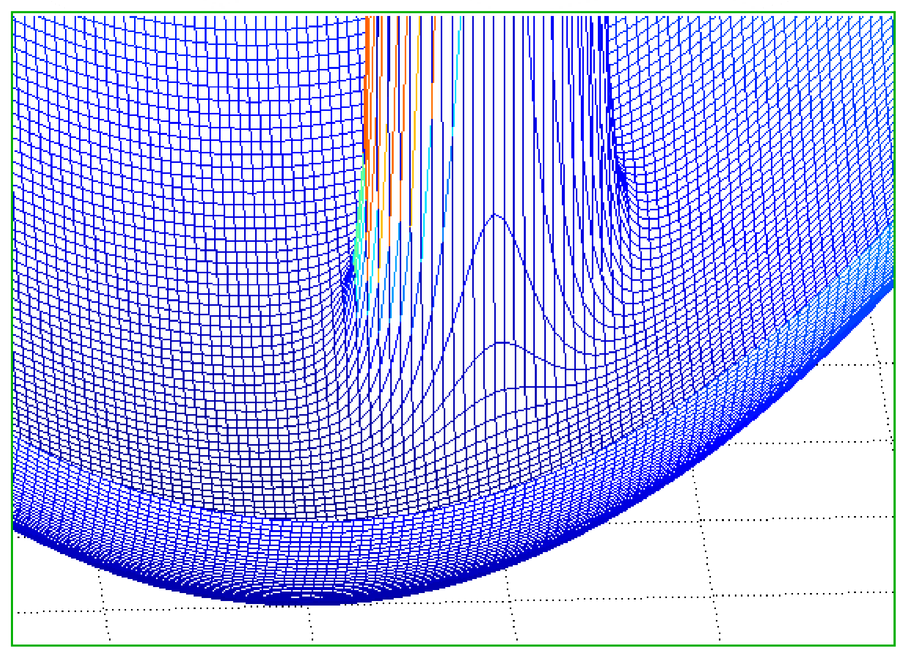
$Q > 0$



Potential function

$$U(q) = U_{att} + \sum_{obs_i} U_{rep_i}(q)$$





<u>pro</u>

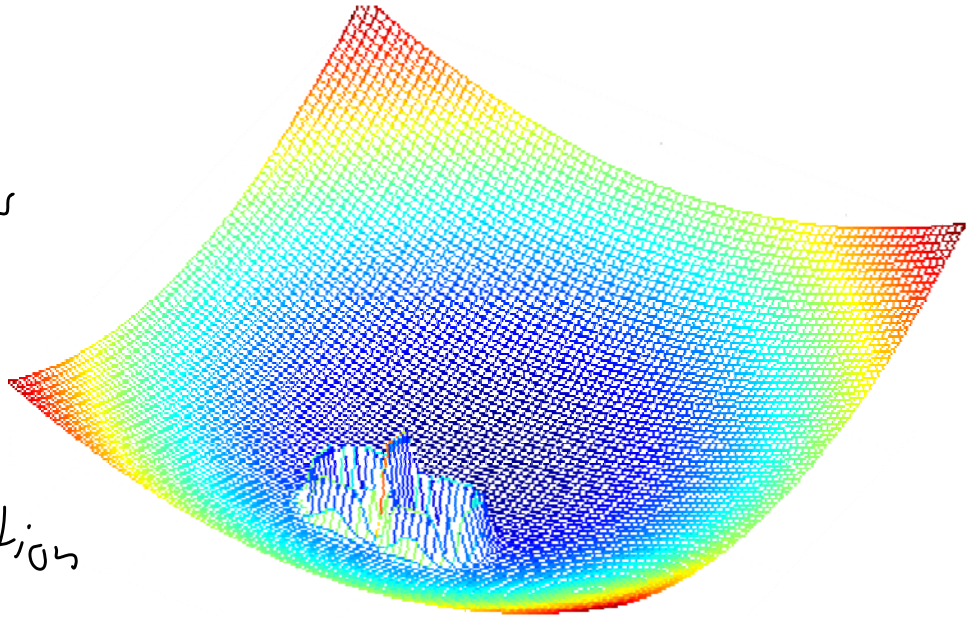
complete (every initial point will reach the goal*)

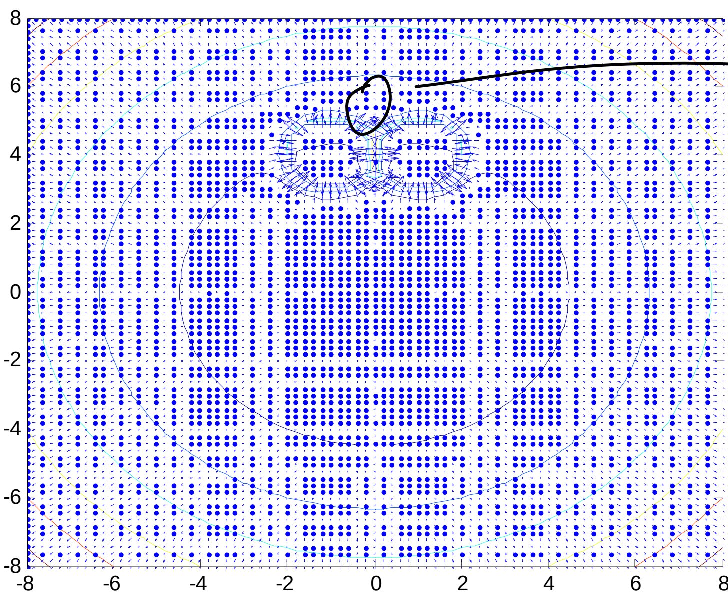
Problem

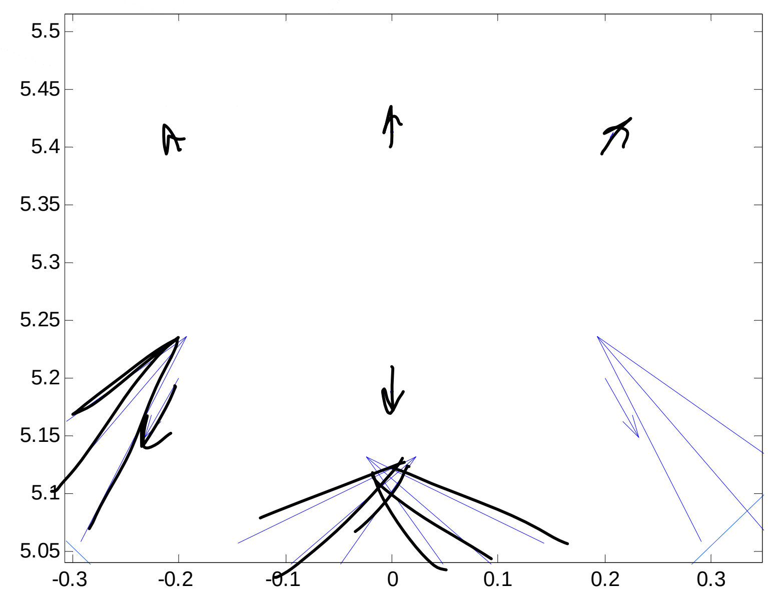
local minima!

solutions: - Navigation functions

- potential functions over cell decomposition

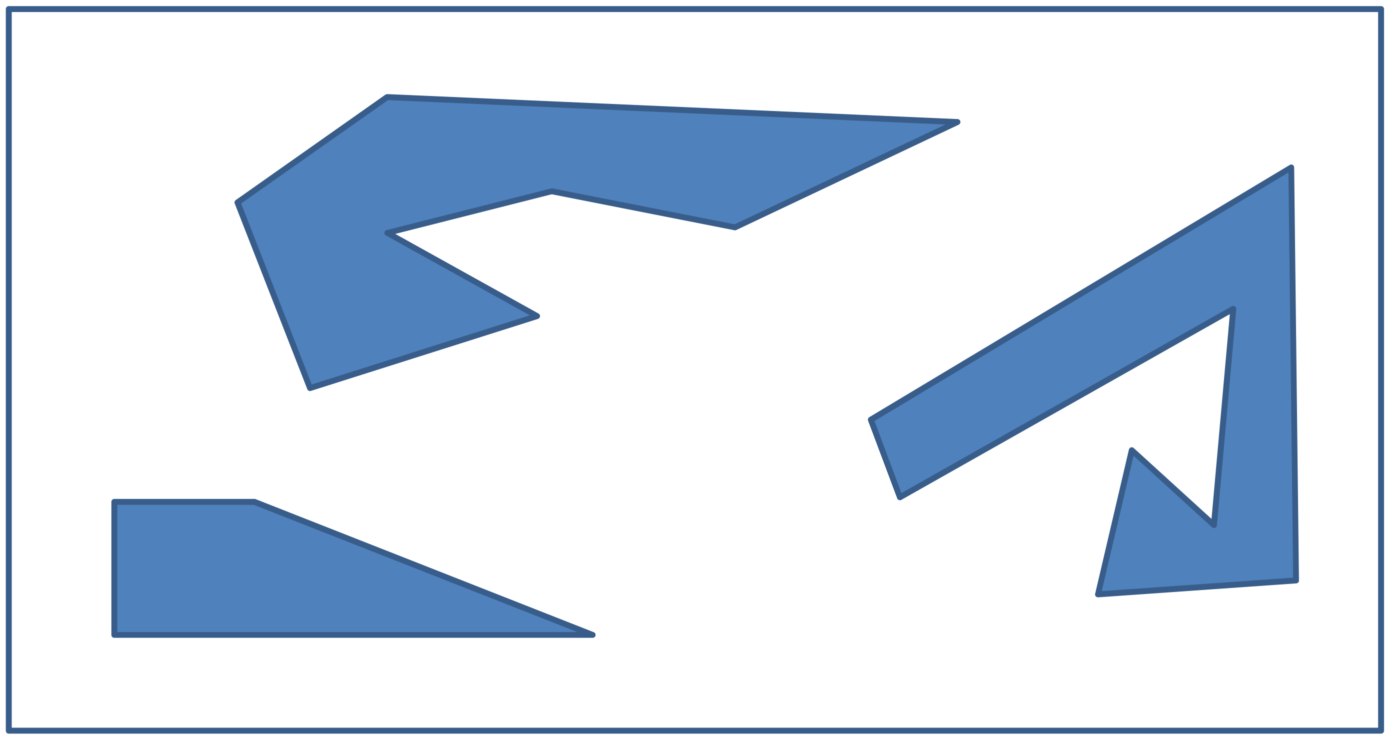




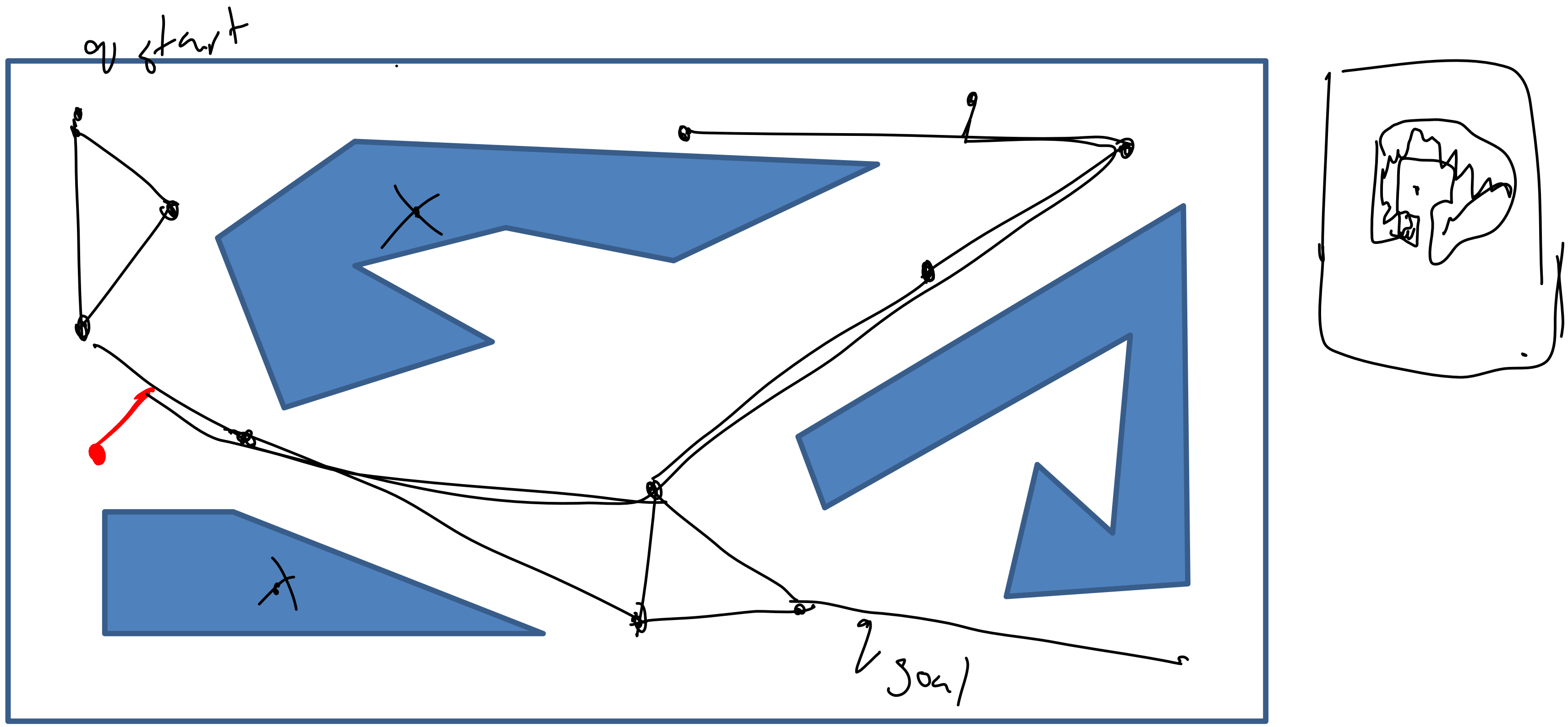


Problem (2)

Complex environment



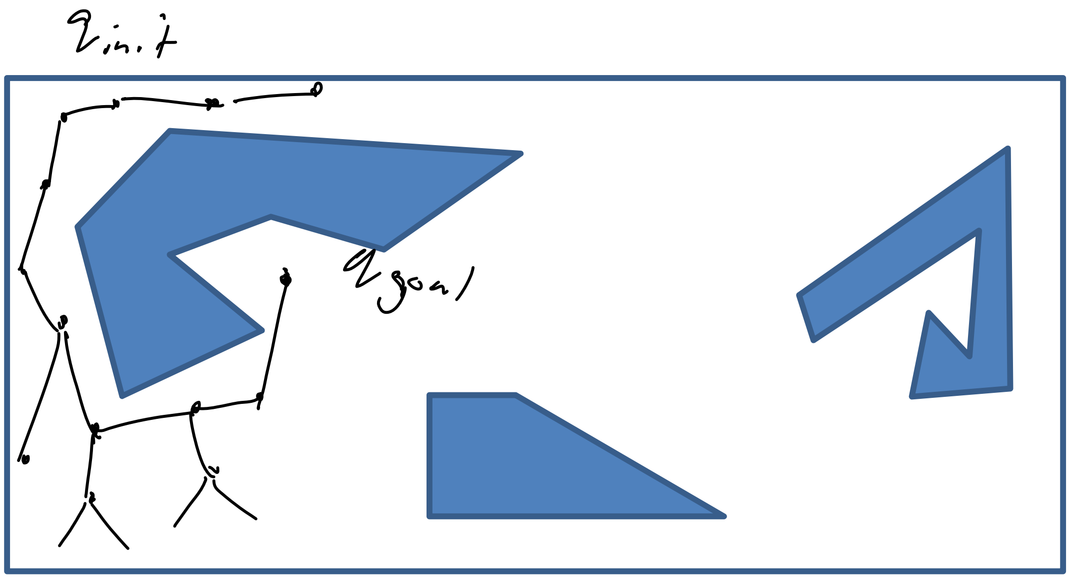
Different approach - samples



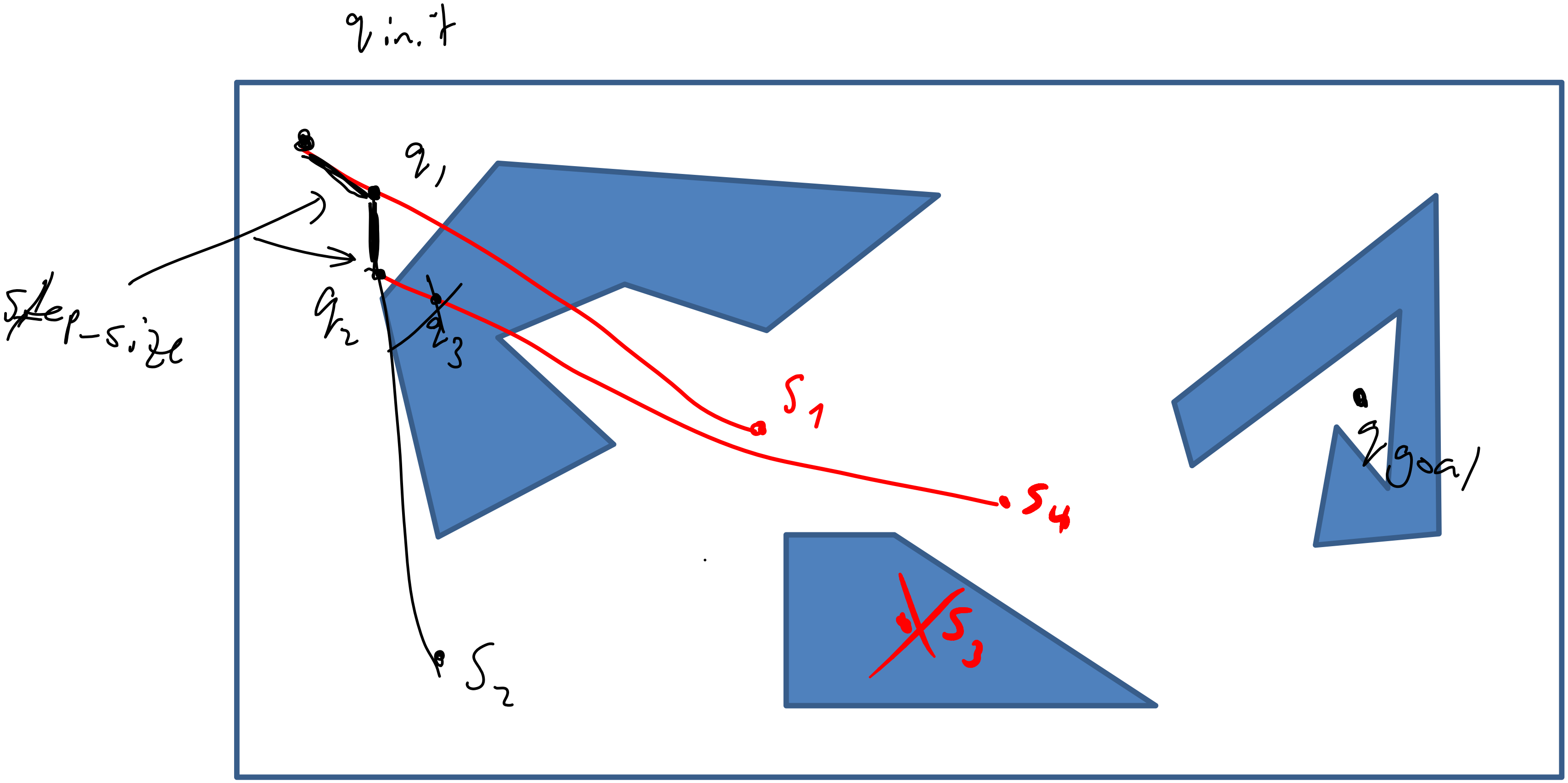
- Probabilistically\resolution complete
- Good for complex configuration spaces

Single queries

- Find a path from q_{init} to q_{goal}
- Idea:
  - grow tree(s) spanning "relevant" space
  - Connect tree(s)



Rapidly-Exploring Random Trees (RRT)



RRTs

Algorithm:

Given: q_{start}, q_{end}, step_size, n = # of attempts to grow the tree

Find: $G = (V, E)$ $v \in \mathbb{R}^n$ $e \in \mathbb{R}^n \times \mathbb{R}^n$

Init: $V = \{q_{start}\}$ $E = \emptyset$

For i=1:n

- sample $q_{rand} \in C_{free}$
- find q_{near} = closest point $q \in V$ to q_{rand}
- generate q_{new}: point on line (q_{rand}, q_{near}) that is step_size away from q_{near}
- if $q_{new} \in C_{free}$ AND $(q_{near}, q_{new}) \in C_{free}$ then $V = V \cup \{q_{new}\}$, $E = E \cup \{(q_{near}, q_{new})\}$

$\Downarrow$

how to sample?

how to choose?

- try to connect q_{new} to q_{end}
  if successful → done!