PhD Public Defense

On the Complexity of Optimization: Curved Spaces and Benign Landscapes

Candidate: Chris Criscitiello

Advisor: Nicolas Boumal

EPFL

OPTIM, Chair of Continuous Optimization

September 12, 2025

Outline

What is optimization?

Optimization algorithm?

Steepest descent

Best possible algorithm?

Outline

What is optimization?

Optimization algorithm?

Steepest descent

Best possible algorithm?

Based on:

- "Negative curvature obstructs acceleration for strongly geodesically convex optimization” - C & **Boumal** – COLT’22
- “Curvature and Complexity: Better lower bounds for geodesically convex optimization” - C & **Boumal** – COLT’23
- “Synchronization on circles and spheres with nonlinear interactions” - C, **Rebjock, McRae, Boumal** - under review
- “The sensor network localization problem has benign landscape under mild rank relaxation” - C, **Rebjock, McRae, Boumal** – under review

What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

$$\min_{x \in M} f(x)$$

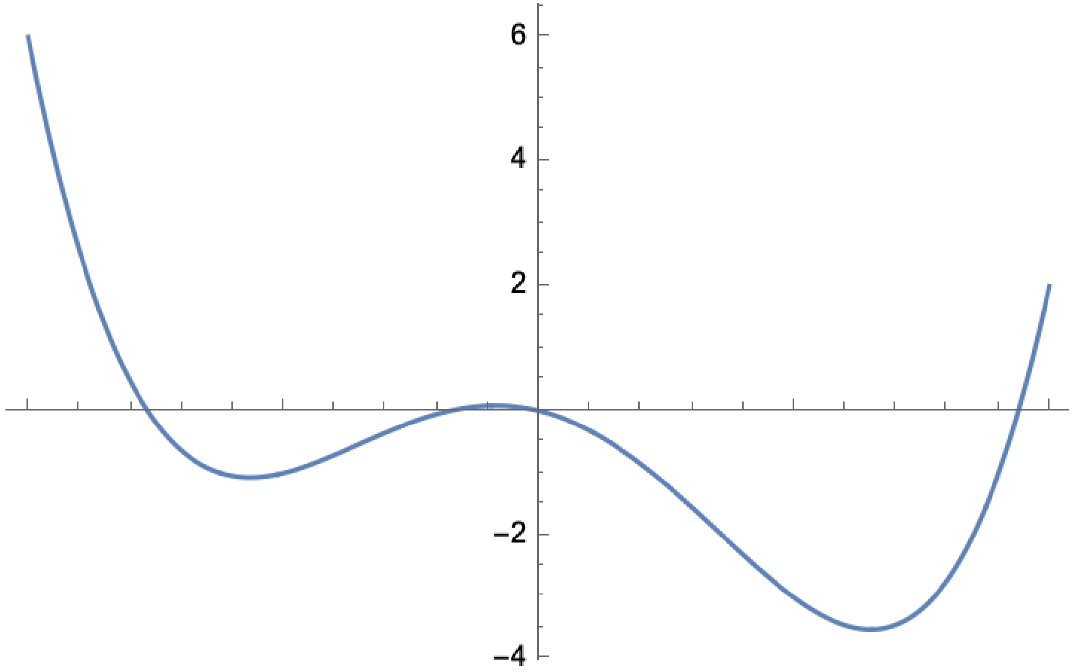
What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

$$\min_{x \in M} f(x)$$

Examples:

$f(x) = x^4 - 3x^2 - x$
$M = \{\text{numbers}\}$



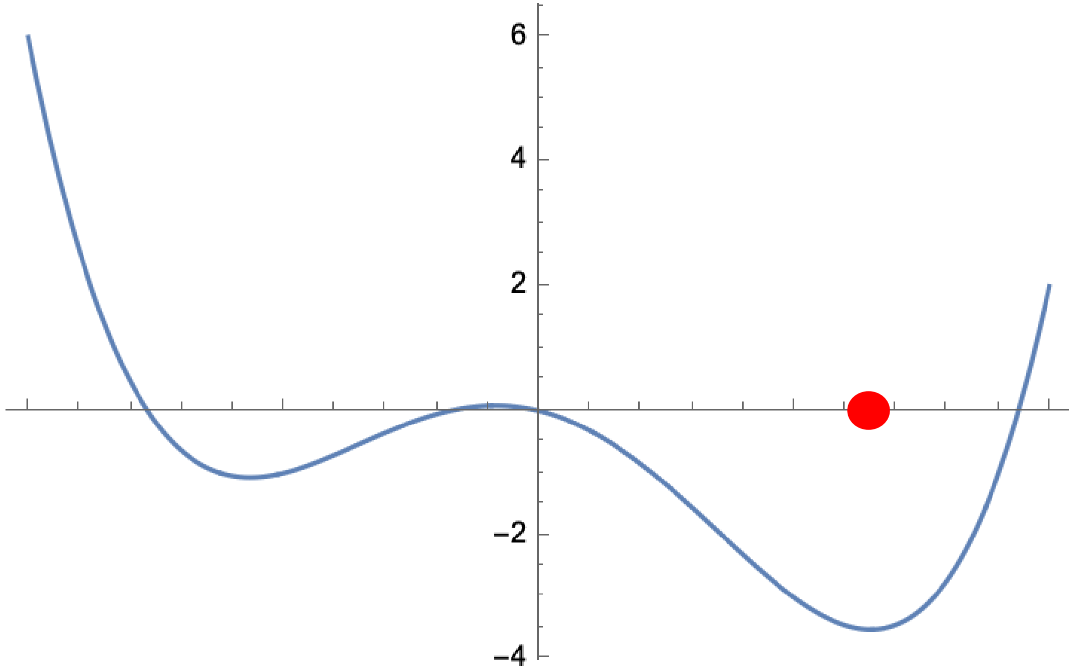
What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

$$\min_{x \in M} f(x)$$

Examples:

$f(x) = x^4 - 3x^2 - x$
$M = \{\text{numbers}\}$



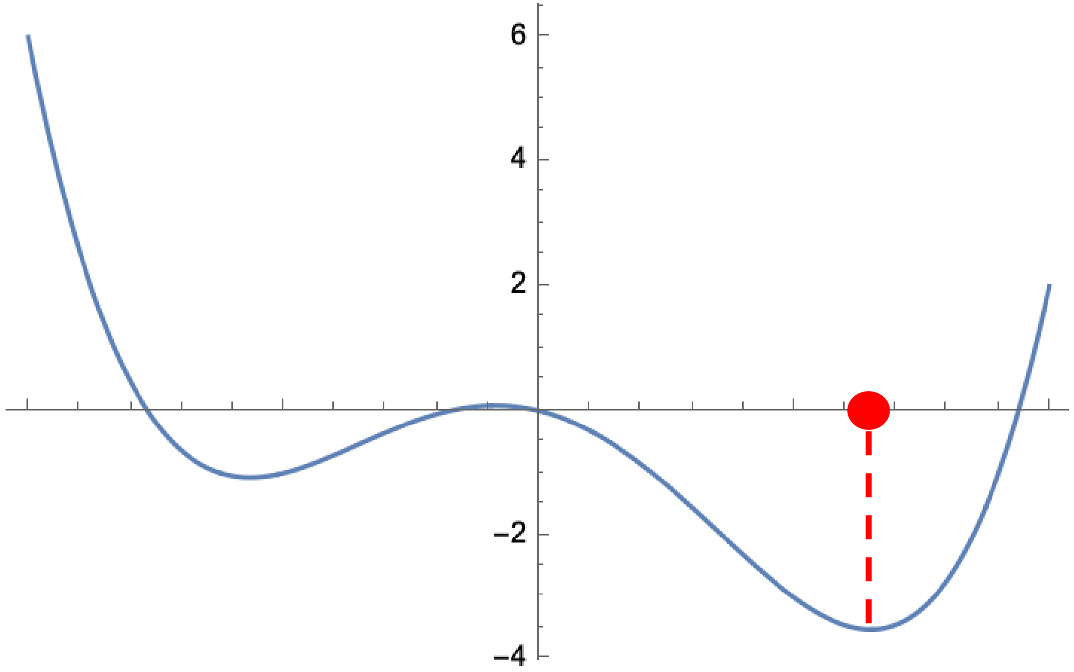
What is optimization?

Find minimum of function f over M = {set of possibilities}

$$\min_{x \in M} f(x)$$

Examples:

$f(x) = x^4 - 3x^2 - x$
M = {numbers}

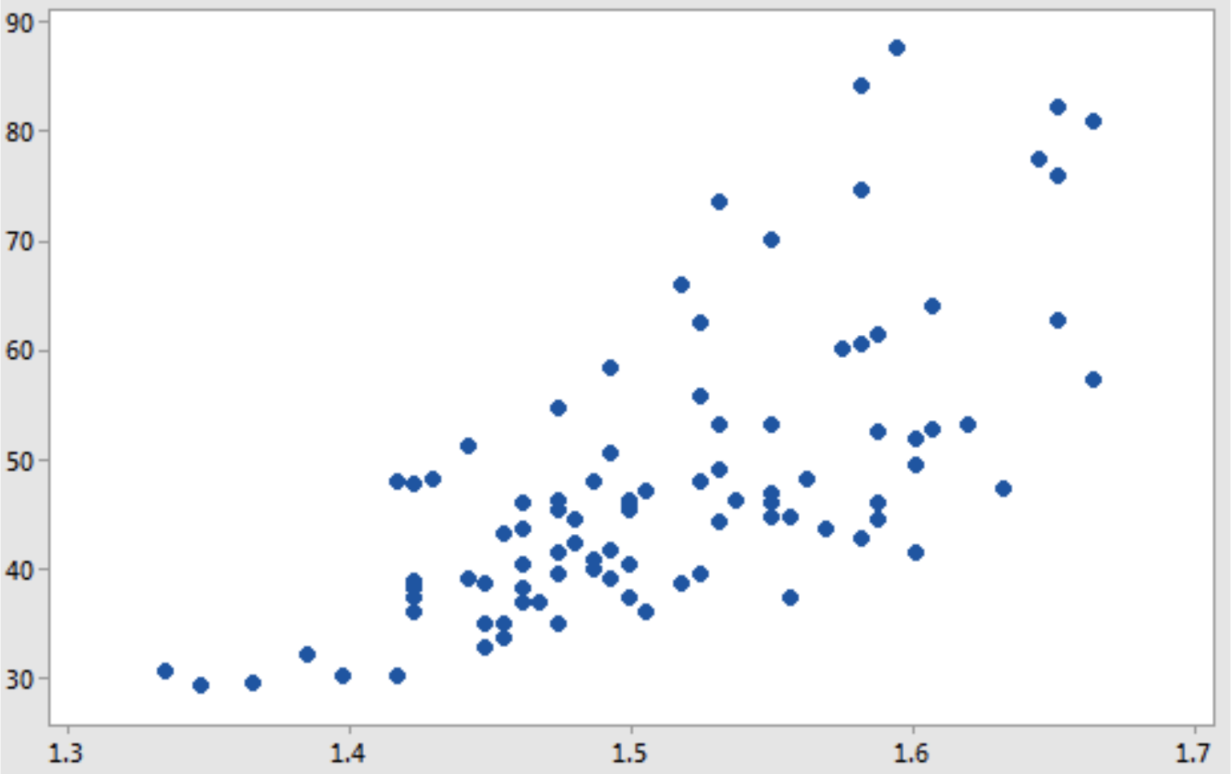


What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

$$\min_{x \in M} f(x)$$

Examples:



What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

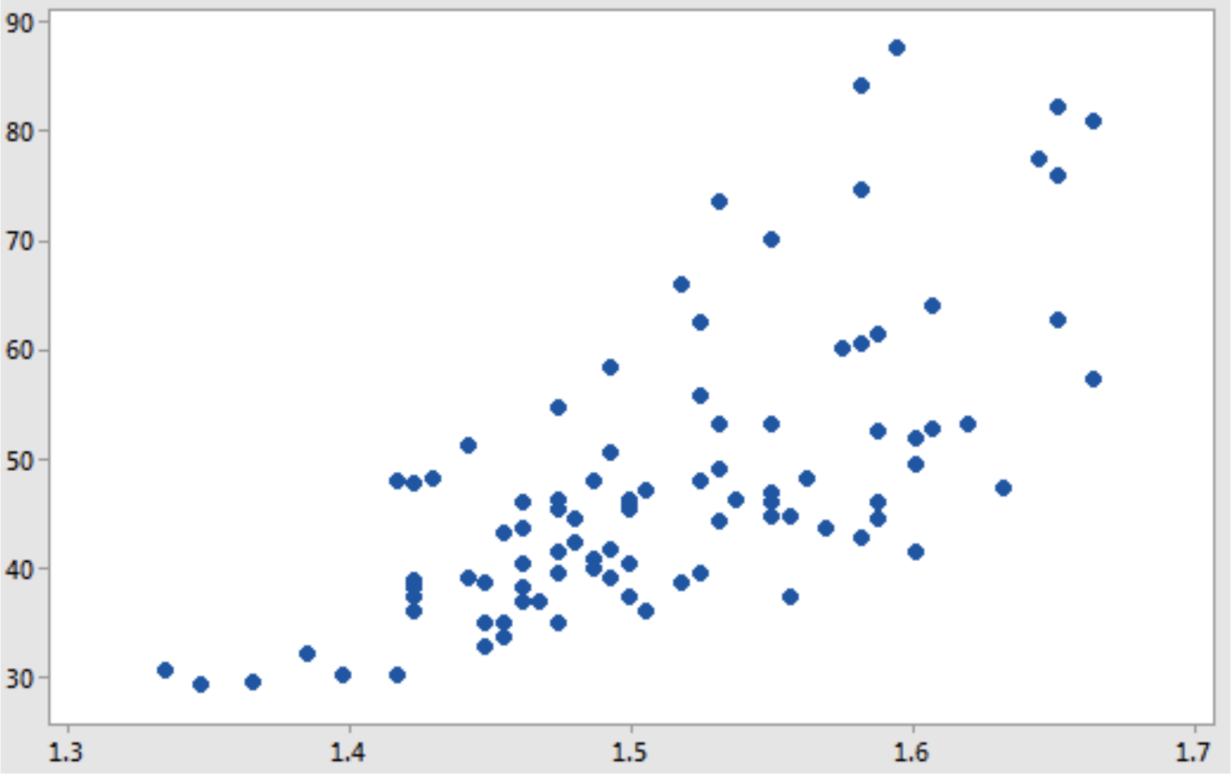
$$\min_{x \in M} f(x)$$

Examples:

$M = \{\text{all lines}\}$

$f(\text{line}) =$ distance between points and line

$$f(\beta) = \|X\beta - y\|^2$$



What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

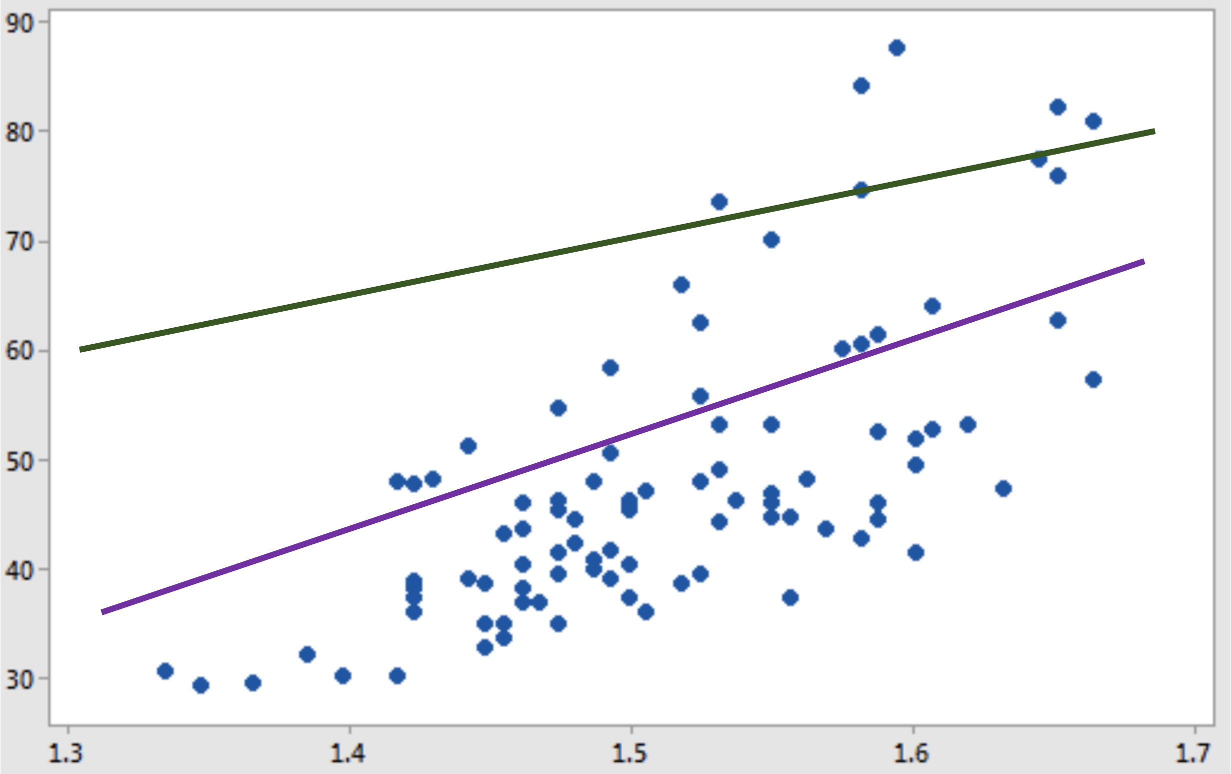
$$\min_{x \in M} f(x)$$

Examples:

$M = \{\text{all lines}\}$

$f(\text{line}) = \text{distance between points and line}$

$f(\beta) = \|X\beta - y\|^2$



What is optimization?

Find minimum of function f over $M = \{\text{set of possibilities}\}$

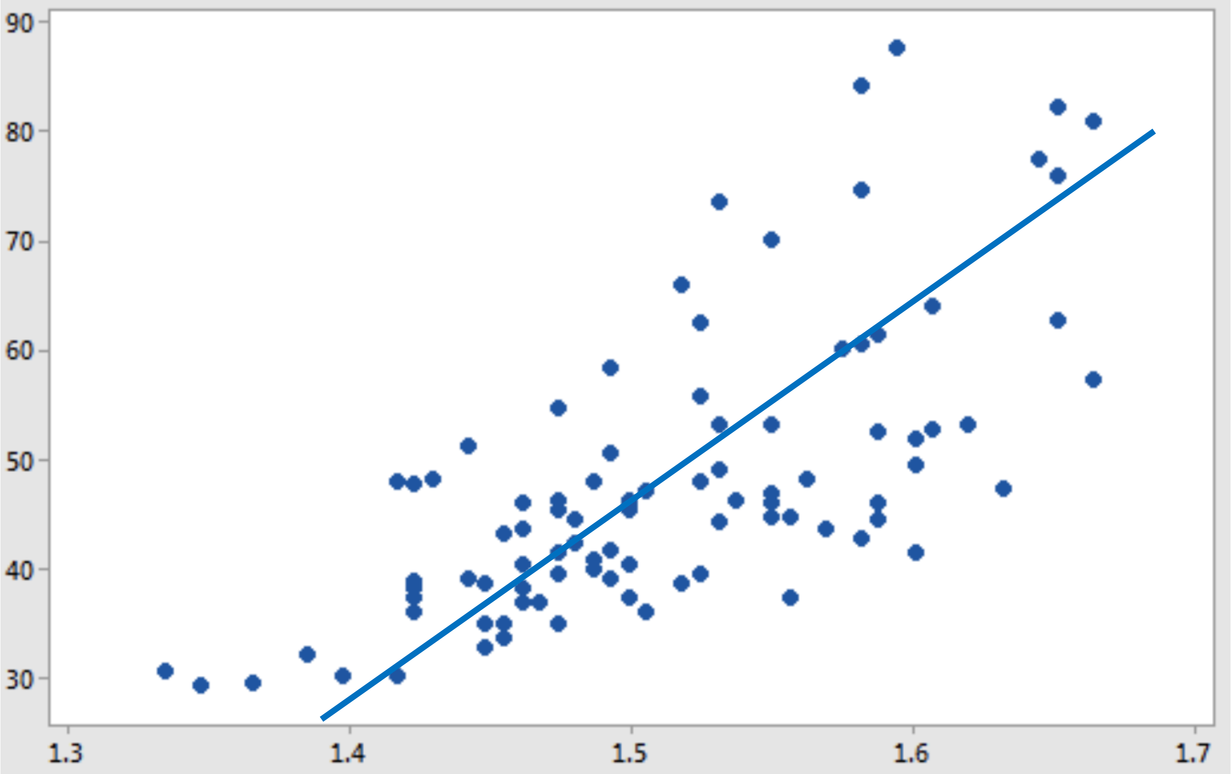
$$\min_{x \in M} f(x)$$

Examples:

$M = \{\text{all lines}\}$

$f(\text{line}) =$ distance between points and line

$$f(\beta) = \|X\beta - y\|^2$$



What is optimization?

Find minimum of function f over M = {set of possibilities}

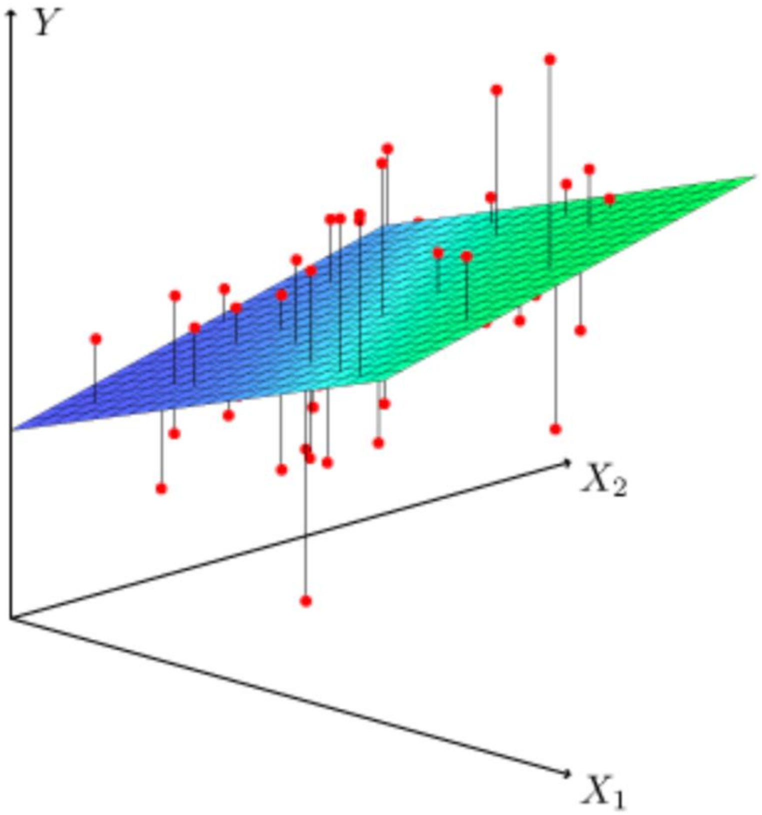
$$\min_{x \in M} f(x)$$

Examples:

Higher dimensional variants

Harder, can't visualize!

Easily encounter **millions** of features



What is optimization?

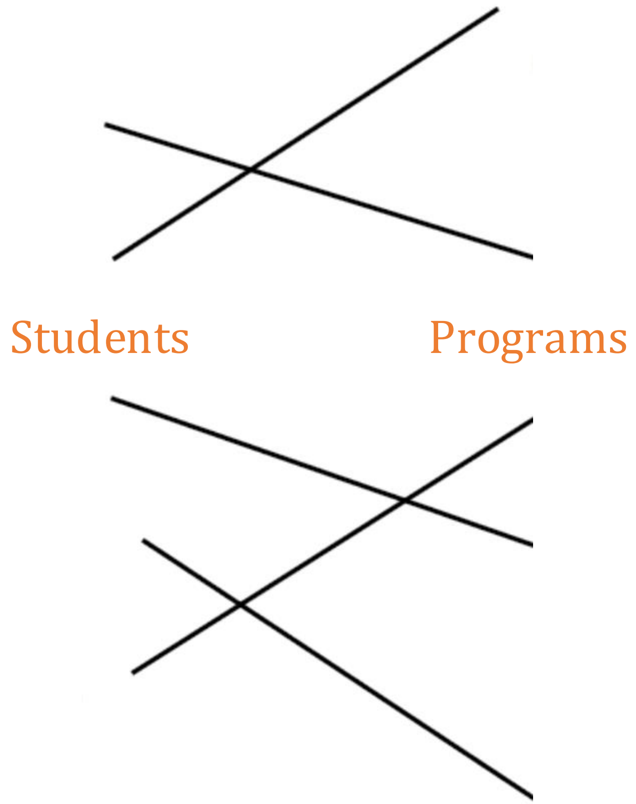
Find minimum of function f over $M = \{\text{set of possibilities}\}$

$$\min_{x \in M} f(x)$$

Examples:

How to find *best* matching of medical students to residency programs?

$M = \{\text{all possible matchings}\}$



What is an optimization algorithm?

Algorithm = method to solve an optim problem

What is an optimization algorithm?

Algorithm = method to solve an optim problem

Imagine super-complicated function f stored on a computer/server

What is an optimization algorithm?

Algorithm = method to solve an optim problem

Imagine super-complicated function f stored on a computer/server

What can you do to minimize f?

What is an optimization algorithm?

Algorithm = method to solve an optim problem

Imagine super-complicated function f stored on a computer/server

What can you do to minimize f?

- You can evaluate f on an input x, and get $f(x)$

$x \longrightarrow$  $\longrightarrow f(x)$

What is an optimization algorithm?

Algorithm = method to solve an optim problem

Imagine super-complicated function f stored on a computer/server

What can you do to minimize f?

- You can evaluate f on an input x, and get $f(x)$
- You can evaluate f on an input x, and get "gradient" $\nabla f(x)$

x  $f(x), \nabla f(x)$

What is an optimization algorithm?

Algorithm = method to solve an optim problem

Imagine super-complicated function f stored on a computer/server

What can you do to minimize f?

- You can evaluate f on an input x, and get $f(x)$
- You can evaluate f on an input x, and get “gradient” $\nabla f(x)$

$\boxed{\text{query}}$ $\quad x \longrightarrow$

$\longrightarrow \quad f(x), \nabla f(x)$

What is an optimization algorithm?

Algorithm = a method to choose queries $x_0, x_1, x_2, \ldots$

$x_0 \longrightarrow$  $\longrightarrow f(x_0), \nabla f(x_0)$

What is an optimization algorithm?

Algorithm = a method to choose queries $x_0, x_1, x_2, \ldots$

$x_0 \longrightarrow$  $\longrightarrow f(x_0), \nabla f(x_0)$

Algo $\longrightarrow x_1$

What is an optimization algorithm?

Algorithm = a method to choose queries $x_0, x_1, x_2, \ldots$

$x_0 \longrightarrow$

$\longrightarrow f(x_0), \nabla f(x_0)$

Algo $\longrightarrow x_1 \longrightarrow$

$\longrightarrow f(x_1), \nabla f(x_1)$

What is an optimization algorithm?

Algorithm = a method to choose queries $x_0, x_1, x_2, \ldots$

$x_0 \longrightarrow$  $\longrightarrow f(x_0), \nabla f(x_0)$

$x_1 \longrightarrow$  $\longrightarrow f(x_1), \nabla f(x_1)$

Algo $\longrightarrow x_2 \longrightarrow$  $\longrightarrow f(x_2), \nabla f(x_2)$

What is an optimization algorithm?

Algorithm = a method to choose queries $x_0, x_1, x_2, \ldots$

$x_0 \longrightarrow$

$\longrightarrow f(x_0), \nabla f(x_0)$

$x_1 \longrightarrow$

$\longrightarrow f(x_1), \nabla f(x_1)$

$x_2 \longrightarrow$

$\longrightarrow f(x_2), \nabla f(x_2)$

$\cdots$

x_T so that $f(x_t)$ is small

Steepest descent

gradient $\nabla f(x)$ = direction of steepest descent

Steepest descent

gradient $\nabla f(x)$ = direction of steepest descent

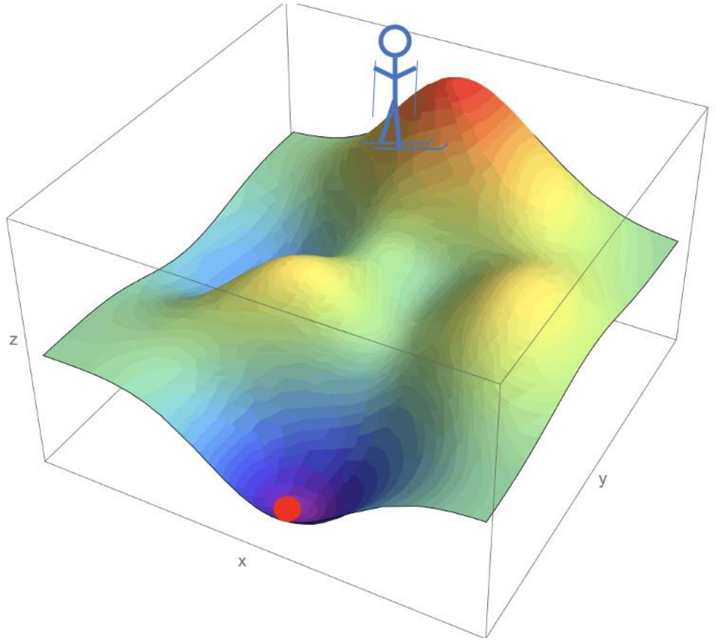
Imagine you are skiing in a blizzard, or hiking in a dense forest

Steepest descent

gradient $\nabla f(x)$ = direction of steepest descent

Imagine you are skiing in a blizzard, or hiking in a dense forest

You want to minimize your elevation as quickly as possible



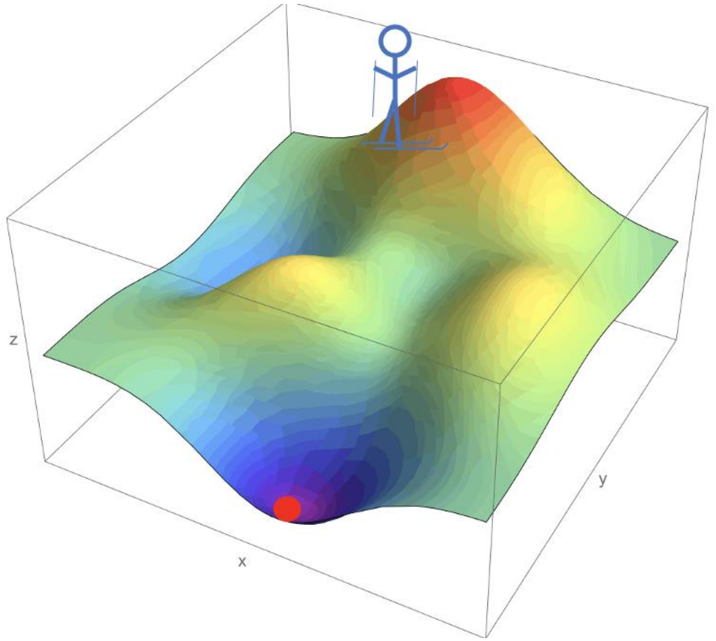
Steepest descent

gradient $\nabla f(x)$ = direction of steepest descent

Imagine you are skiing in a blizzard, or hiking in a dense forest

You want to minimize your elevation as quickly as possible

Simple idea: follow direction of steepest descent!



Steepest descent

gradient $\nabla f(x)$ = direction of steepest descent

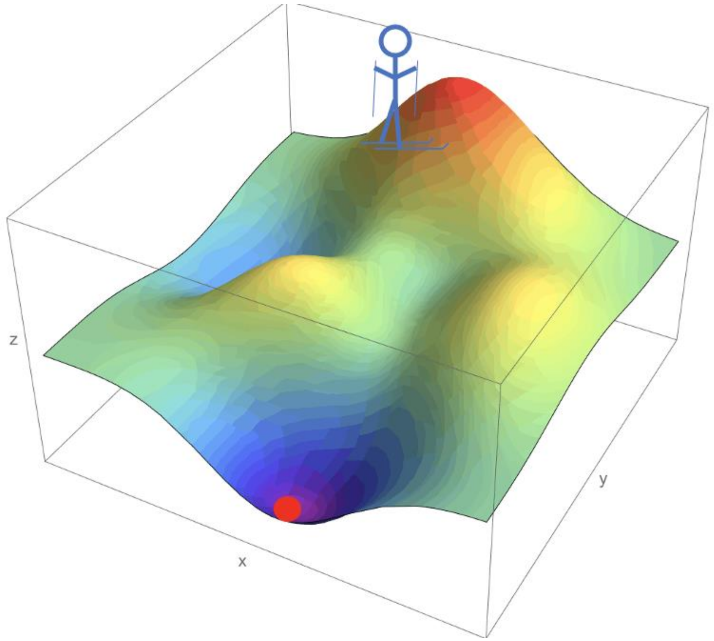
Imagine you are skiing in a blizzard, or hiking in a dense forest

You want to minimize your elevation as quickly as possible

Simple idea: follow direction of steepest descent!

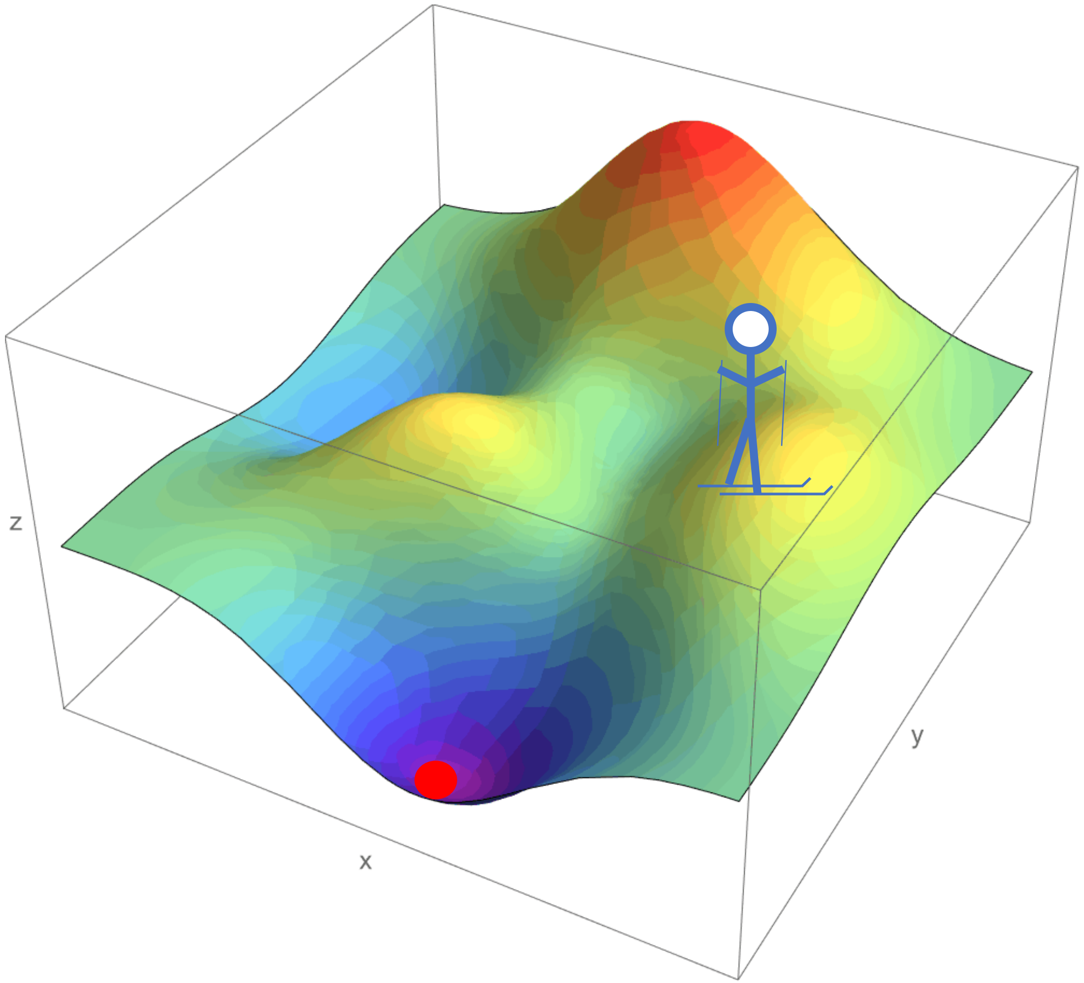
Algorithm = **Steepest Descent**

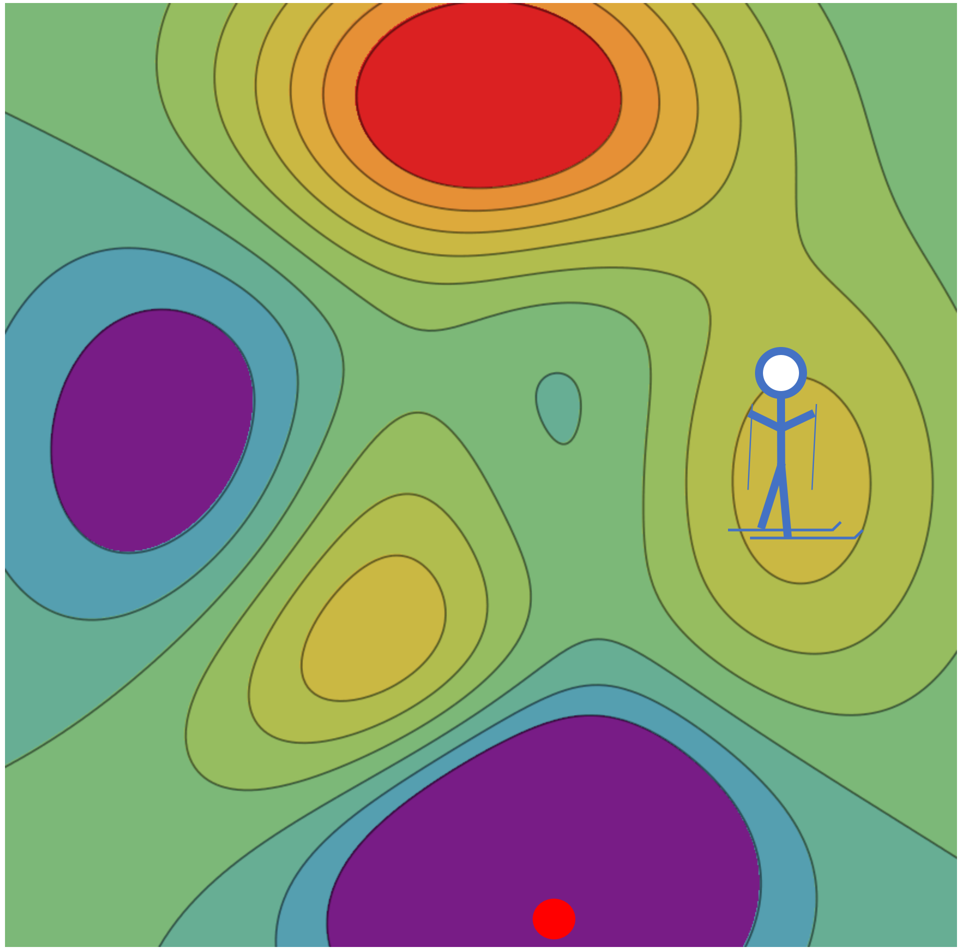
$$x_{k+1} = x_k - \eta \nabla f(x_k)$$



Steepest Descent

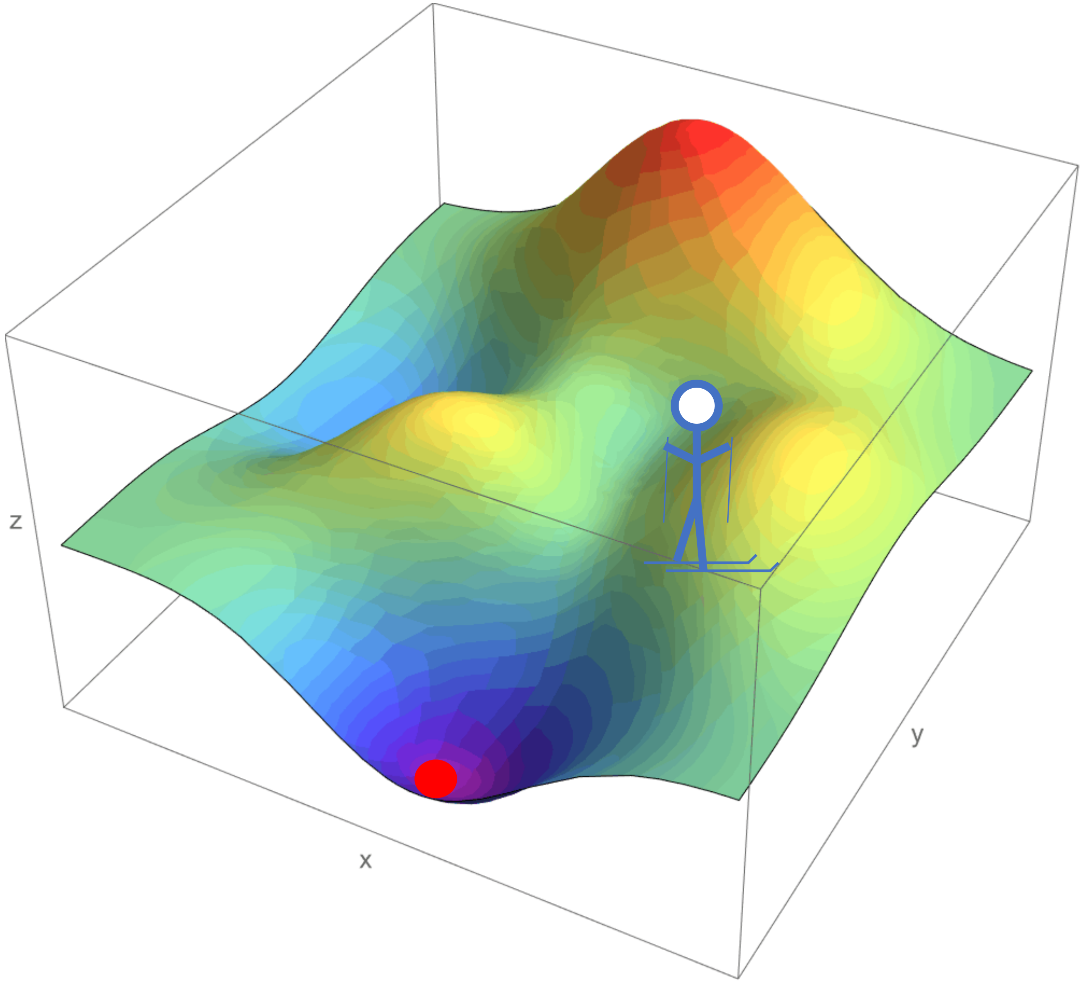
Simple idea: follow direction of steepest descent!

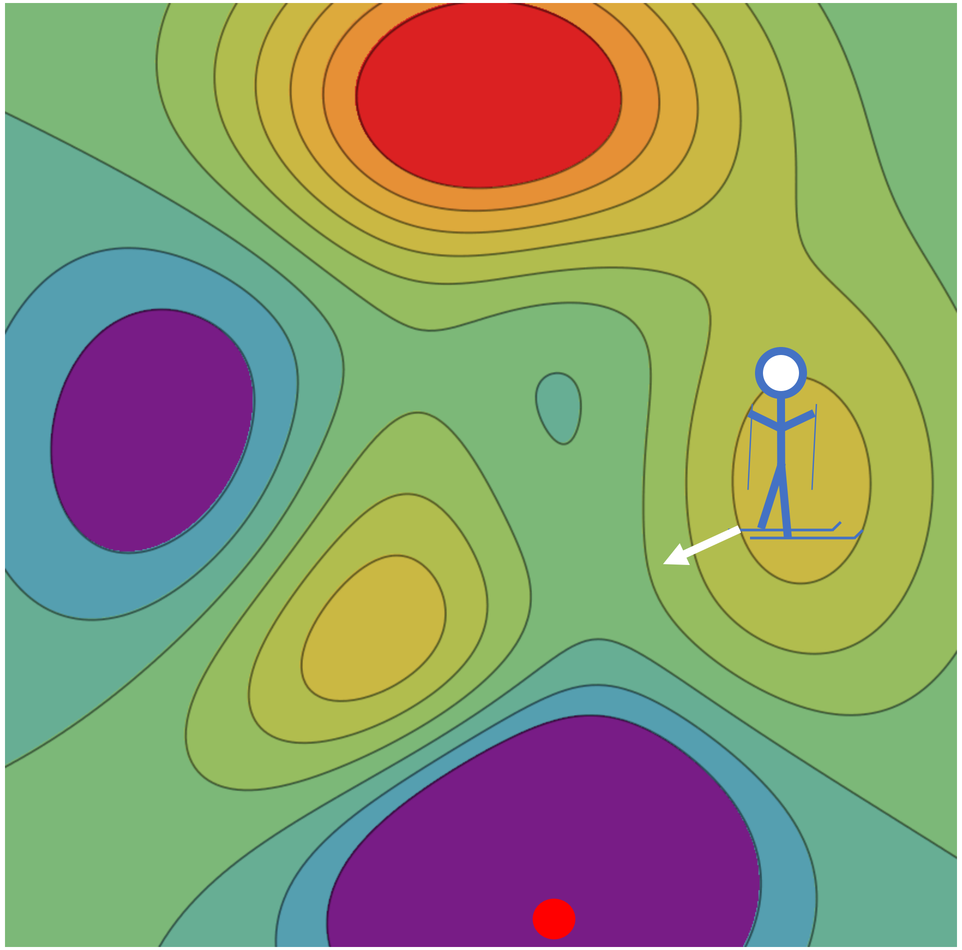




Steepest Descent

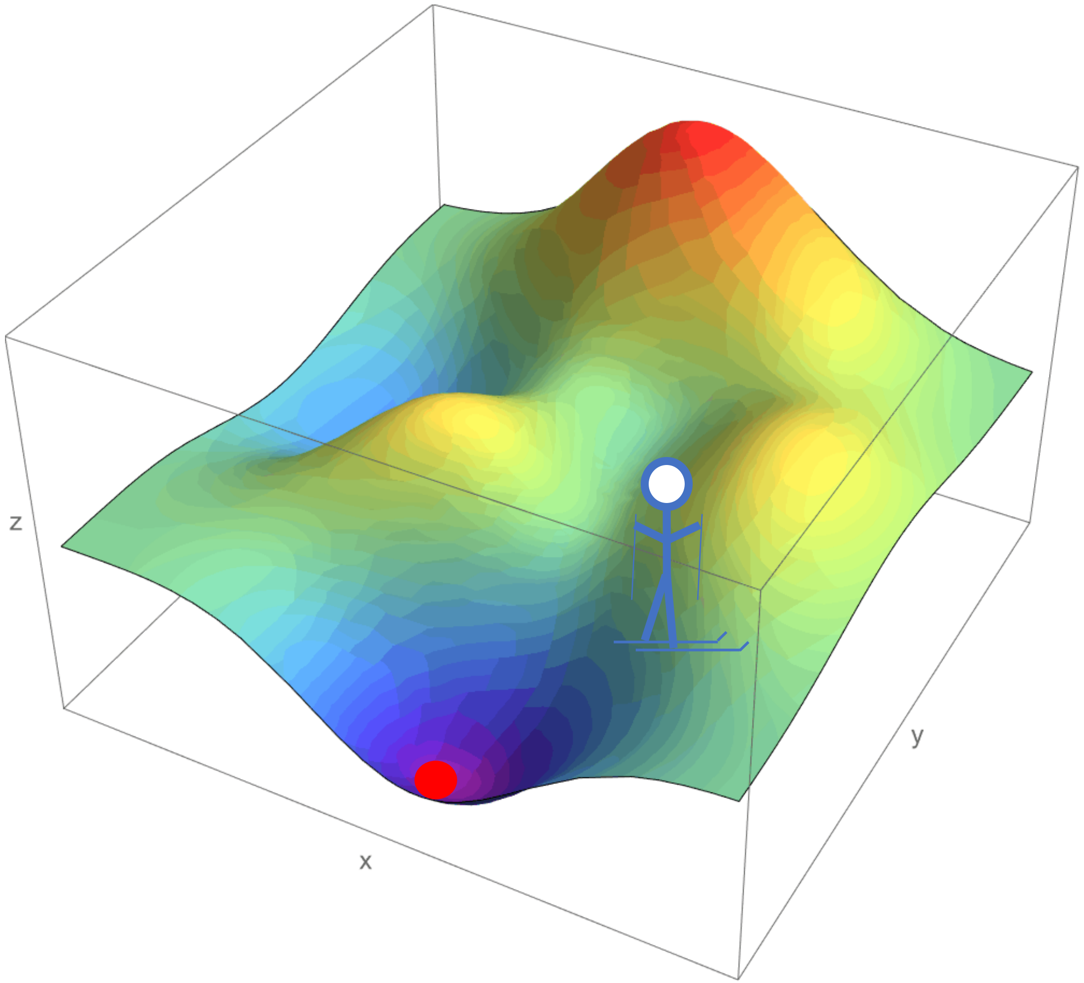
Simple idea: follow direction of steepest descent!

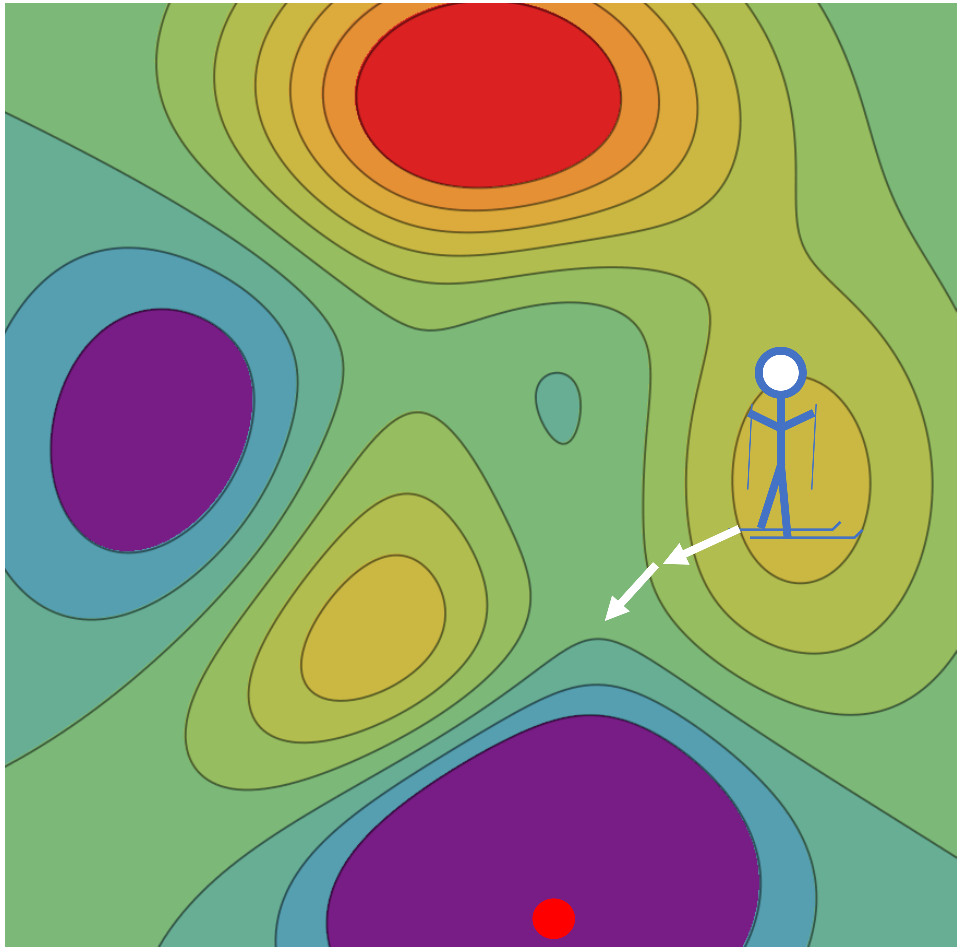




Steepest Descent

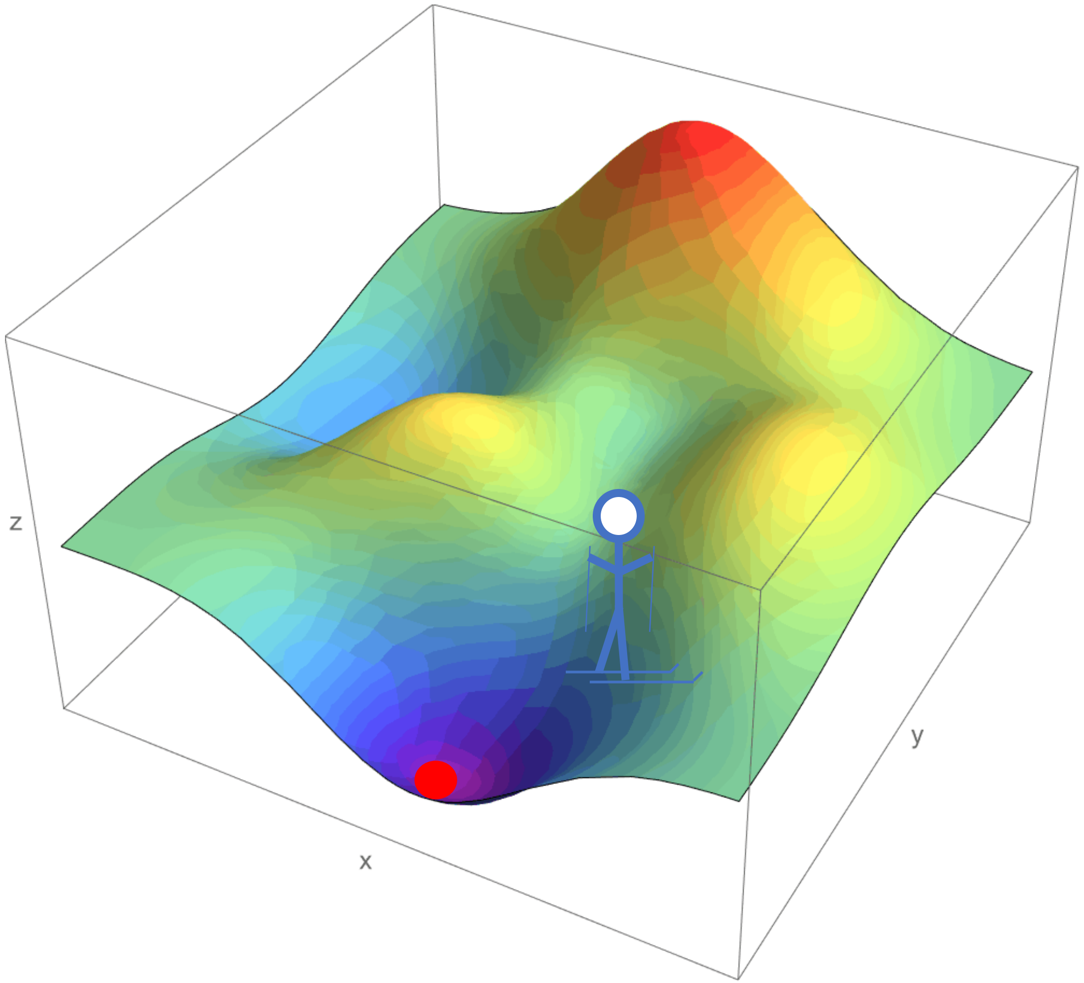
Simple idea: follow direction of steepest descent!

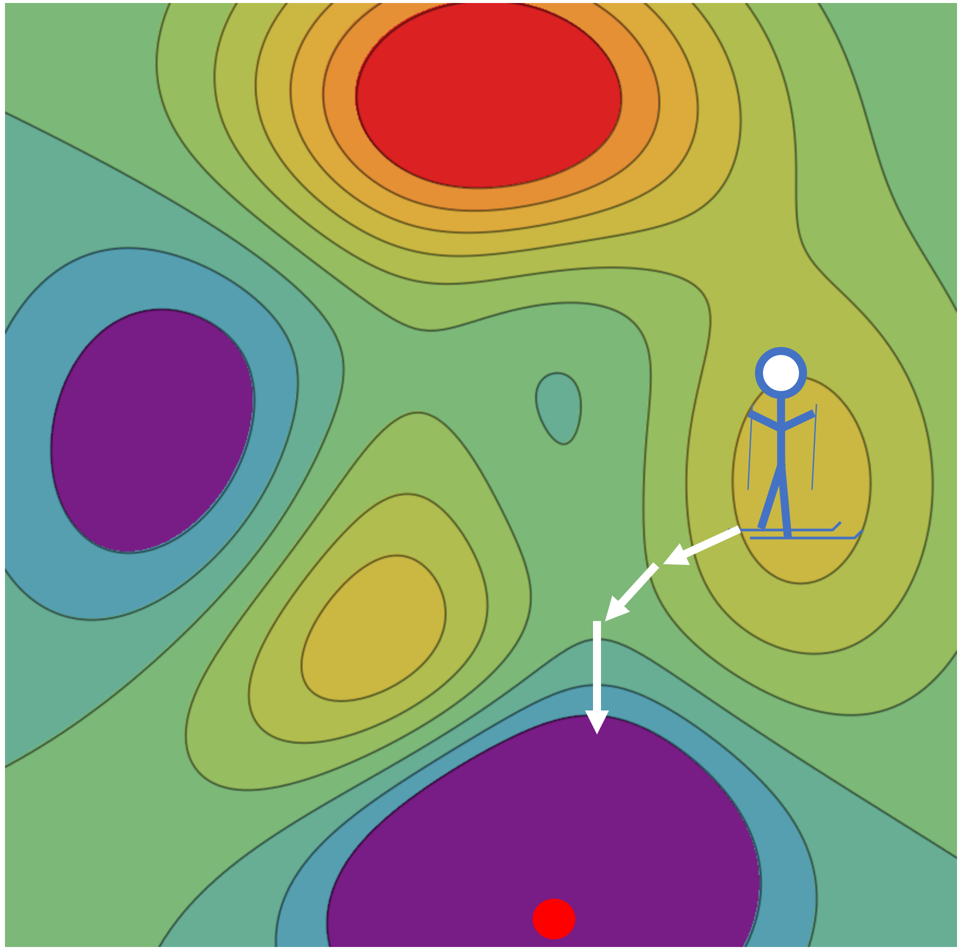




Steepest Descent

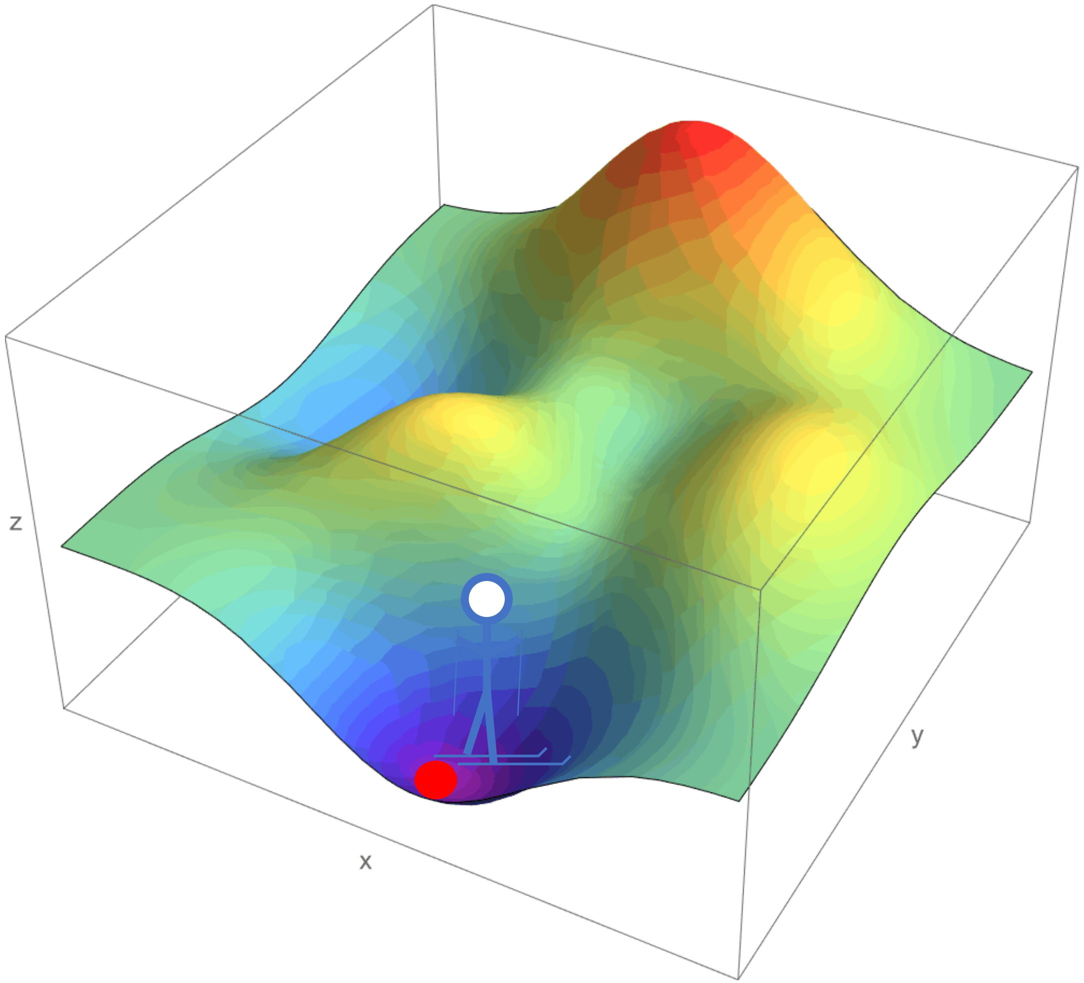
Simple idea: follow direction of steepest descent!

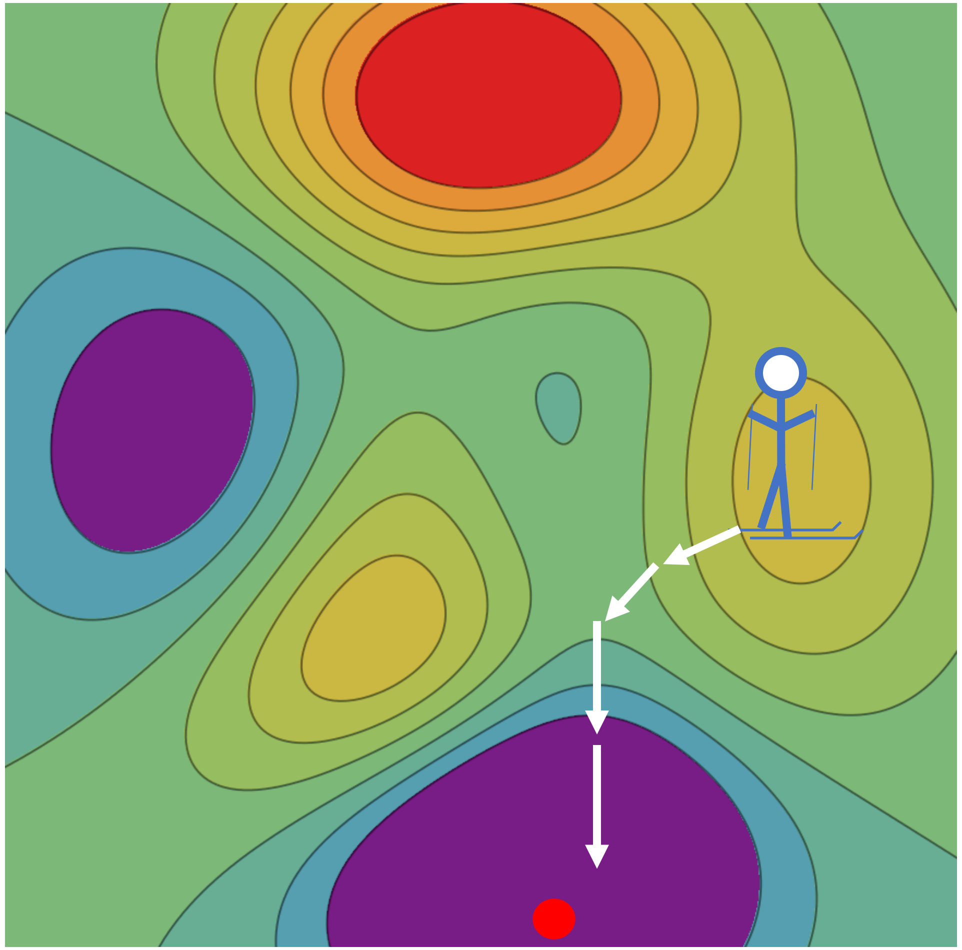




Steepest Descent

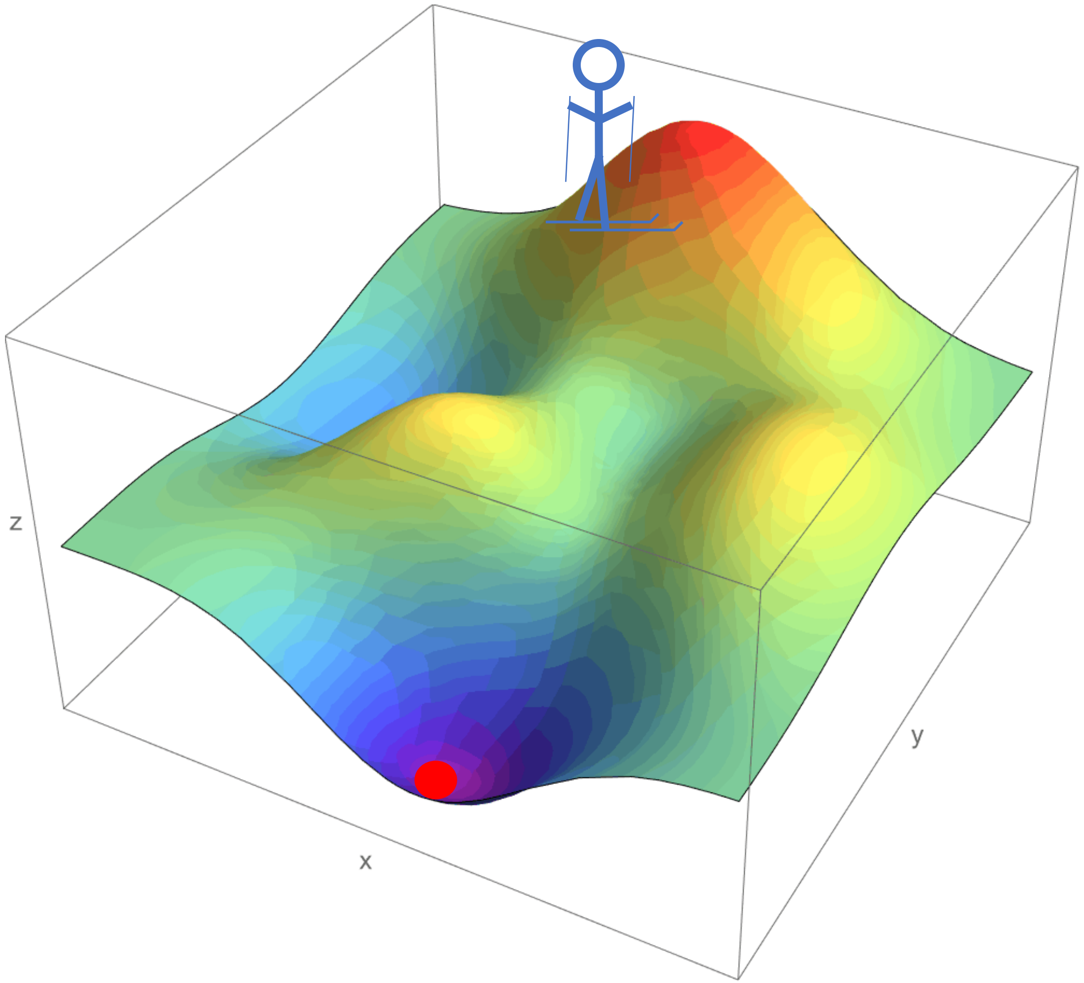
Simple idea: follow direction of steepest descent!

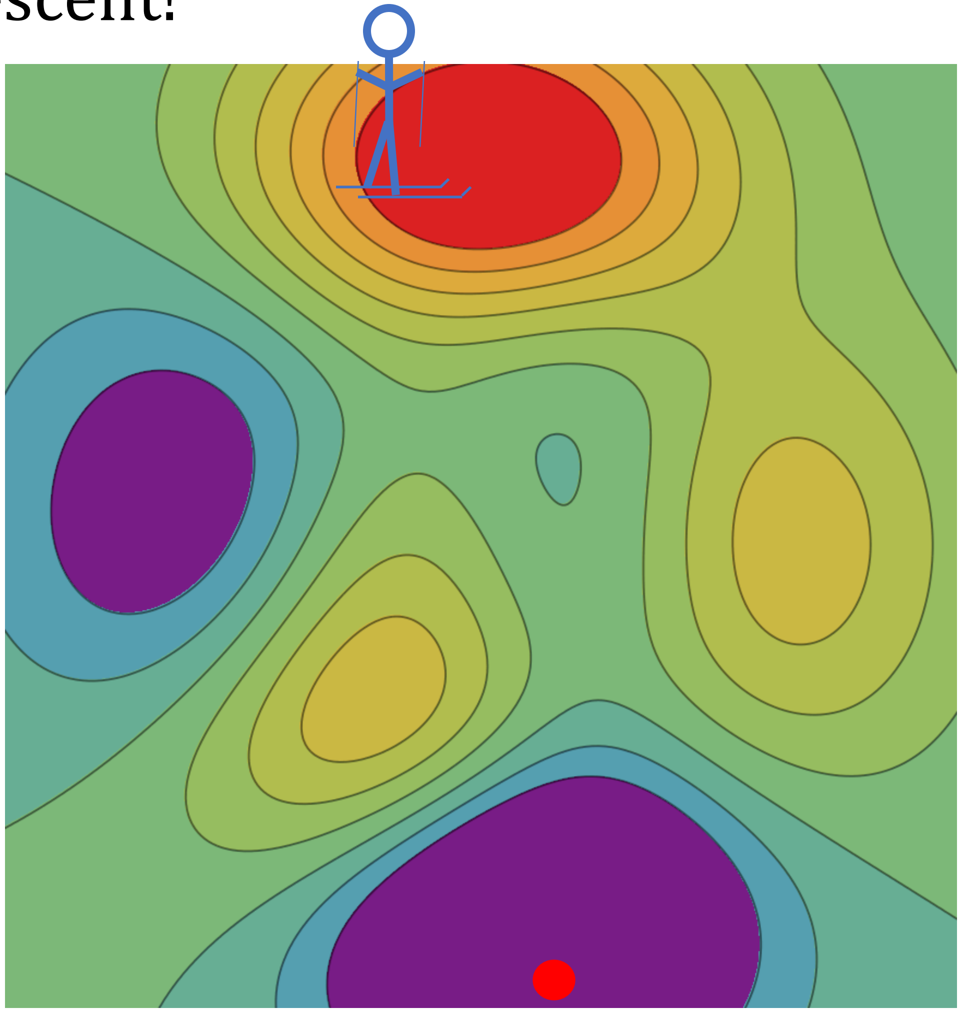




Steepest Descent

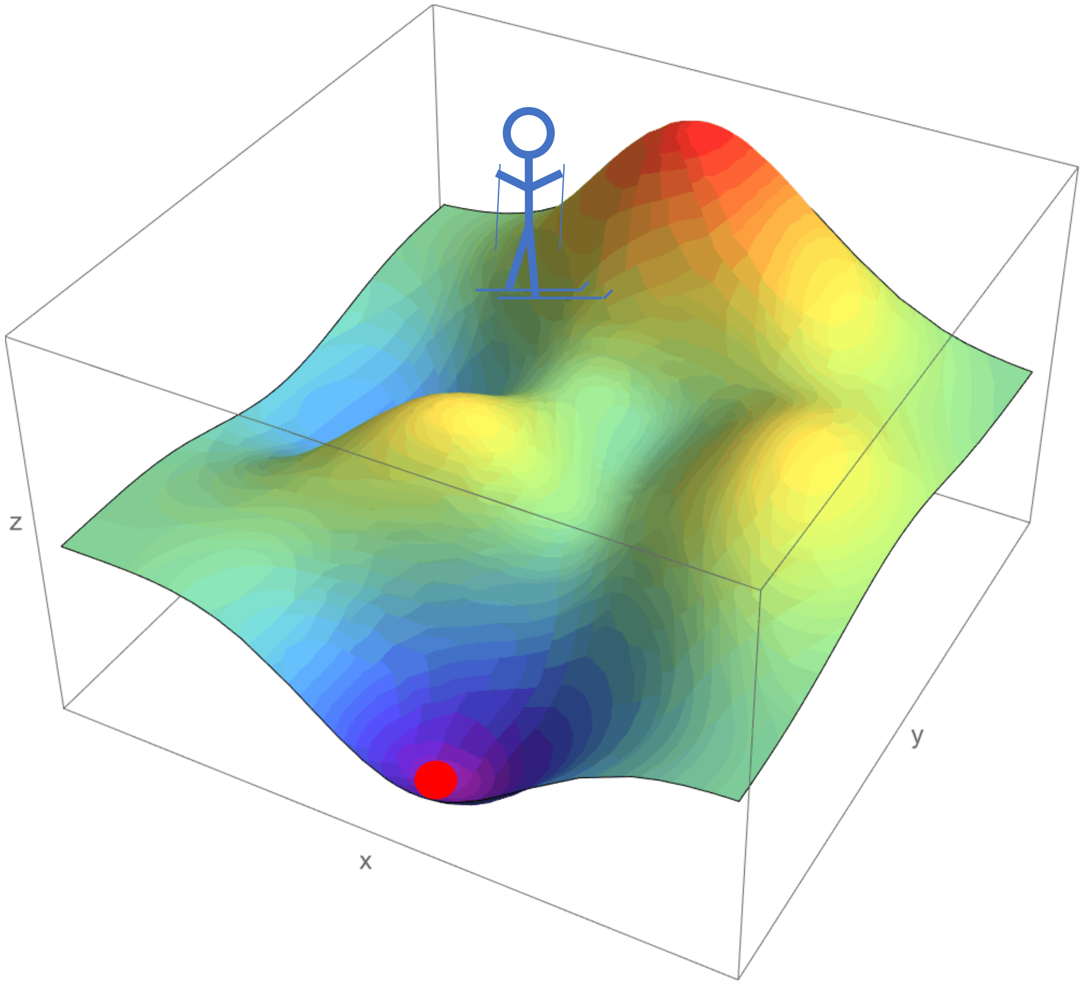
Simple idea: follow direction of steepest descent!

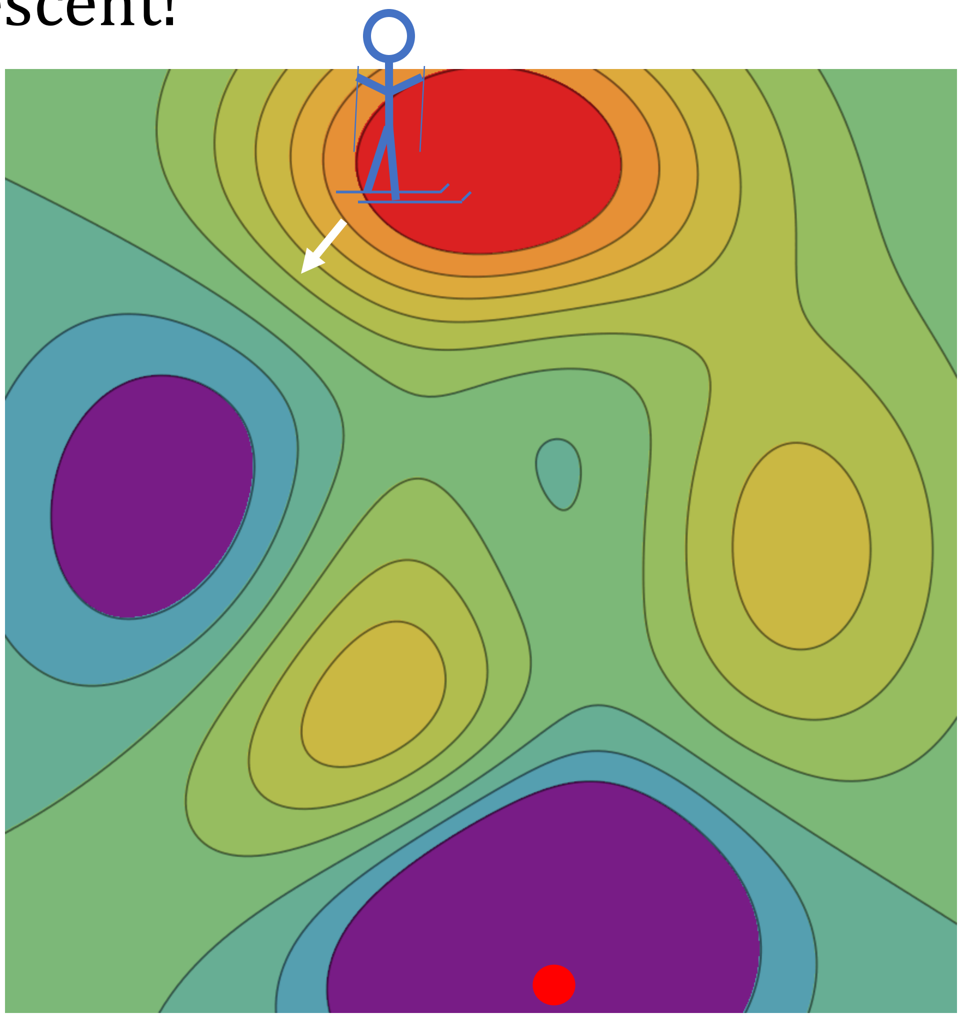




Steepest Descent

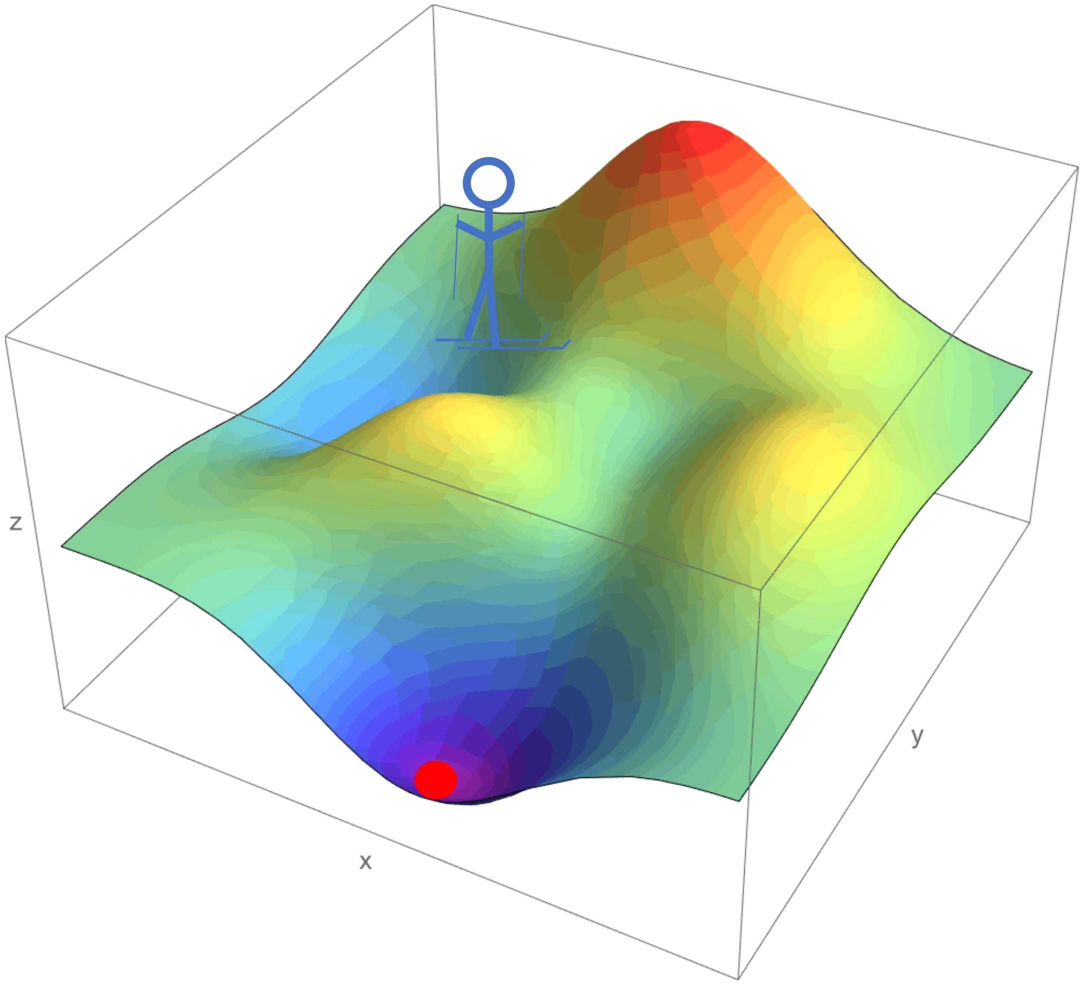
Simple idea: follow direction of steepest descent!

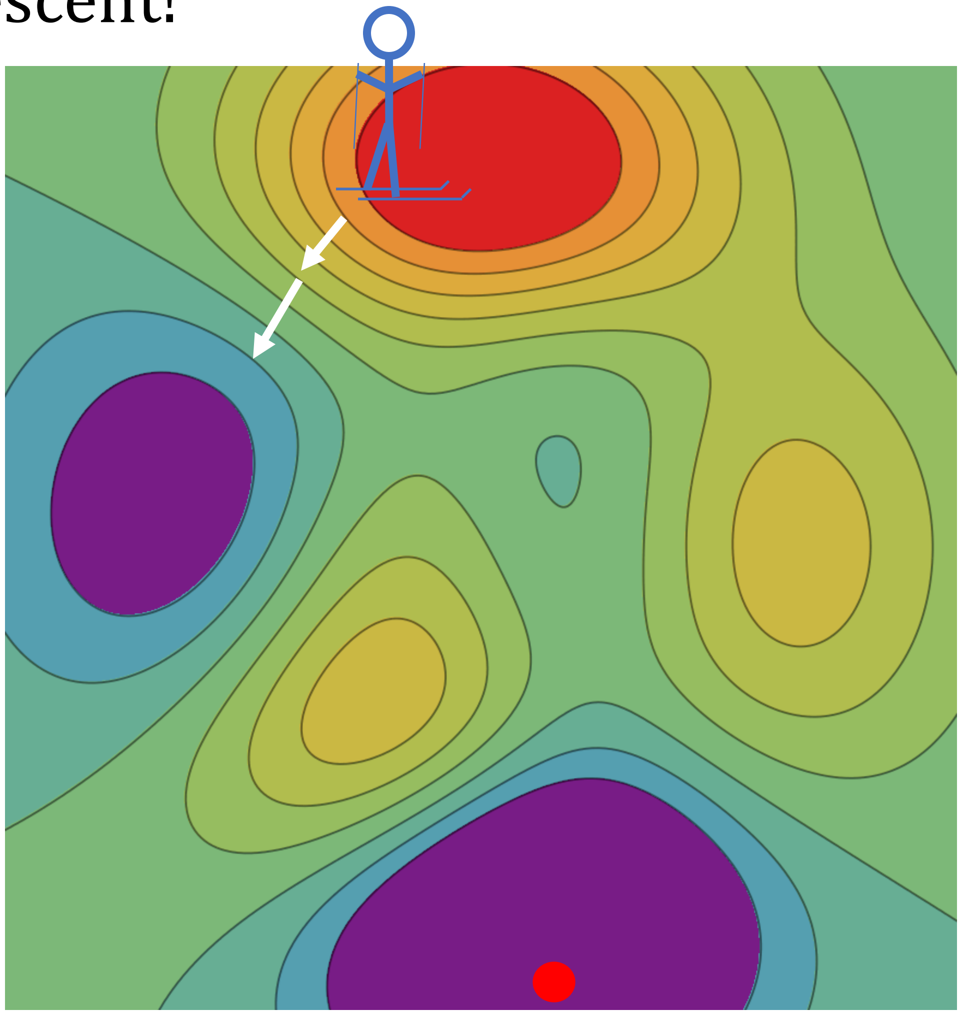




Steepest Descent

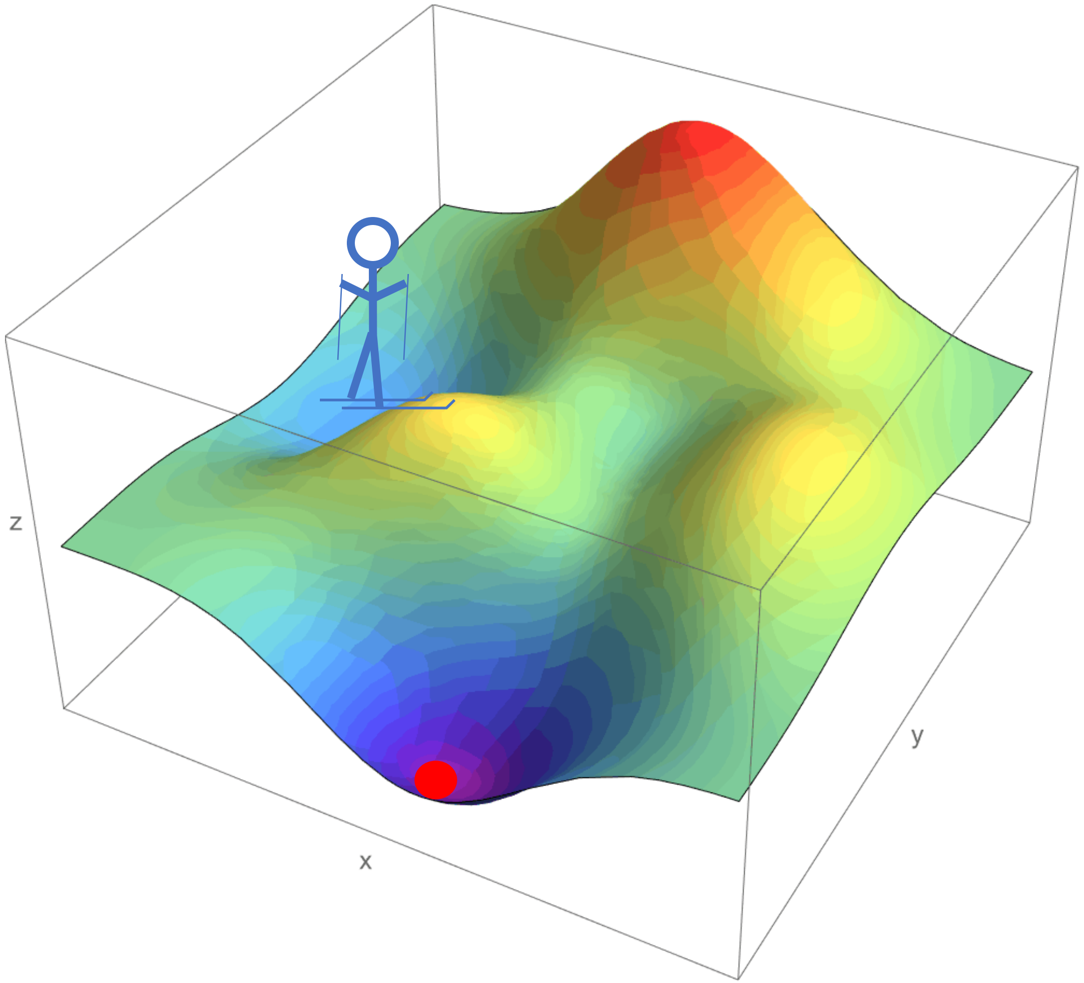
Simple idea: follow direction of steepest descent!

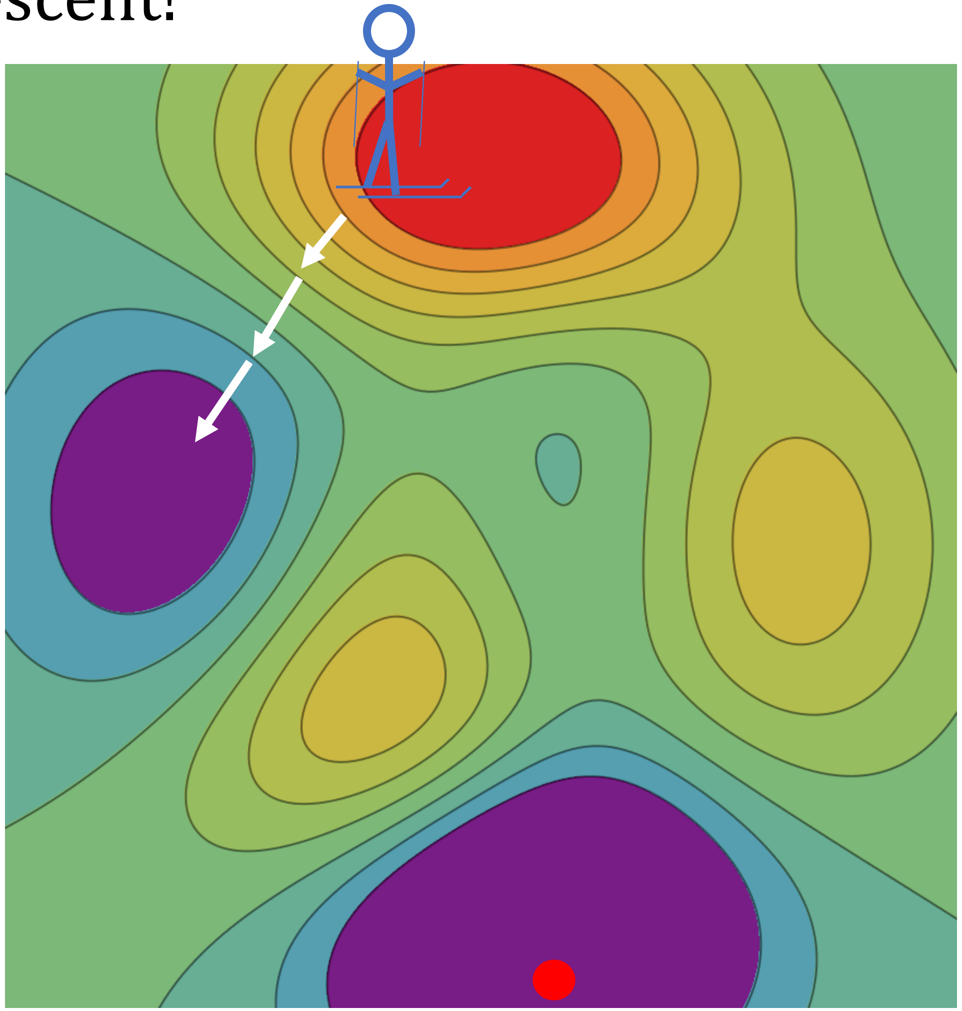




Steepest Descent

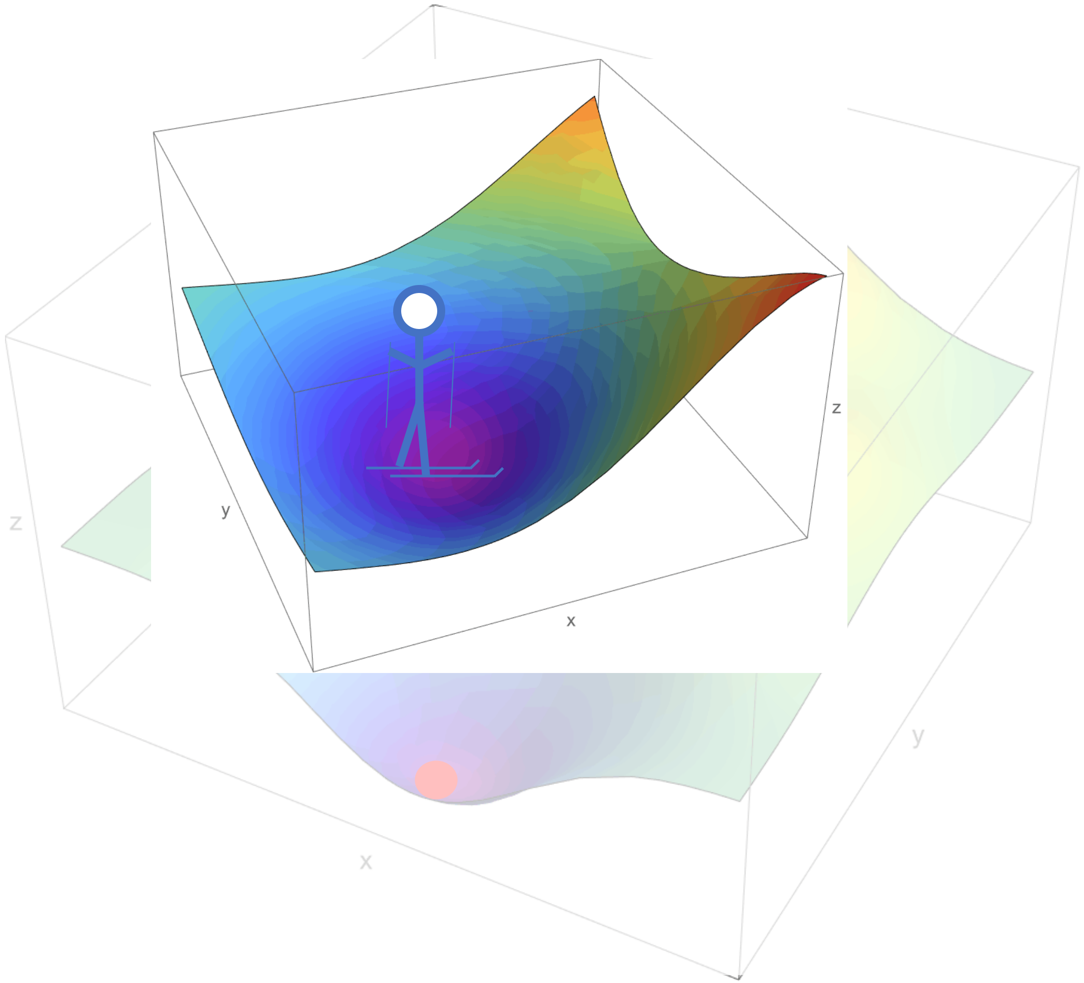
Simple idea: follow direction of steepest descent!

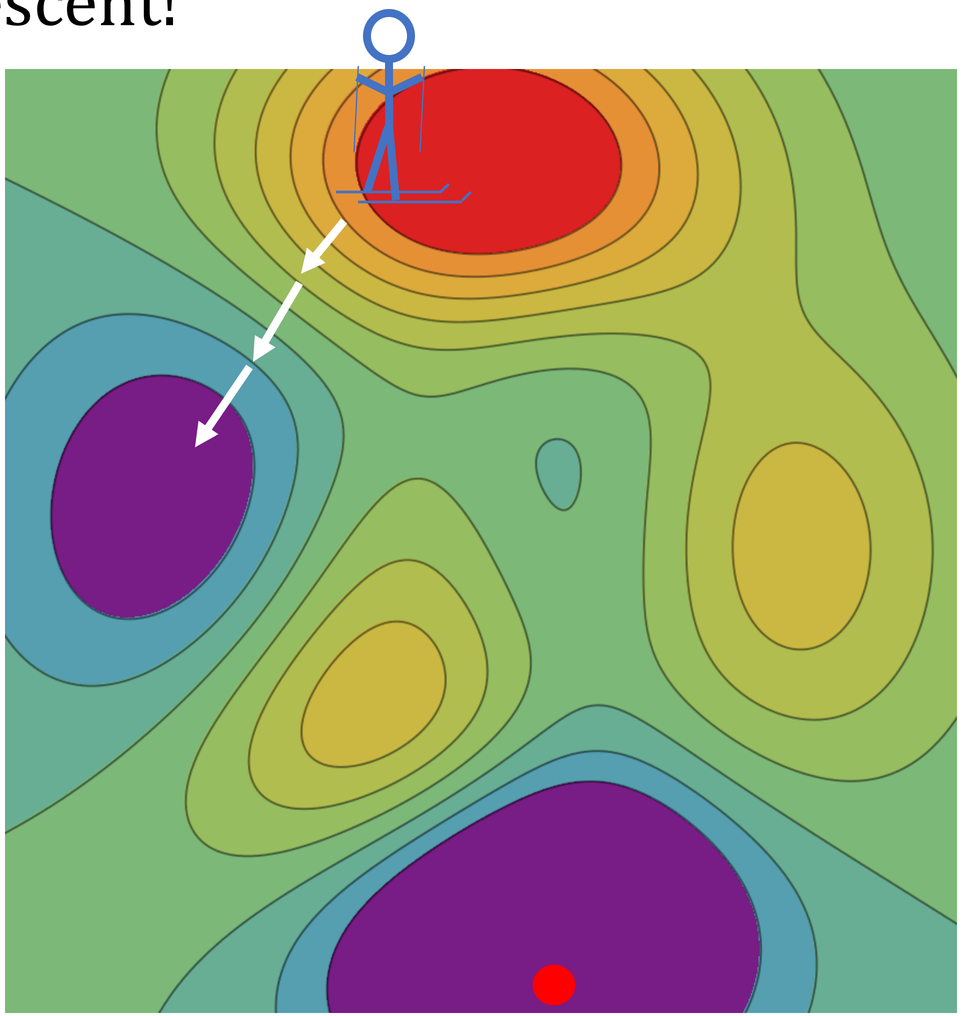




Steepest Descent

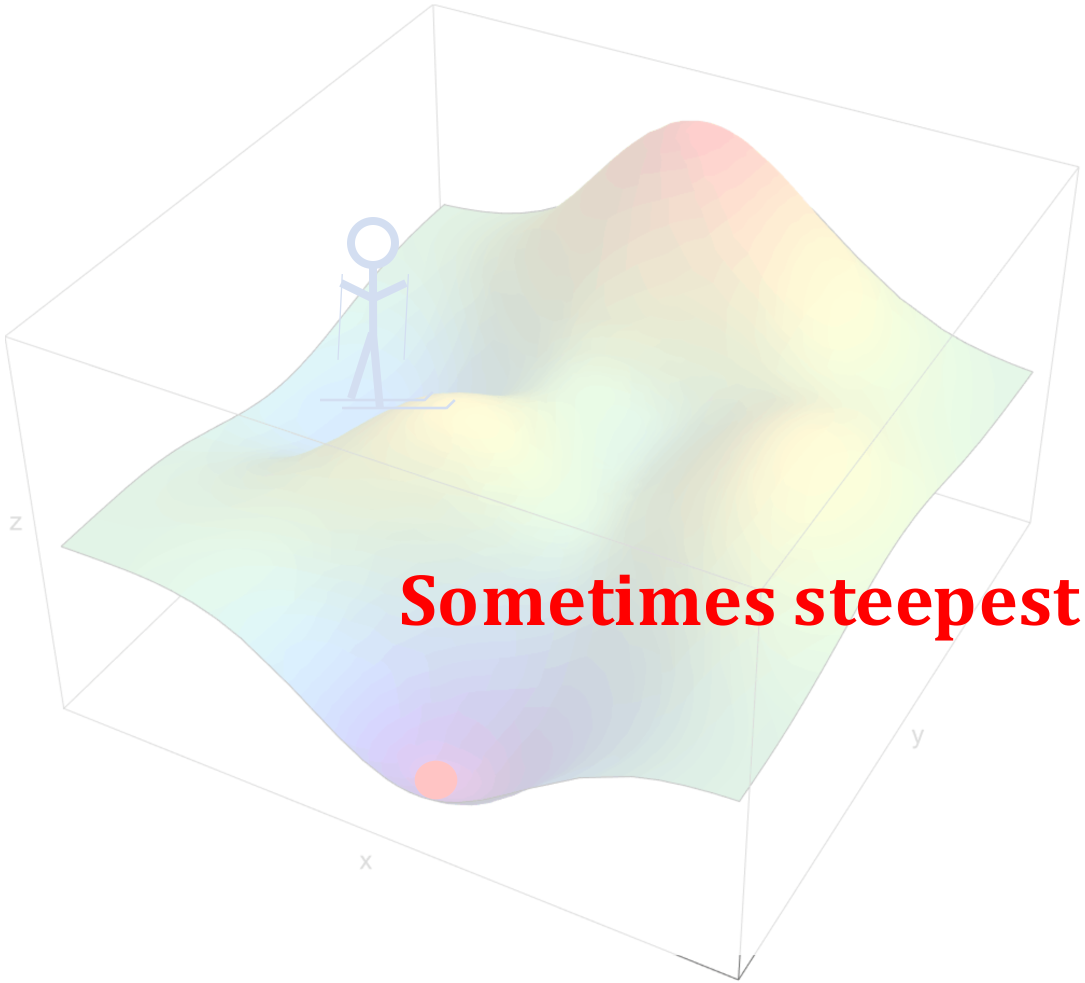
Simple idea: follow direction of steepest descent!

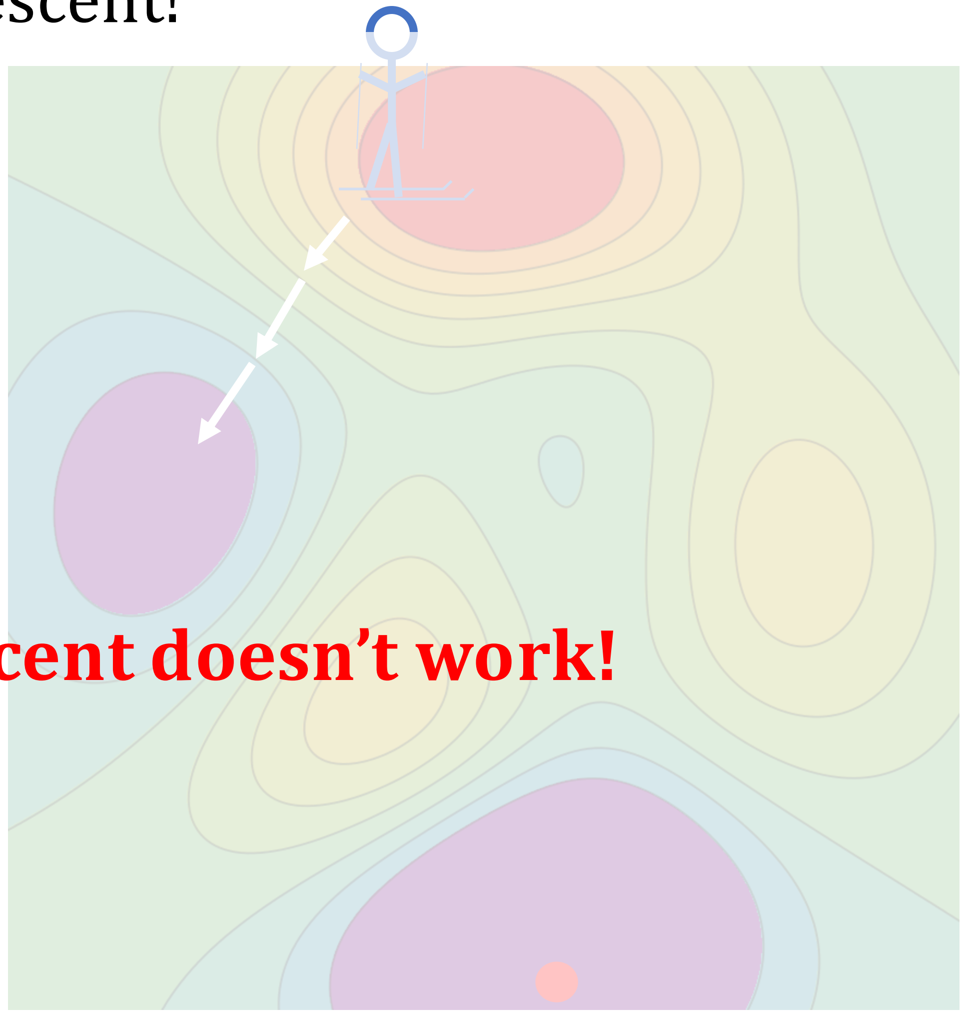




Steepest Descent

Simple idea: follow direction of steepest descent!





Sometimes steepest descent doesn't work!

Steepest Descent

Q: For which optim problems does steepest descent work?

Steepest Descent

Q: For which optim problems does steepest descent work?

→ see Part II of my PhD thesis!

Steepest Descent

Q: For which optim problems does steepest descent work?
→ see Part II of my PhD thesis!

Simple application from thesis: Locating a drone from distance sensors

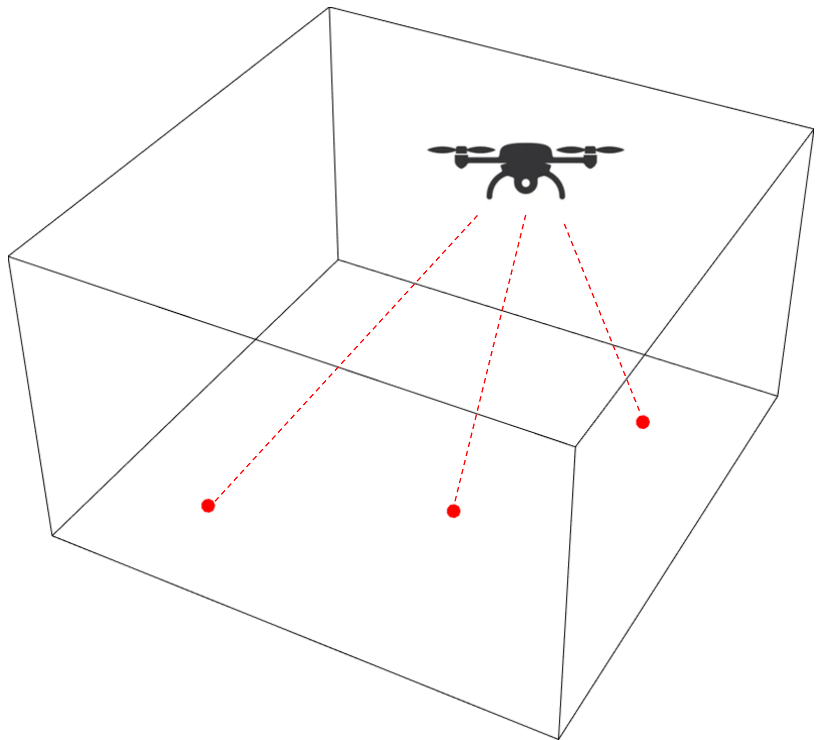
Steepest Descent

Q: For which optim problems does steepest descent work?

→ see Part II of my PhD thesis!

Simple application from thesis: Locating a drone from distance sensors

Given: distances between **drone** and **sensors**



Steepest Descent

Q: For which optim problems does steepest descent work?
→ see Part II of my PhD thesis!

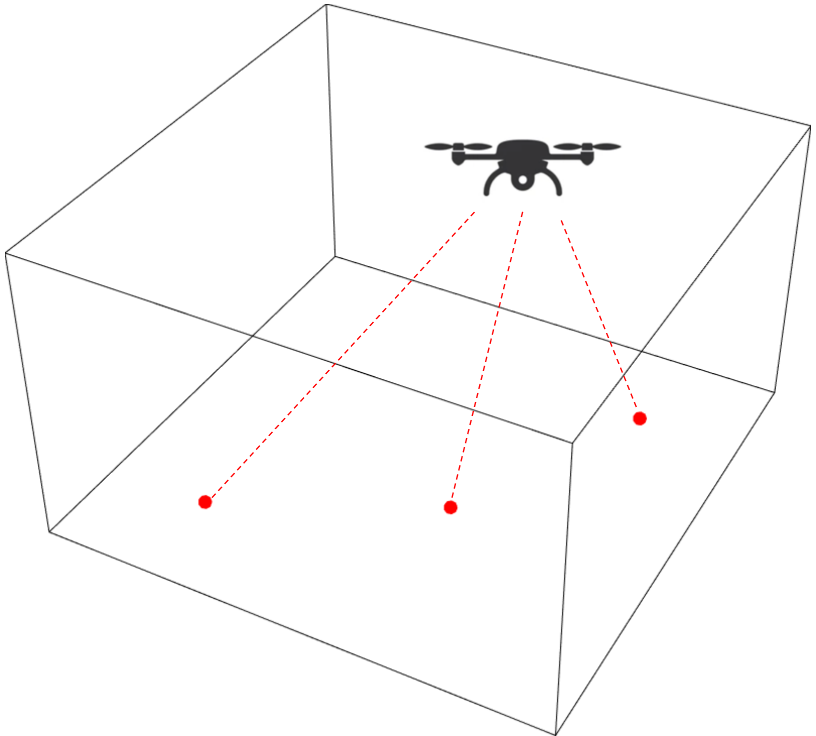
Simple application from thesis: Locating a drone from distance sensors

Given: distances between **drone** and **sensors**

Goal: Find location of drone, i.e., x, y, z-coordinates

$$\min \sum_{ij \in E} \left(\|z_i - z_j\|^2 - d_{ij}^2 \right)^2,$$

over $z_1, z_2, \dots, z_n \in \mathbb{R}^k$



Steepest Descent

Q: For which optim problems does steepest descent work?
→ see Part II of my PhD thesis!

Simple application from thesis: Locating a drone from distance sensors

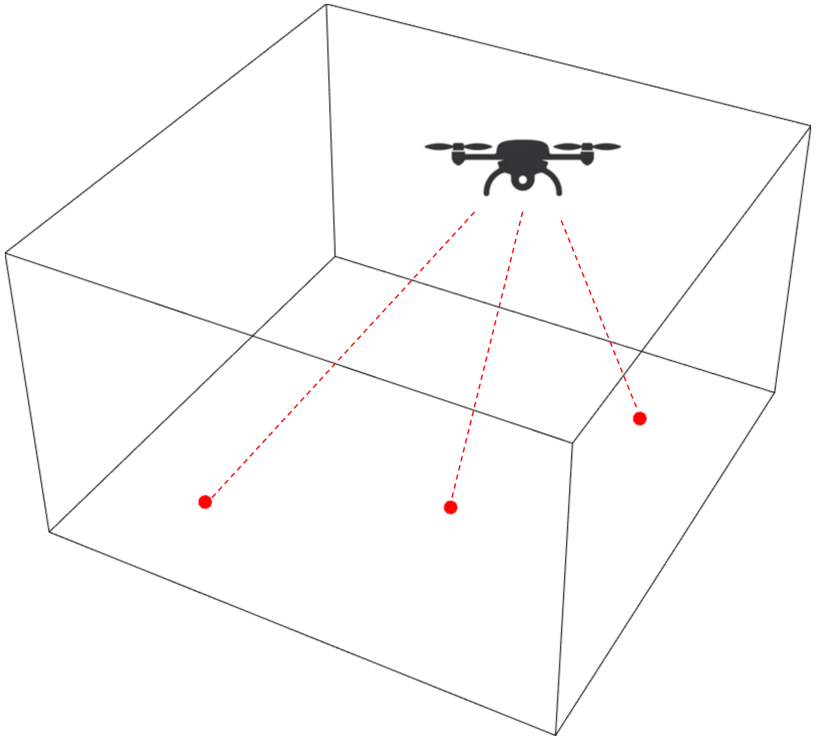
Given: distances between **drone** and **sensors**

Goal: Find location of drone, i.e., x, y, z-coordinates

Thm: Steepest descent always works! (under assumptions)

$$\min \sum_{ij \in E} \left(\|z_i - z_j\|^2 - d_{ij}^2 \right)^2,$$

over $z_1, z_2, \ldots, z_n \in \mathbb{R}^k$



Steepest Descent

Q: For which optim problems does steepest descent work?

→ see Part II of my PhD thesis!

Simple application from thesis: Locating a drone from distance sensors

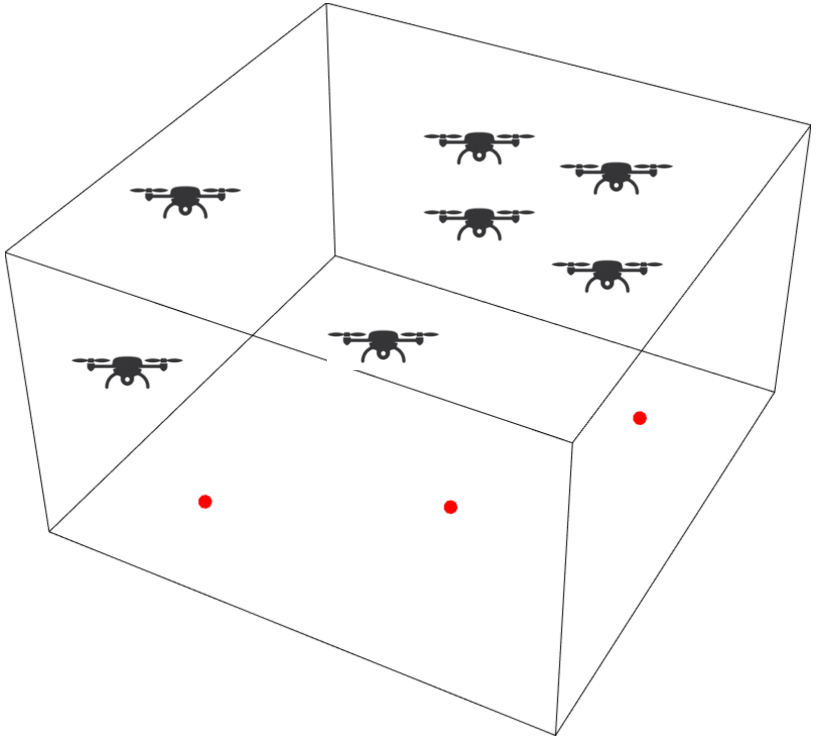
Given: distances between **drone** and **sensors**

Goal: Find location of drone, i.e., x, y, z-coordinates

Thm: Steepest descent always works! (under assumptions)

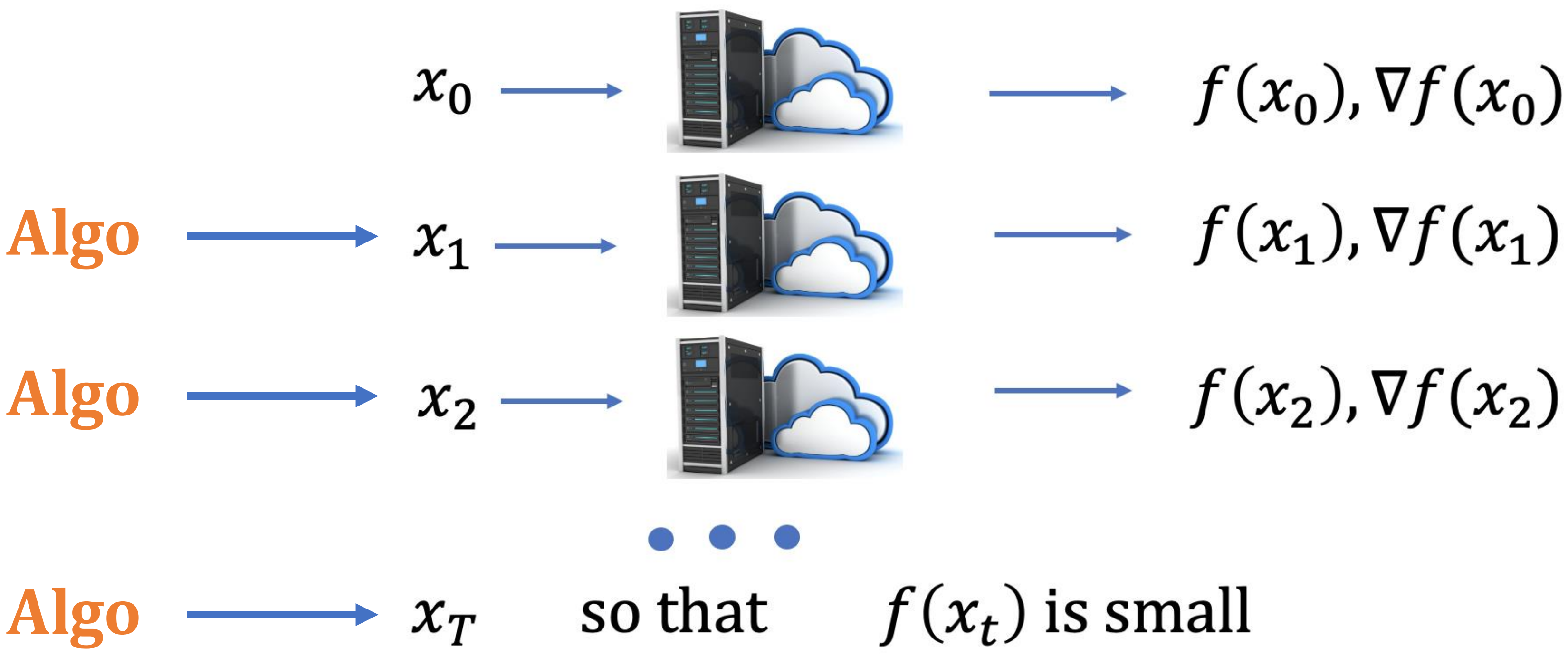
$$\min \sum_{ij \in E} \left(\|z_i - z_j\|^2 - d_{ij}^2 \right)^2,$$

over $z_1, z_2, \dots, z_n \in \mathbb{R}^k$



Best algorithm?

Other algorithms than steepest descent?



Best algorithm?

Other algorithms than steepest descent?

Yes! SGD, Nesterov acceleration, Trust regions, …

$x_0 \longrightarrow$  $\longrightarrow f(x_0), \nabla f(x_0)$

Algo $\longrightarrow x_1 \longrightarrow$  $\longrightarrow f(x_1), \nabla f(x_1)$

Algo $\longrightarrow x_2 \longrightarrow$  $\longrightarrow f(x_2), \nabla f(x_2)$

• • •

Algo $\longrightarrow x_T$ so that $f(x_t)$ is small

Best algorithm?

Other algorithms than steepest descent?

Yes! SGD, Nesterov acceleration, Trust regions, …

Which algo is **best**? [Smallest time T to finish. Eg, assume 1 query per second]

Best algorithm?

Other algorithms than steepest descent?

Yes! SGD, Nesterov acceleration, Trust regions, …

Which algo is **best**? [Smallest time *T* to finish. Eg, assume 1 query per second]

- Space of algos is huge (infinite), can't test all of them on your problem!

Best algorithm?

Other algorithms than steepest descent?

Yes! SGD, Nesterov acceleration, Trust regions, …

Which algo is **best**? [Smallest time T to finish. Eg, assume 1 query per second]

- Space of algos is huge (infinite), can't test all of them on your problem!
- That's where math enters – ***prove*** there is no better algorithm

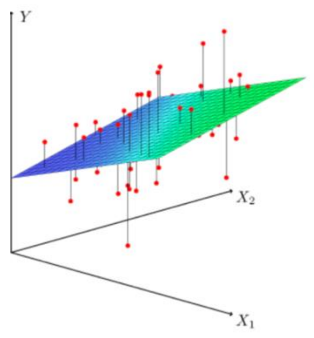
Best algorithm?

Consider a **class** of optimization problems $\{f_1, f_2, \ldots\}$

$$\min_{x \in M} f_i(x)$$

Best algorithm?

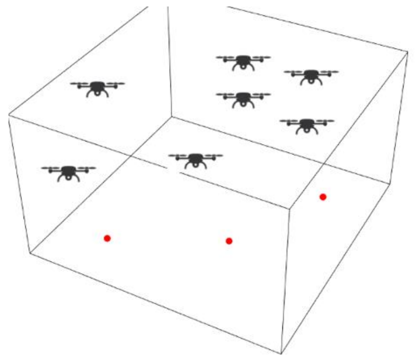
Consider a **class** of optimization problems $\{f_1, f_2, \dots\}$



$$\min_{x \in M} f_i(x)$$

Examples: Linear regression with different sets of data,

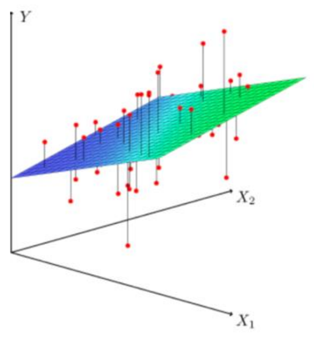
Drone localization with different distance measurements



Best algorithm?

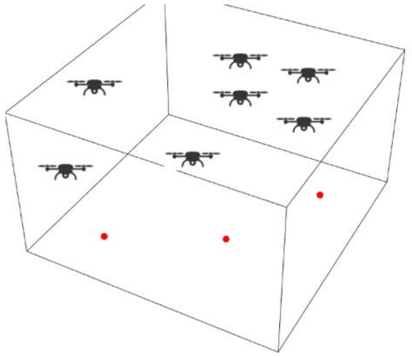
Consider a **class** of optimization problems $\{f_1, f_2, \ldots\}$

$$\min_{x \in M} f_i(x)$$



Examples: Linear regression with different sets of data,

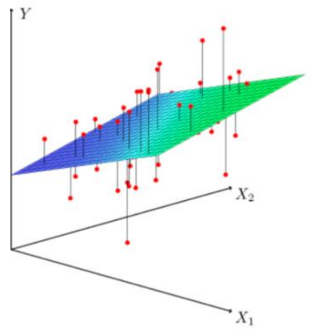
Drone localization with different distance measurements



Amanda, Chris, Joe each have an algo

Best algorithm?

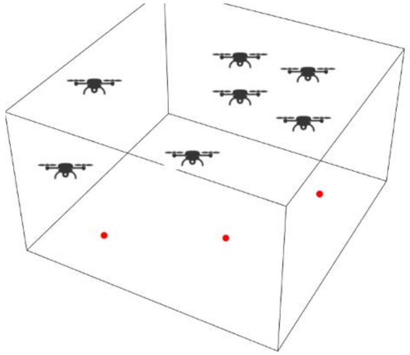
Consider a **class** of optimization problems $\{f_1, f_2, \ldots\}$



$$\min_{x \in M} f_i(x)$$

Examples: Linear regression with different sets of data,

Drone localization with different distance measurements



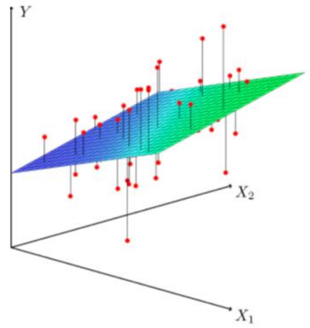
Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

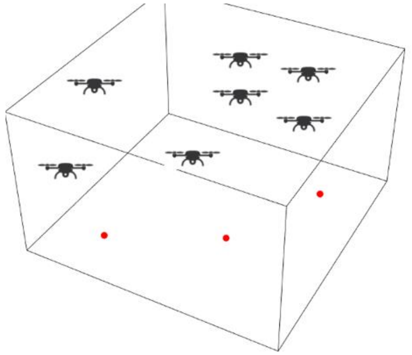
Best algorithm?

Consider a **class** of optimization problems $\{f_1, f_2, \dots\}$

$$\min_{x \in M} f_i(x)$$



Examples: Linear regression with different sets of data,
Drone localization with different distance measurements



Amanda, Chris, Joe each have an algo

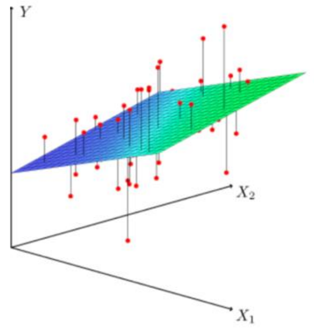
Amanda: Mine minimizes **every** problem in the class in at most 200 seconds.

Chris: ” ” 100 seconds.

Best algorithm?

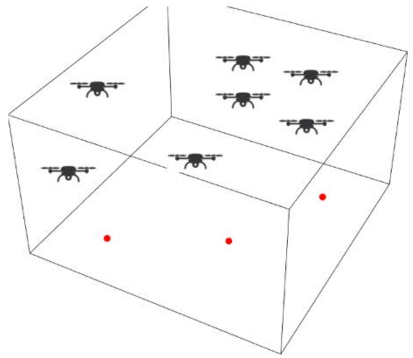
Consider a **class** of optimization problems $\{f_1, f_2, \dots\}$

$$\min_{x \in M} f_i(x)$$



Examples: Linear regression with different sets of data,

Drone localization with different distance measurements



Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Chris: I can prove that no other algo can do better than 100 seconds. Therefore,

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Chris: I can prove that no other algo can do better than 100 seconds. Therefore,

- My algo is the best (optimal)

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Chris: I can prove that no other algo can do better than 100 seconds. Therefore,

- My algo is the best (optimal)
- Amanda’s is suboptimal

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Chris: I can prove that no other algo can do better than 100 seconds. Therefore,

- My algo is the best (optimal)
- Amanda’s is suboptimal
- Joe is lying

Best algorithm?

Amanda, Chris, Joe each have an algo

Amanda: Mine minimizes **every** problem in the **class** in at most 200 seconds.

Chris: ” ” 100 seconds.

Joe: “ “ 50 seconds.

Chris: I can prove that no other algo can do better than 100 seconds. Therefore,

- My algo is the best (optimal)
- Amanda’s is suboptimal
- Joe is lying

Best algorithm?

Consider a **class** of optimization problems $\{f_1, f_2, \ldots\}$

$$\min_{x \in M} f_i(x)$$

Chris: I can prove that no other algo can do better than 100 seconds.

Best algorithm?

Consider a **class** of optimization problems $\{f_1, f_2, \dots\}$

$$\min_{x \in M} f_i(x)$$

Chris: I can prove that no other algo can do better than 100 seconds.

What this means: For every algo A, I mathematically construct a problem f in the **class** so that running A on f takes at least 100 seconds to finish.

Best algorithm?

Consider a **class** of optimization problems $\{f_1, f_2, \dots\}$

$$\min_{x \in M} f_i(x)$$

$\boxed{\text{Chris: I can prove that no other algo can do better than 100 seconds.}}$

What this means: For every algo A, I mathematically construct a problem f in the **class** so that running A on f takes at least 100 seconds to finish.

→ Part I of my thesis, looks at a specific **class** (geodesically convex functions), and proves there is a fundamental limit for that class.

Best algorithm?

→ Part I of my thesis, looks at a specific **class** (geodesically convex functions), and proves there is a fundamental limit for that class.

Best algorithm?

→ Part I of my thesis, looks at a specific **class** (geodesically convex functions), and proves there is a fundamental limit for that class.

Some consequences:

- Steepest descent is optimal for that **class** (under some assumptions)

Best algorithm?

→ Part I of my thesis, looks at a specific **class** (geodesically convex functions), and proves there is a fundamental limit for that class.

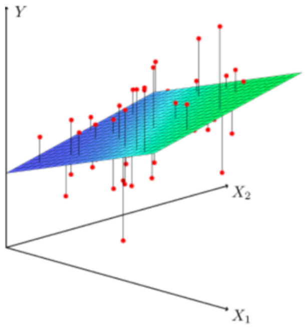
Some consequences:

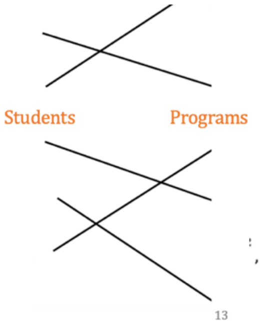
- Steepest descent is optimal for that **class** (under some assumptions)
- Surprising because expectation in community was that steepest descent is not optimal

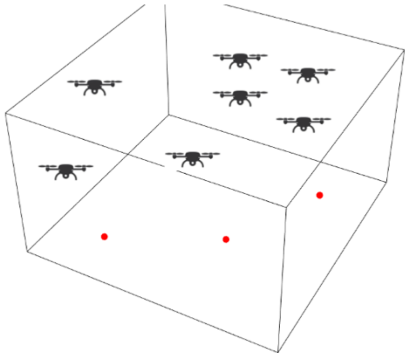
Steepest Descent	Nesterov Accelerated Descent?
Optimal!	**Not Possible!**

Some takeaways

Optimization everywhere

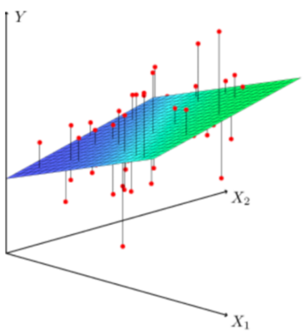


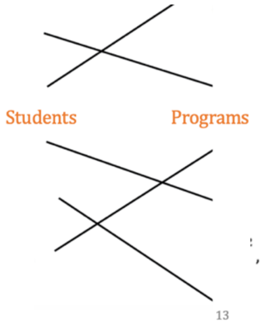


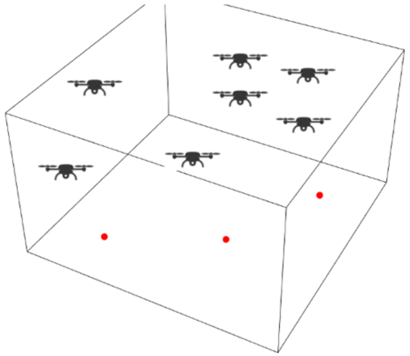


Some takeaways

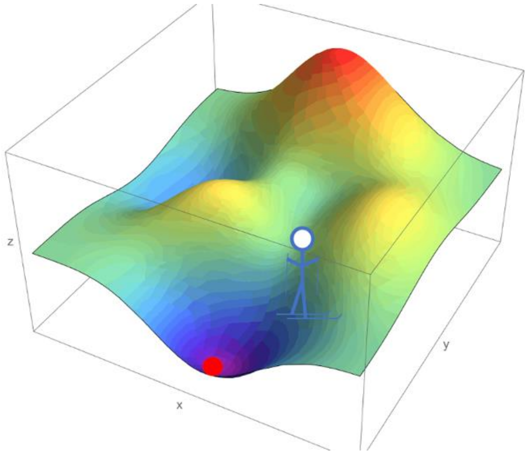
Optimization everywhere

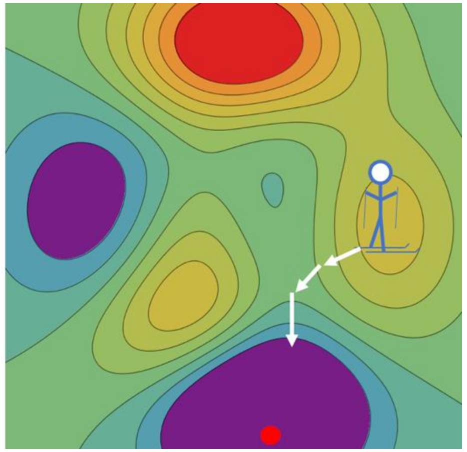






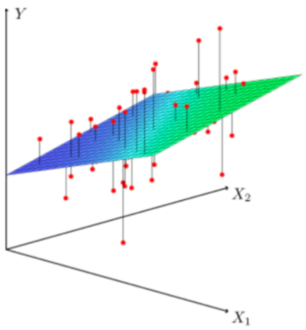
Steepest descent

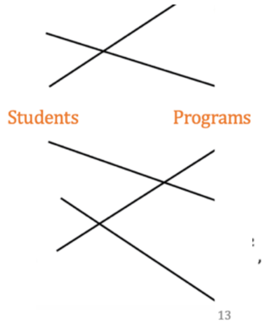


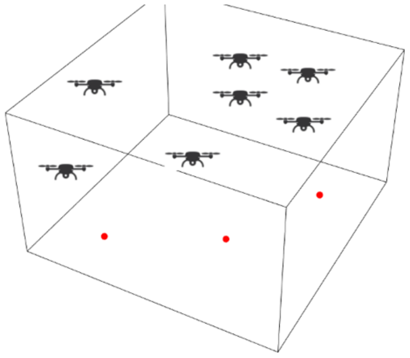


Some takeaways

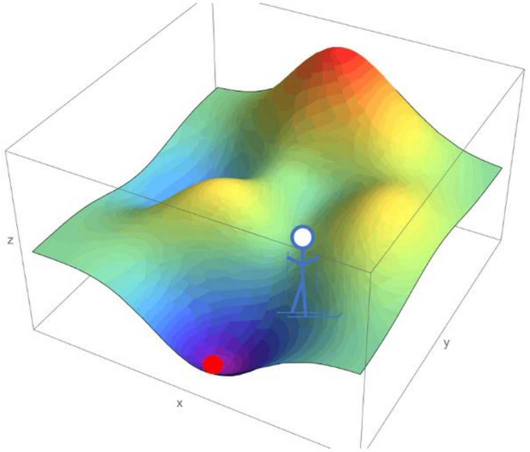
Optimization everywhere

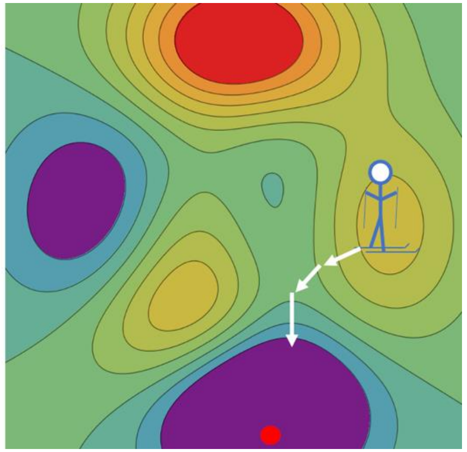






Steepest descent





Math in optim: best algorithm

Steepest Descent
Optimal!

Nesterov Accelerated Descent?
Not Possible!