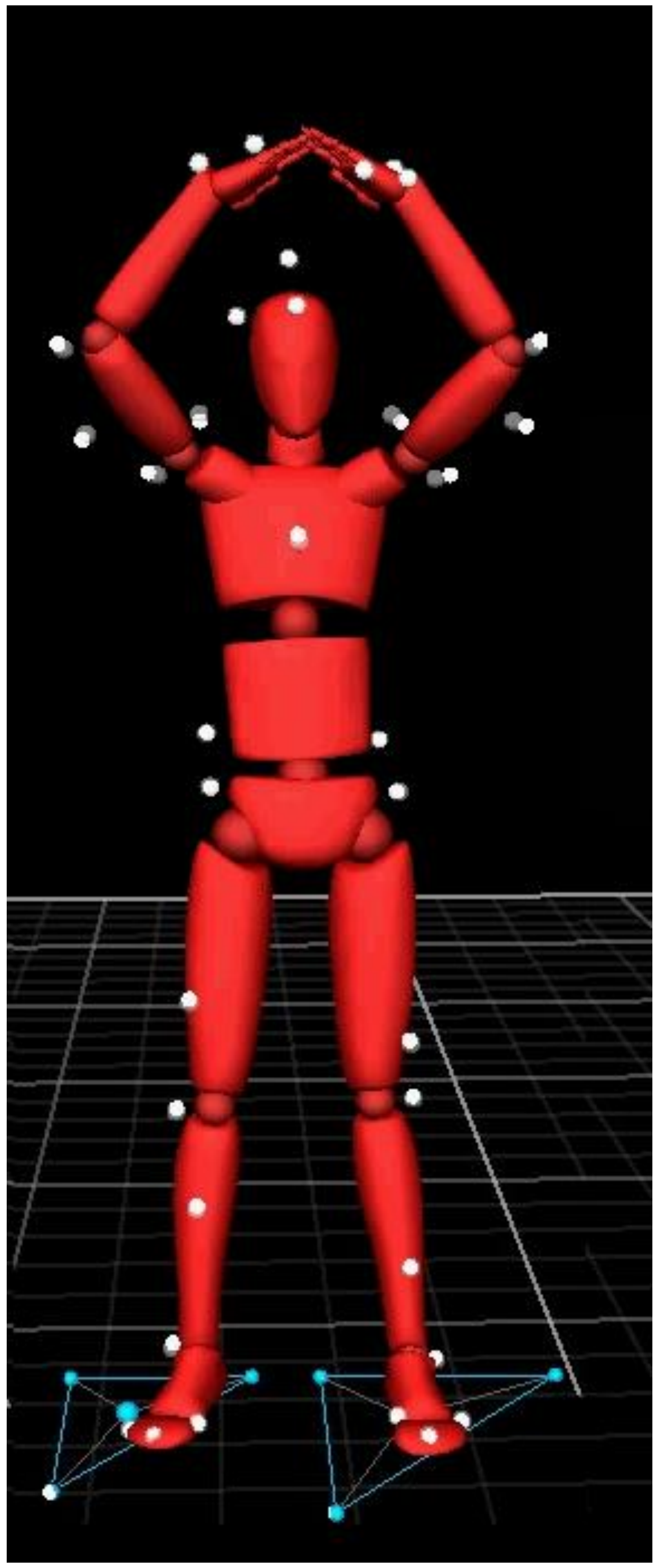
Intoduction to Optimization

Computer Graphics
CMU 15-462/15-662, Fall 2016

Last time: Rigging, FK, IK

- **Last Class:**
  - Rigging, FK, IK
  - Animate high-resolution meshes using a much smaller number of parameters



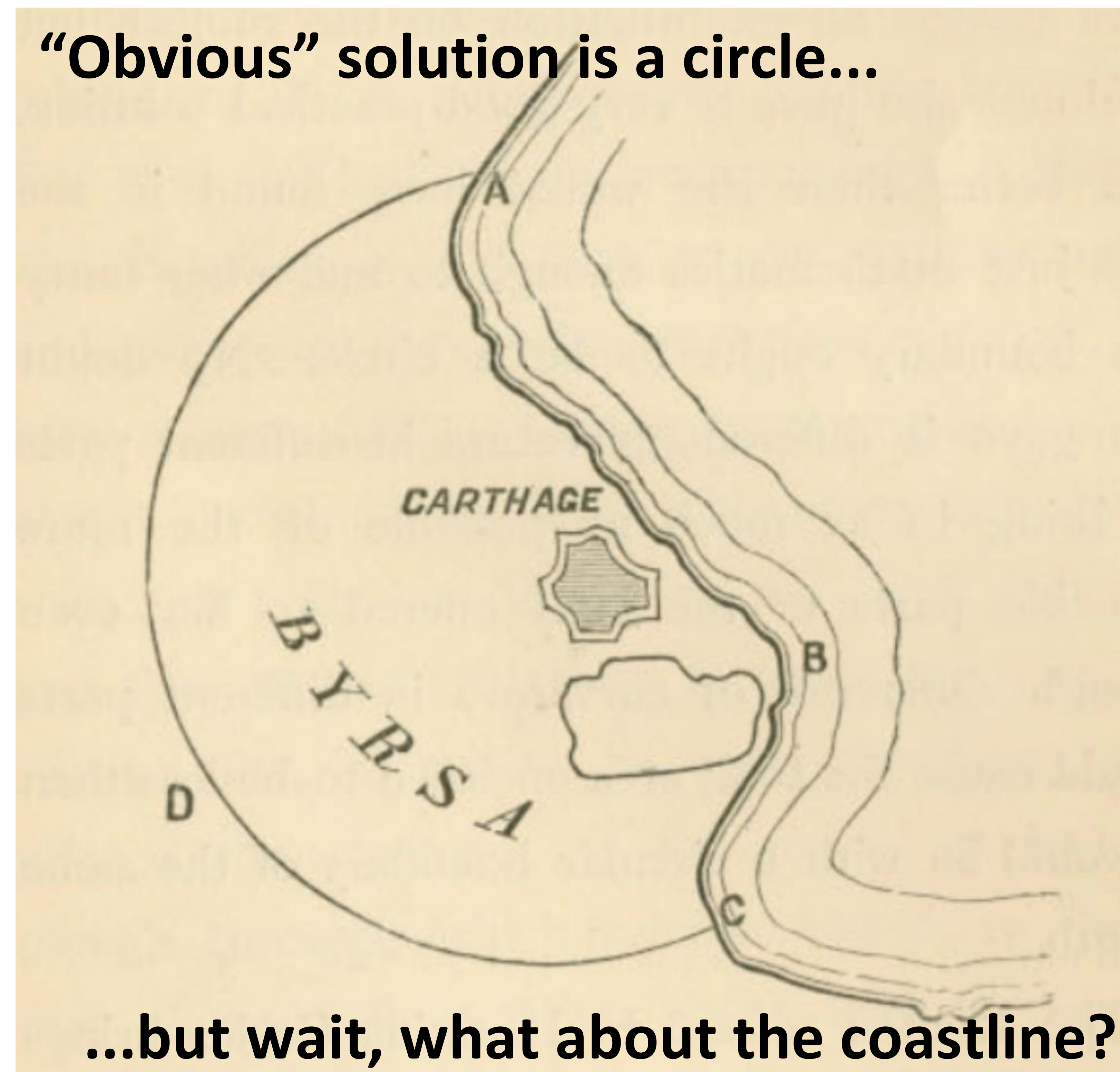
- **Today:** numerical optimization
  - A general, powerful tool
  - basic idea: "ski downhill" to get a better solution
  - used everywhere in graphics (not just animation)
  - We'll see an application for IK
  - image processing, geometry, rendering, ...

What is an optimization problem?

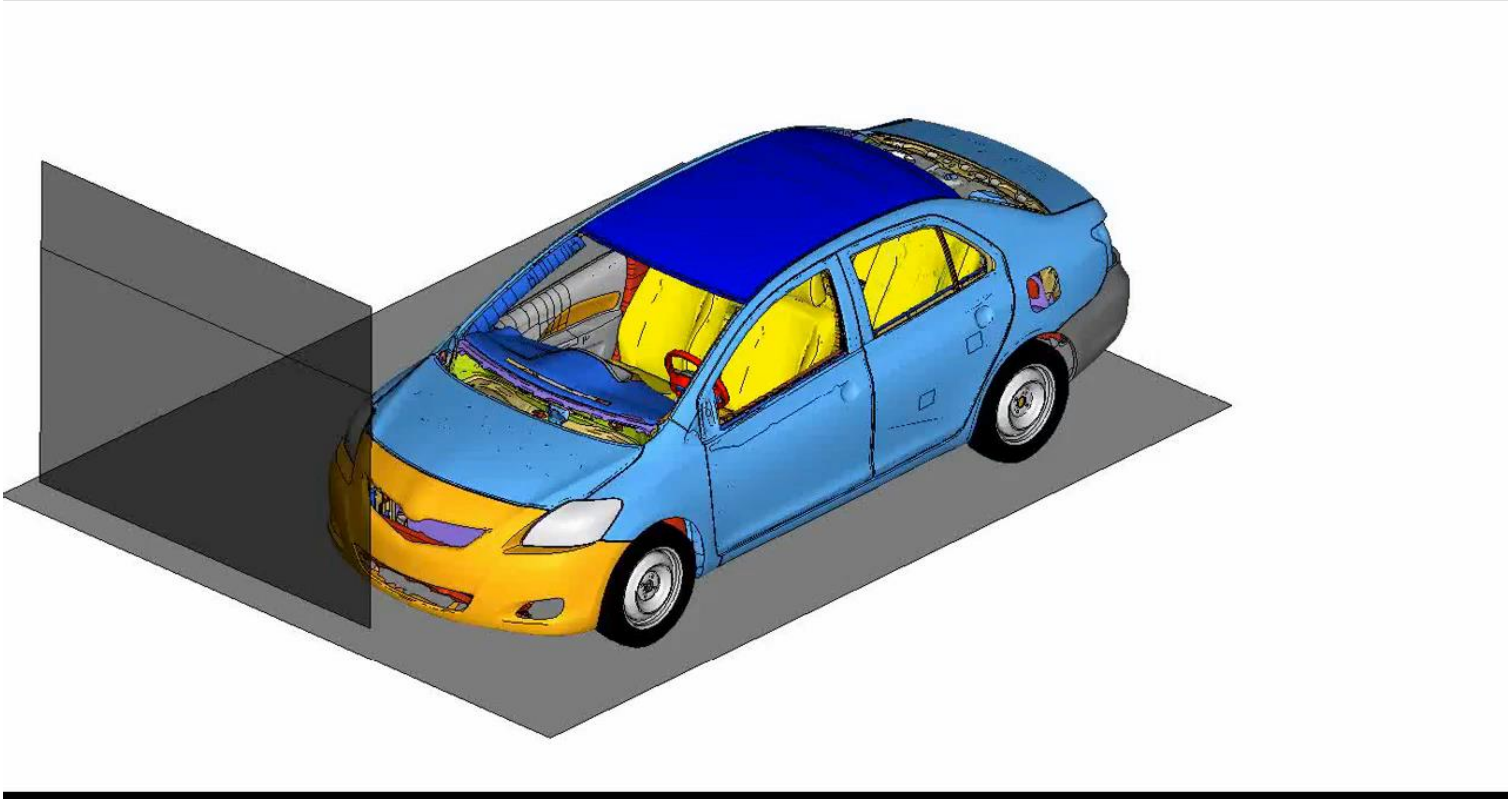
- **Natural human desire: find the best solution among all possibilities (subject to certain constraints)**
- **E.g., cheapest flight, shortest route, tastiest restaurant ...**
- **Has been studied since antiquity, e.g., isoperimetric problem:**

"The first optimization problem known in history was practically solved by Dido, a clever Phoenician princess, who left her Tyrian home and emigrated to North Africa, with all her property and a large retinue, because her brother Pygmalion murdered her rich uncle and husband Acerbas, and plotted to defraud her of the money which he left. On landing in a bay about the middle of the north coast of Africa she obtained a grant from Hiarbas, the native chief of the district, of as much land as she could enclose with an ox-hide. She cut the ox-hide into an exceedingly long strip, and succeeded in enclosing between it and the sea a very valuable territory on which she build Carthage."

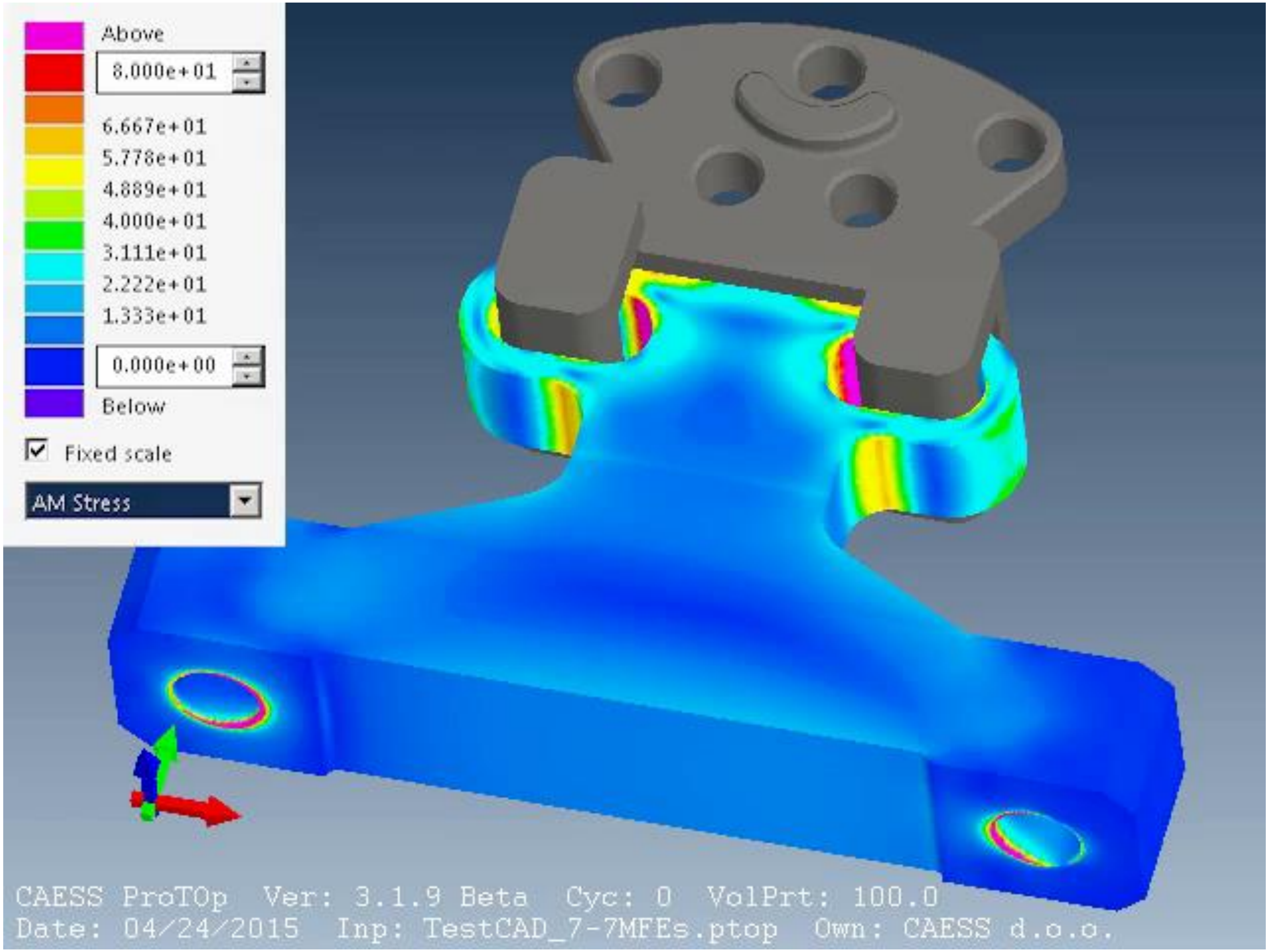
—Lord Kelvin, 1893



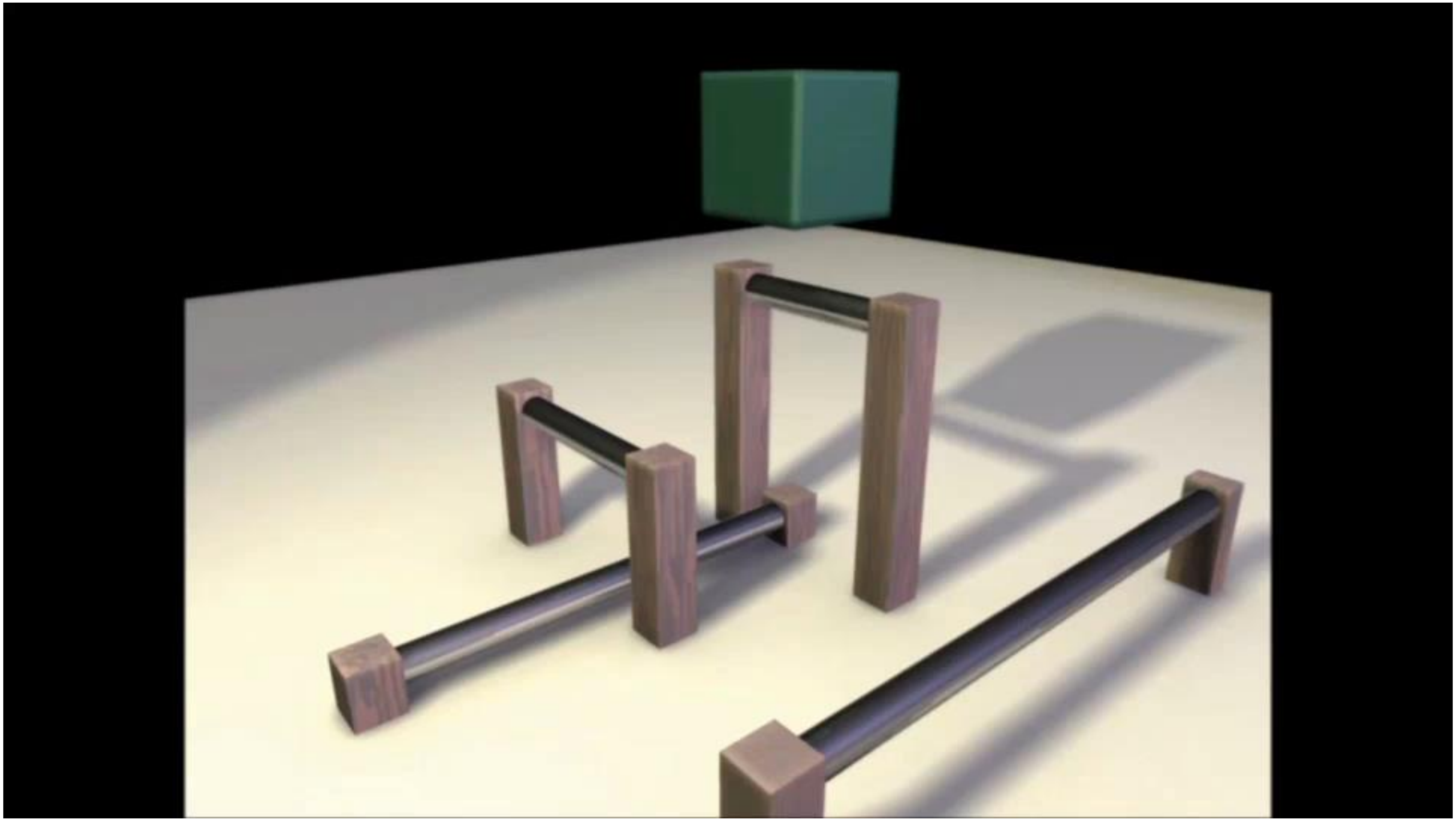
Go Optimization, Go!



Go Optimization, Go!

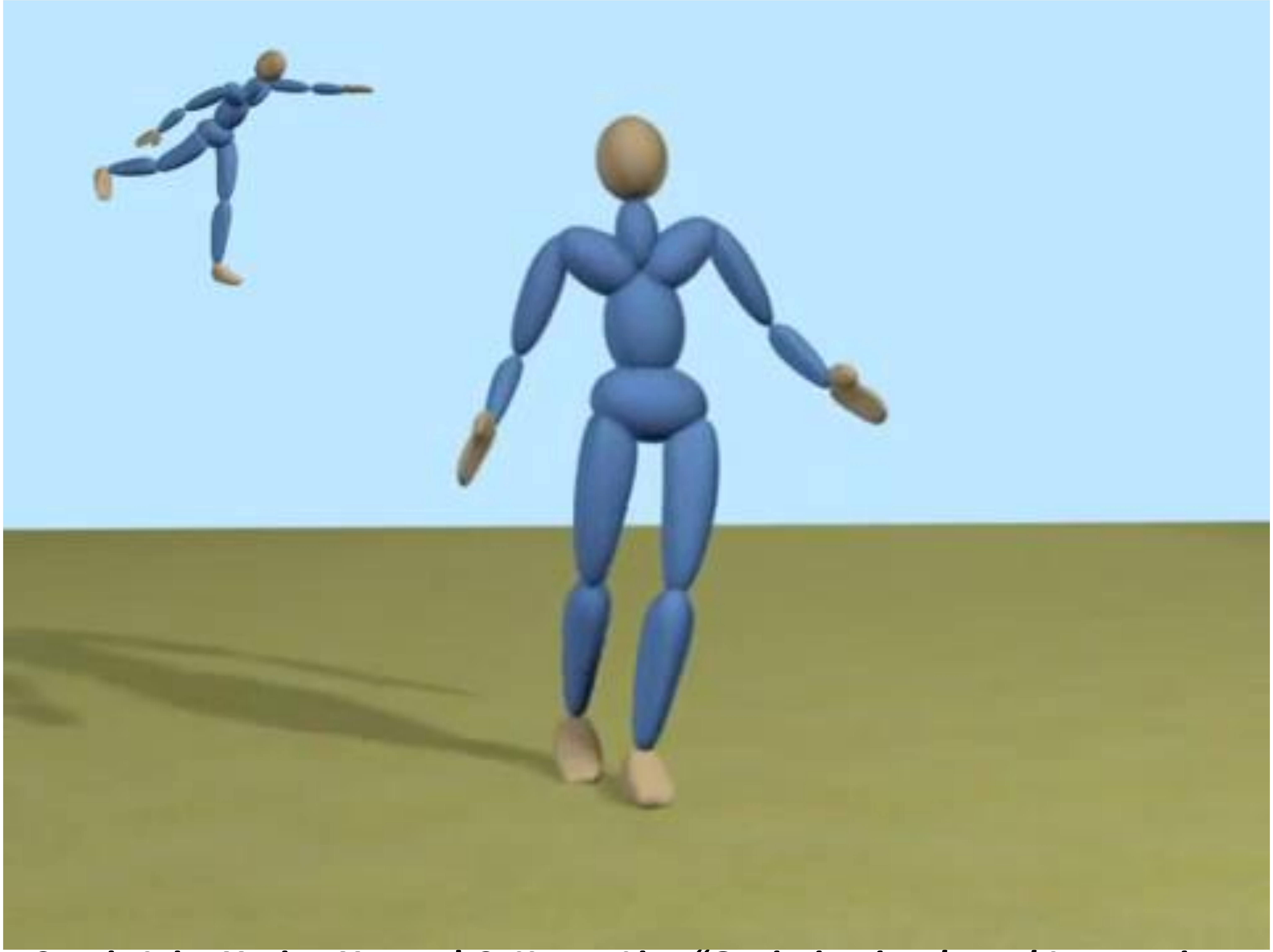


Optimization in Graphics



C. Wojtan and G. Turk, ***"Fast Viscoelastic Behavior with Thin Features"***

Optimization in Graphics

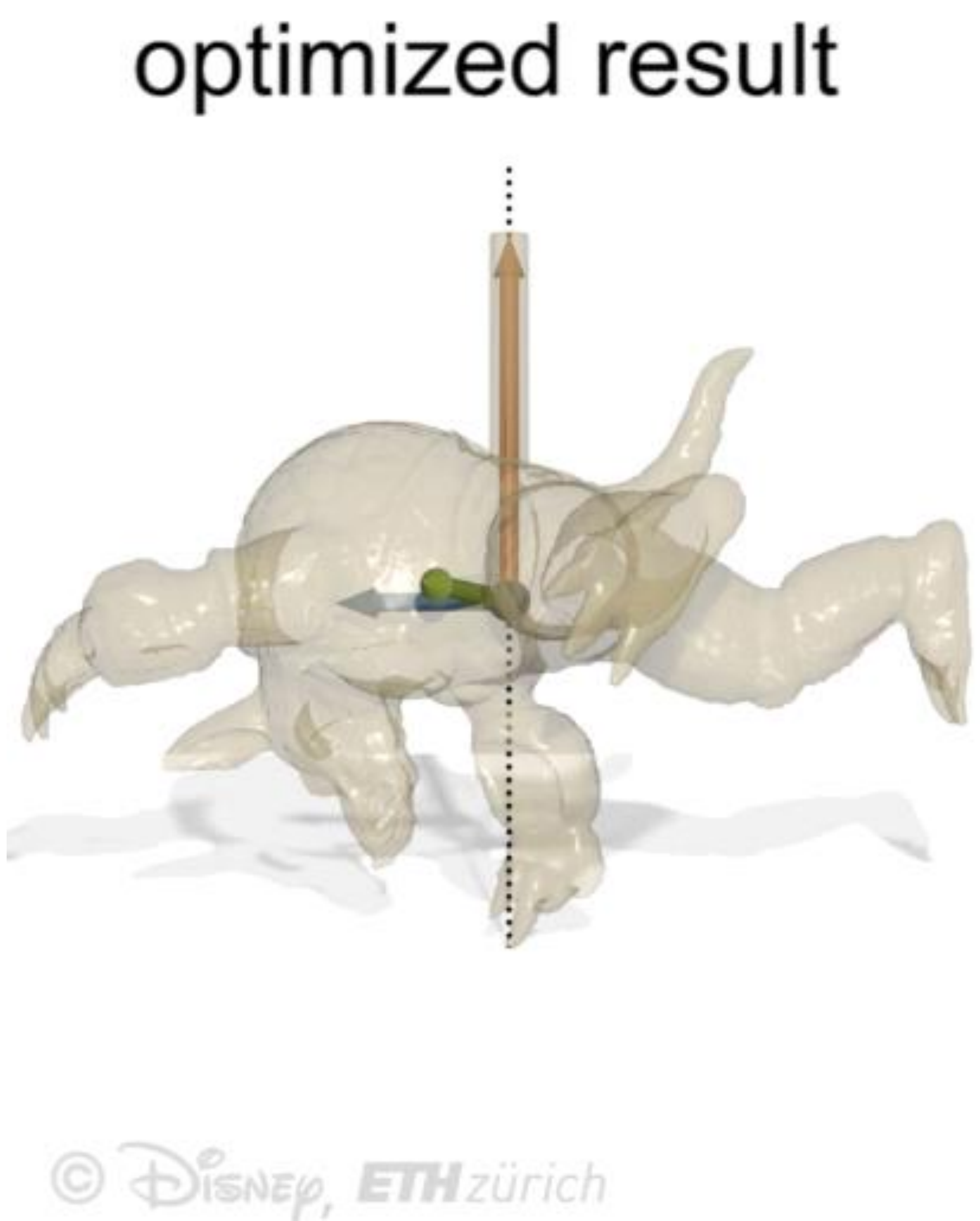


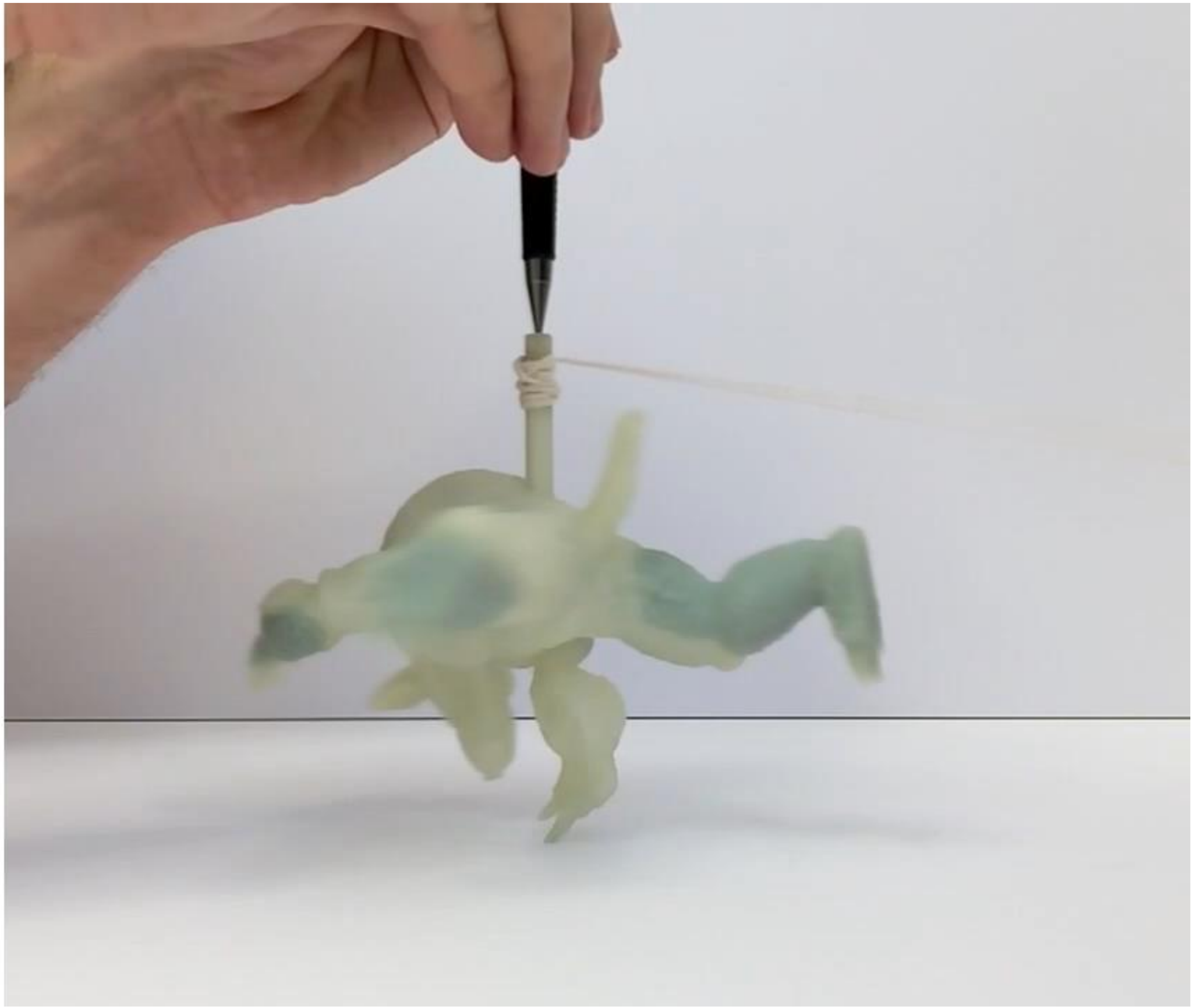
Sumit Jain, Yuting Ye, and C. Karen Liu, *"Optimization-based Interactive Motion Synthesis"*

Optimization in Graphics

Niloy J. Mitra, Leonidas Guibas, Mark Pauly,
"Symmetrization"

Optimization in Graphics





Moritz Bächer, Emily Whiting, Bernd Bickel, Olga Sorkine-Hornung, *"Spin-It: Optimizing Moment of Inertia for Spinnable Objects"*

Optimization in Graphics

Nobuyuki Umetani, Yuki Koyama, Ryan Schmidt & Takeo Igarashi,
"Pteromys: Interactive Design and Optimization of Free-formed Free-flight Model Airplanes"

Continuous vs. Discrete Optimization

- DISCRETE:
  - domain is a discrete set (e.g. integers)
  - Example: best vegetable to put in a stew
    - Basic strategy? Try them all! (exponential)
    - sometimes clever strategy (e.g., MST)
    - more often, NP-hard (e.g., TSP)

- CONTINUOUS:
  - domain is not discrete (e.g., real numbers)
  - Example: best temperature to cook an egg
  - still many (NP-)hard problems, but also large classes of “easy” problems (e.g., convex)
  - Gradient information may or may not be available

Optimization Problem in Standard Form

- **Can formulate most continuous optimization problems this way:**

"objective": how much does solution x cost?

$$(f_i : \mathbb{R}^n \to \mathbb{R},\ i = 0, \ldots, m)$$

often (but not always) continuous, differentiable, ...

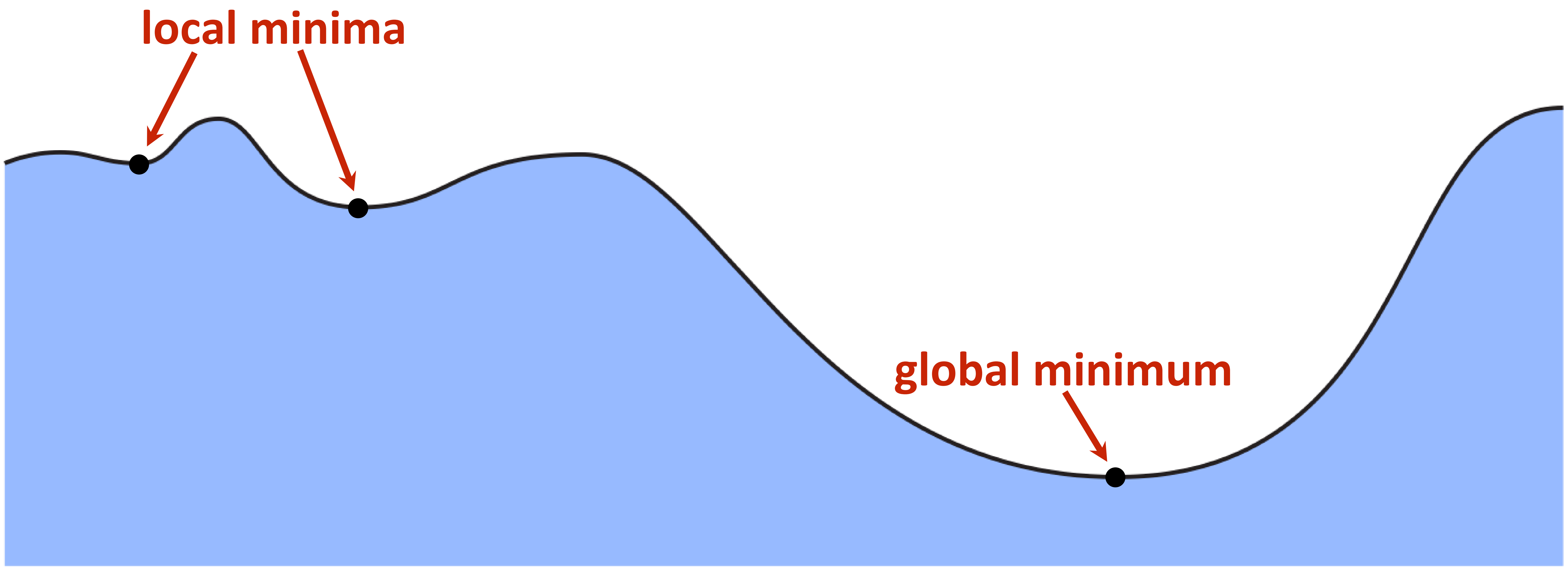
$$\begin{array}{ll} \min\limits_{x \in \mathbb{R}^n} & f_0(x) \\ \text{subject to} & f_i(x) \leq b_i,\ i = 1, \ldots, m \end{array}$$

"constraints": what must be true about x? ("x is *feasible*")

- ***Optimal solution*** **x* has smallest value of f_0 among all feasible x**
- **Q: What if we want to *maximize* something instead?**
- **A: Just flip the sign of the objective!**
- **Q: What if we want *equality* constraints, rather than inequalities?**
- **A: Include two constraints: g(x) ≤ c and g(x) ≤ -c**

Local vs. Global Minima

- *Global* minimum is absolute best among all possibilities
- *Local* minimum is best "among immediate neighbors"

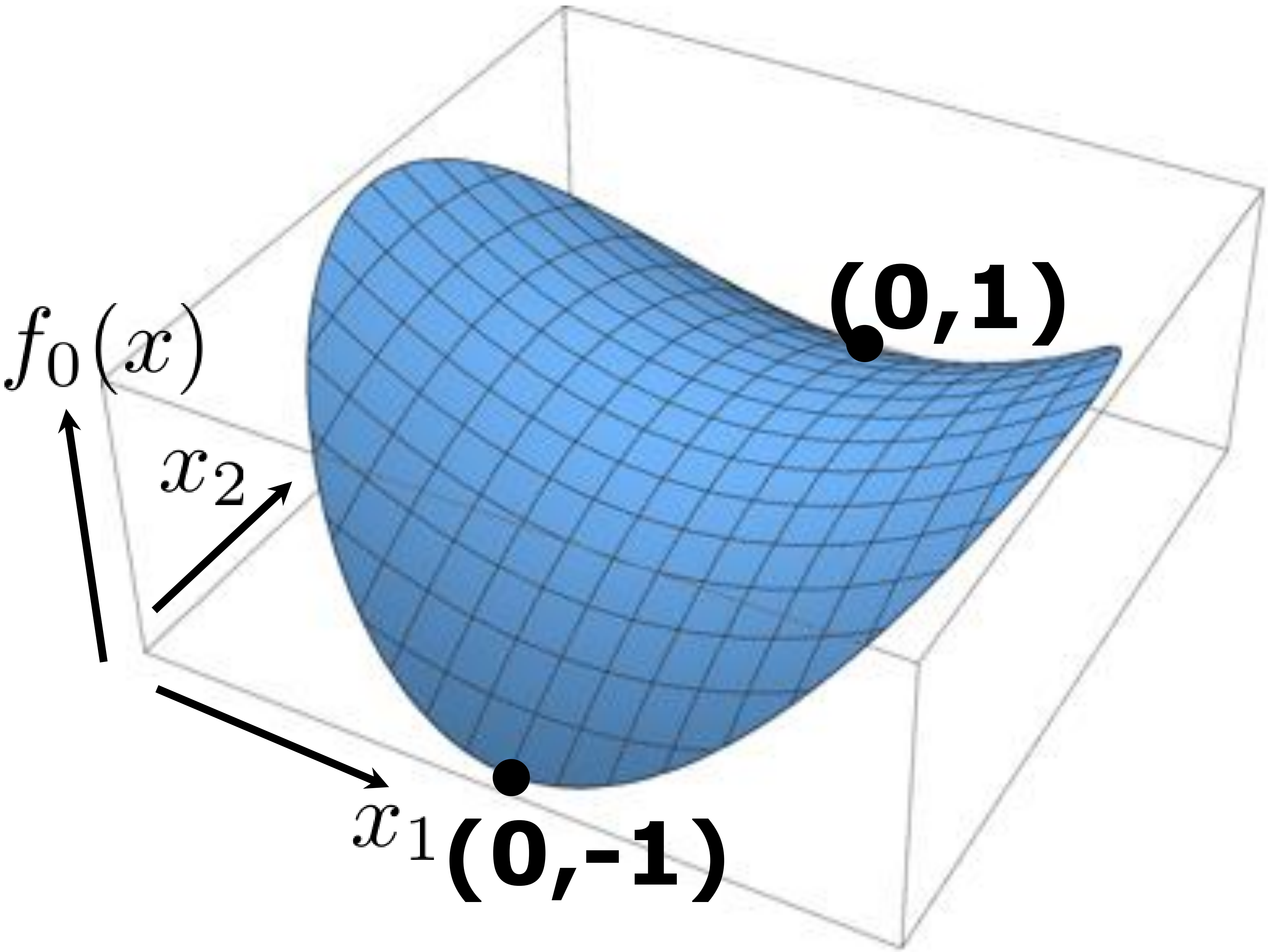


Philosophical question: does a local minimum *"solve"* the problem?

Depends on the problem! (E.g., real protein folding is *local* minimum)

Other times, local minima can be really bad (e.g., path planning)

Optimization Problem, Visualized



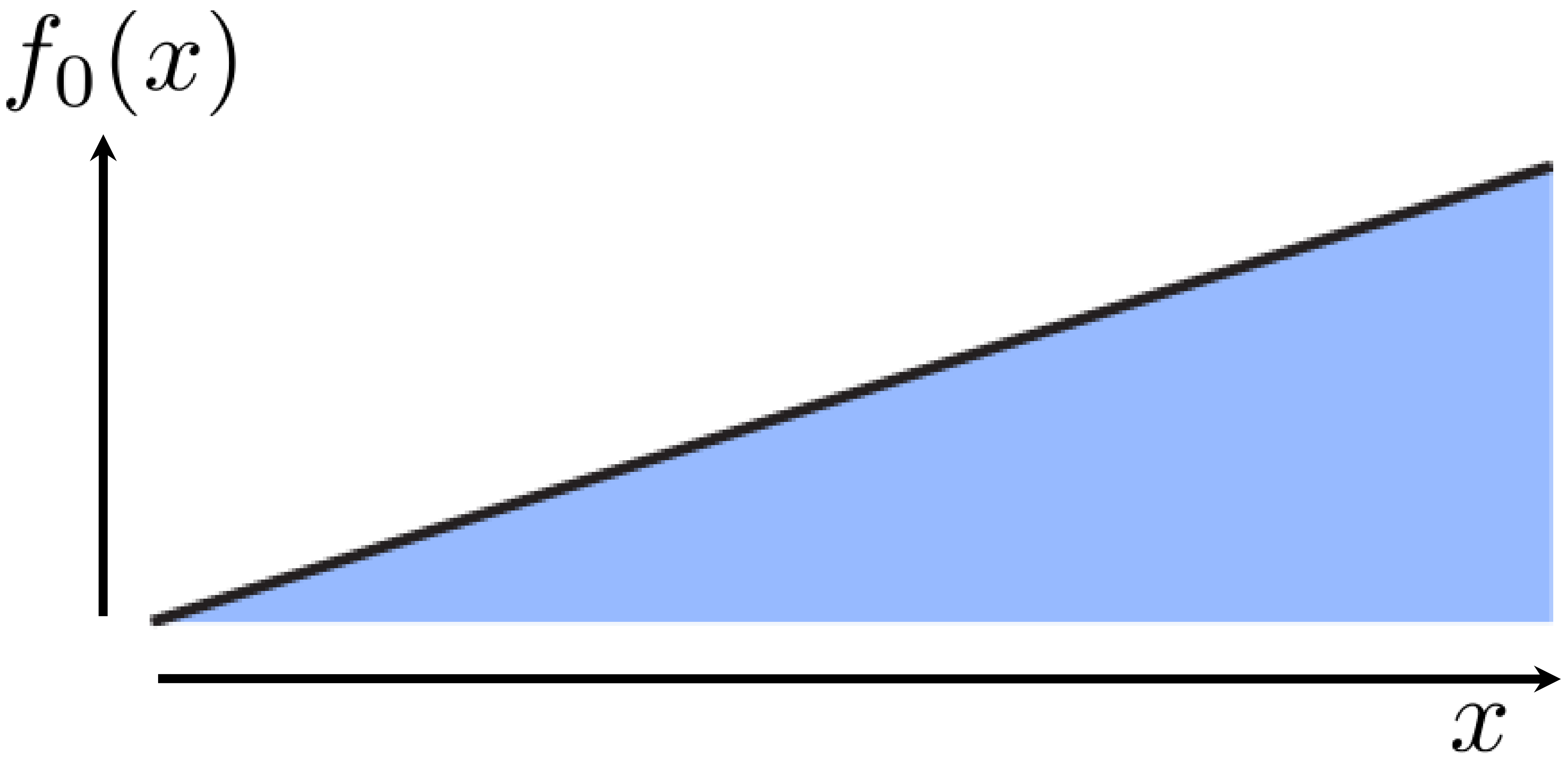
$$\begin{aligned} \min_{x \in \mathbb{R}^2} \quad & x_1^2 - x_2^2 \\ \text{s.t.} \quad & x_1^2 + x_2^2 - 1 \leq 0 \end{aligned}$$

Q: Is this an optimization problem in standard form? A: Yes.

Q: Where is the optimal solution? A: There are two, (0,1), (0,-1).

Existence & Uniqueness of Minimizers

- **Already saw that (global) minimizer is not unique.**
- **Does it always exist? Why?**
- **Just consider all possibilities and take the smallest one, right?**



perfectly reasonable optimization *problem*

$$\min_{x \in \mathbb{R}} x$$

clearly has no *solution*
(can always pick smaller x)

- **WRONG! Not all objectives are bounded from below.**
- **It's like that old adage: *"no matter how good you are, there will always be someone better than you."***

Feasibility

- Ok, but suppose the objective is bounded from below.
- Then we can just take the best feasible solution, right?

value of objective doesn't depend on x;
all feasible solutions are equally good

$$\min_{x \in \mathbb{R}^n} \quad 0$$
$$\text{subject to} \quad f_i(x) \leq b_i, \; i = 1, \ldots, m$$

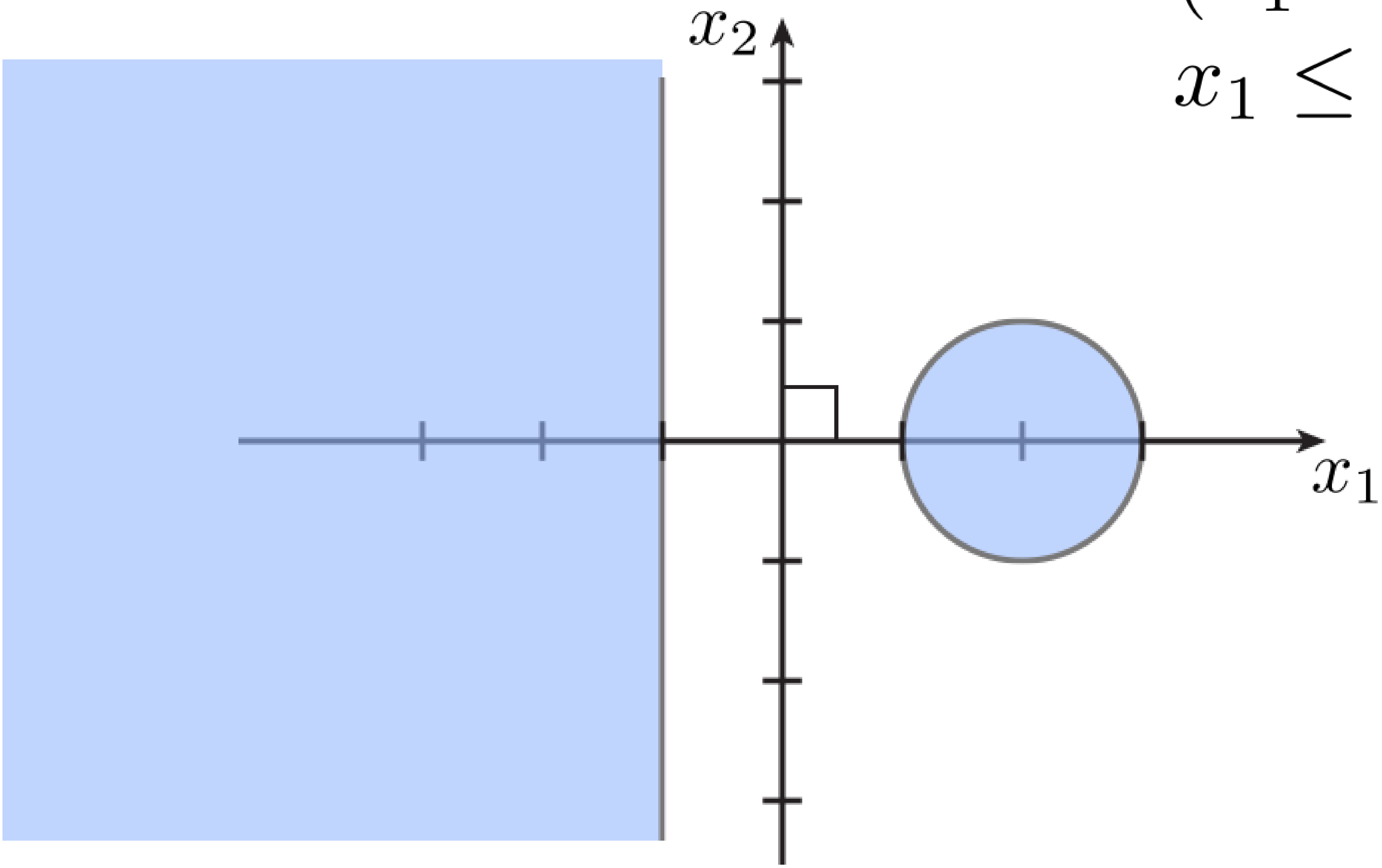
problem now is "just" finding a feasible solution—can be really hard, or impossible!

- Not if there aren't any!
- Every system of equations is an optimization problem.
- But not all problems have solutions!
- (You'll appreciate this more as you get older.)

Feasibility - Example

Q: Is this problem feasible?

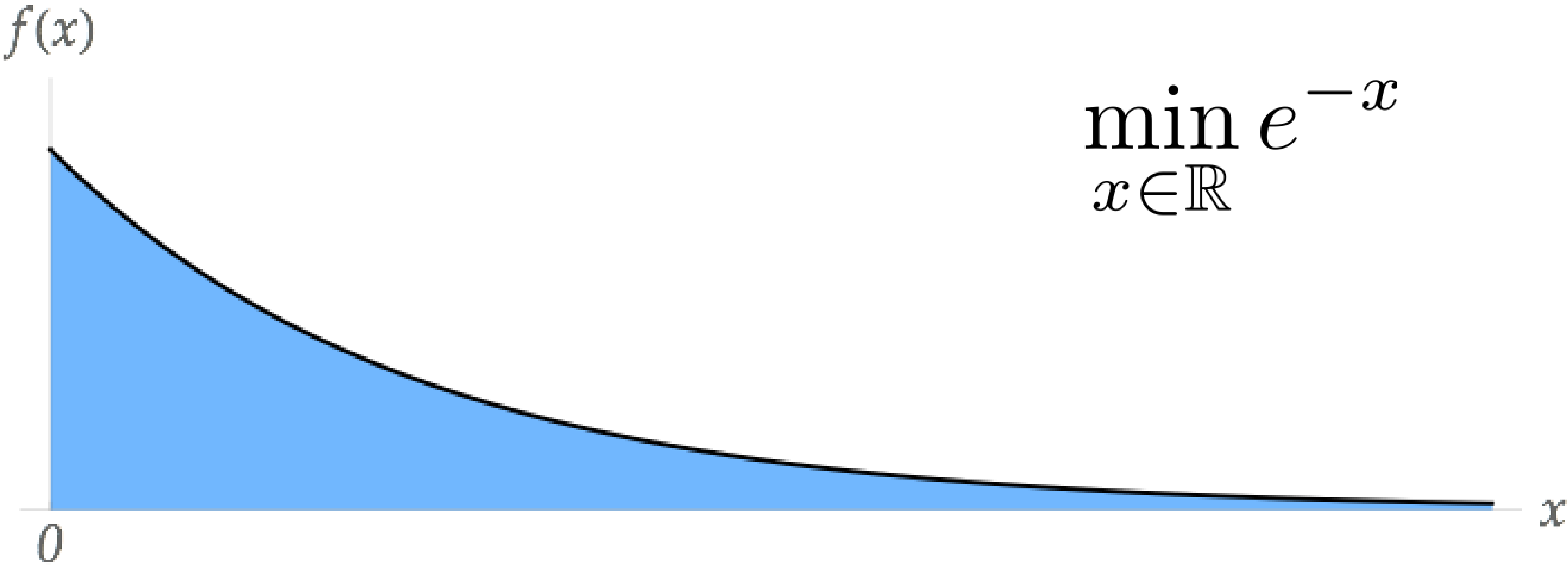
$$\min_{x\in\mathbb{R}^2} \quad \sin(x_1) + x_2^2$$
$$\text{s.t.} \quad (x_1 - 2)^2 + x_2^2 \leq 1,$$
$$x_1 \leq -1$$



A: No—the two sublevel sets (points where f_i(x) ≤ 0) have no common points, i.e., they do not overlap.

Existence & Uniqueness of Minimizers, cont.

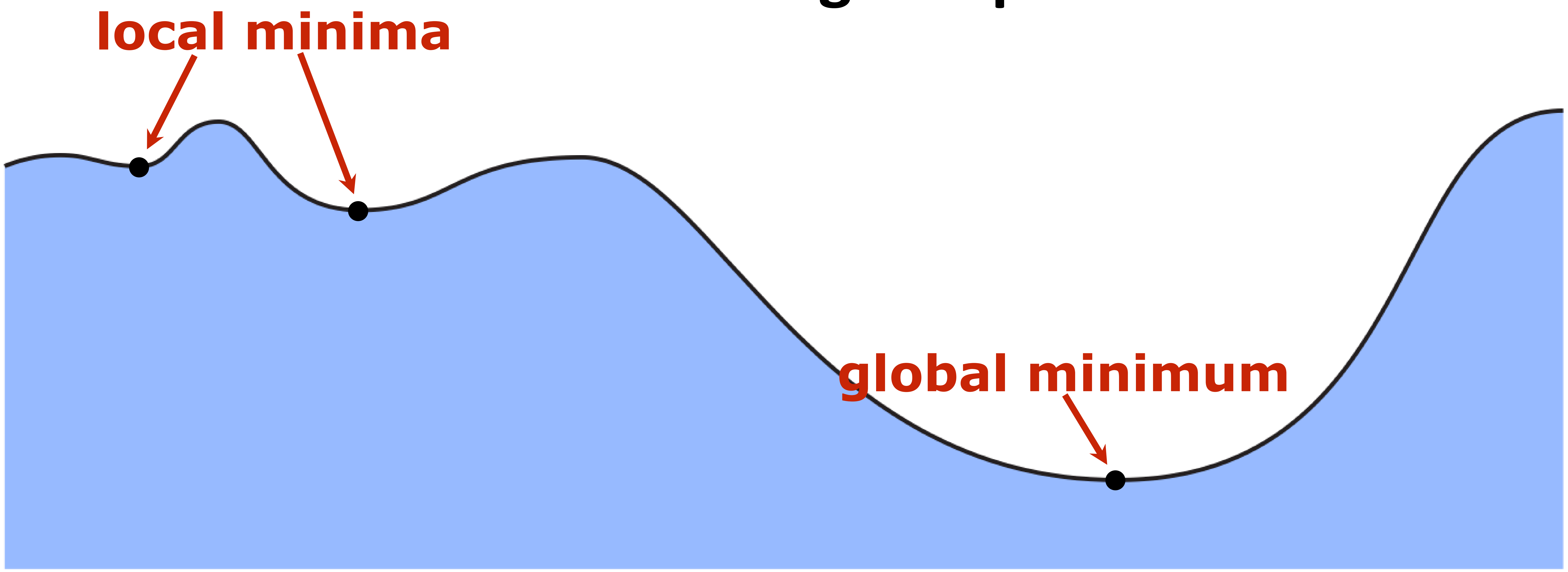
- **Even being bounded from below is not enough:**



- **No matter how big x is, we never achieve the lower bound (0)**
- **So when does a solution exist? Two *sufficient* conditions:**
- ***Extreme value theorem:* continuous objective & compact domain**
- ***Coercivity:* objective goes to +∞ as we travel (far) in any direction**

Characterization of Minimizers

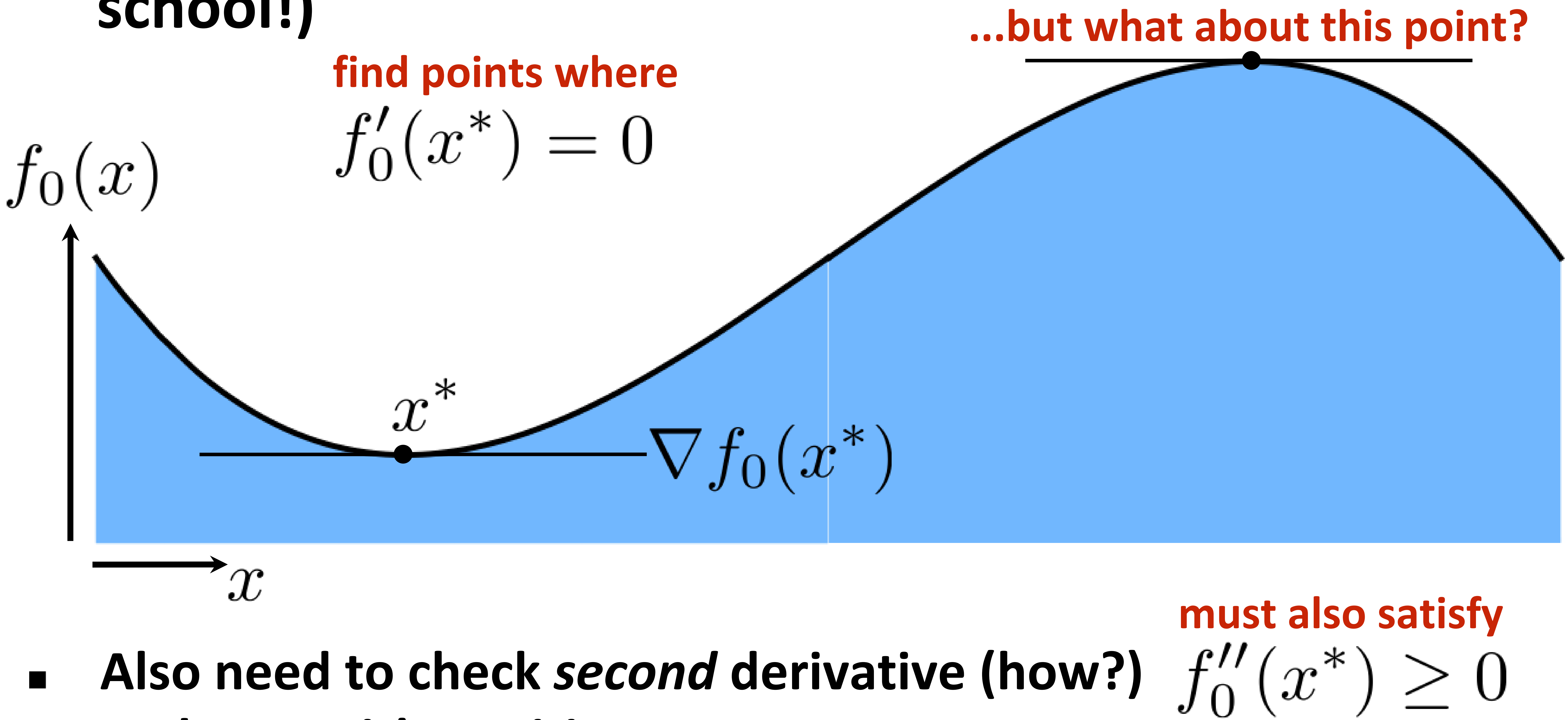
- Ok, so we have some sense of when a minimizer might *exist*
- But how do we know a given point x is a minimizer?



- Checking if a point is a global minimizer is (generally) hard
- But we can certainly test if a point is a local minimum (ideas?)
- (Note: a global minimum is also a local minimum!)

Characterization of Local Minima

- Consider an objective f_0: R → R. How do you find a minimum?
- (Hint: you may have memorized this formula in high school!)



- Also need to check *second* derivative (how?)
- Make sure it's *positive*
- Ok, but what does this all mean for more general functions f_0?

Optimality Conditions (Unconstrained)

- **In general, our objective is f0: $R^n \rightarrow R$**
- **How do we test for a local minimum?**
- **1st derivative becomes *gradient*; 2nd derivative becomes *Hessian***

$$\nabla f := \begin{bmatrix} \partial f / \partial x_1 \\ \vdots \\ \partial f / \partial x_n \end{bmatrix}$$

GRADIENT
(measures "slope")

$$\nabla^2 f := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} & \cdots & \frac{\partial f}{\partial x_1 \partial x_n} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} & \cdots & \frac{\partial f}{\partial x_2 \partial x_n} \\ \vdots & \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \frac{\partial^2 f}{\partial x_n \partial x_2} & \cdots & \frac{\partial f}{\partial x_n^2} \end{bmatrix}$$

HESSIAN
(measures "curvature")

- **Optimality conditions?**

positive semidefinite (PSD)
($u^T A u \geq 0$ for all u)

$$\nabla f_0(x^*) = 0$$

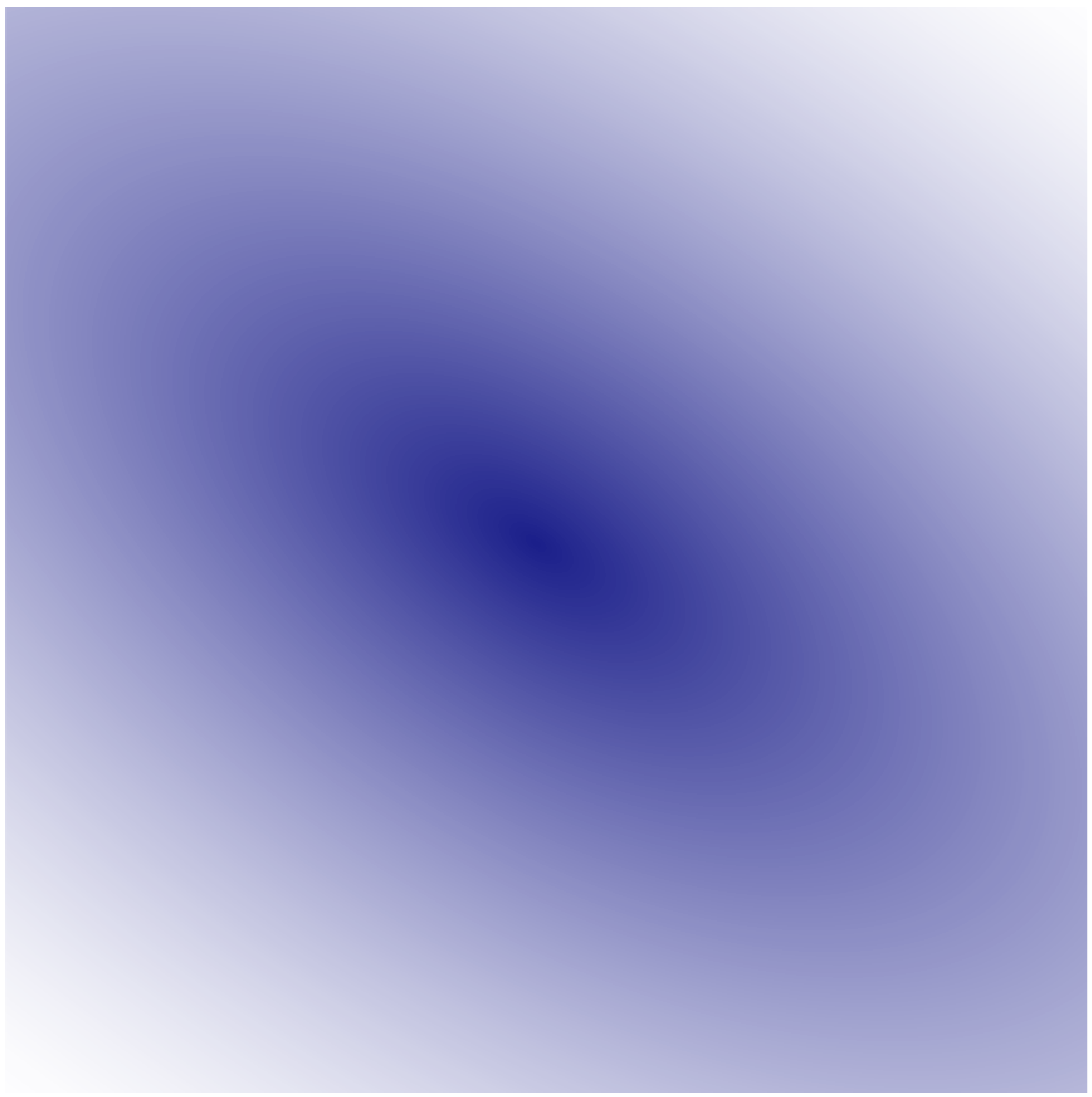
1st order

$$\nabla^2 f_0(x^*) \succeq 0$$

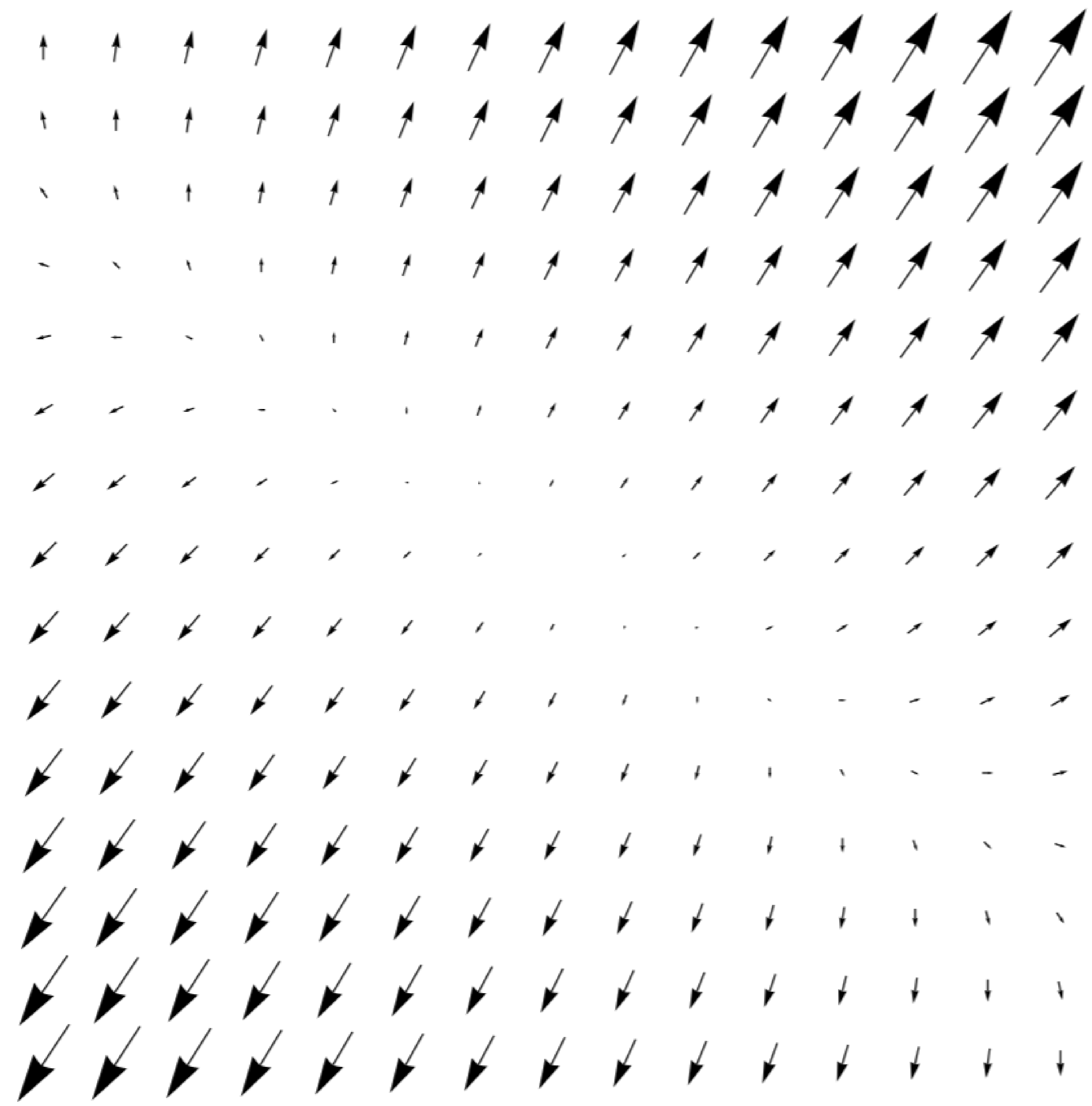
2nd order

Gradient

- **Given a multivariate function, its gradient assigns a vector at each point**



$f(\mathbf{x})$



$\nabla f(\mathbf{x})$

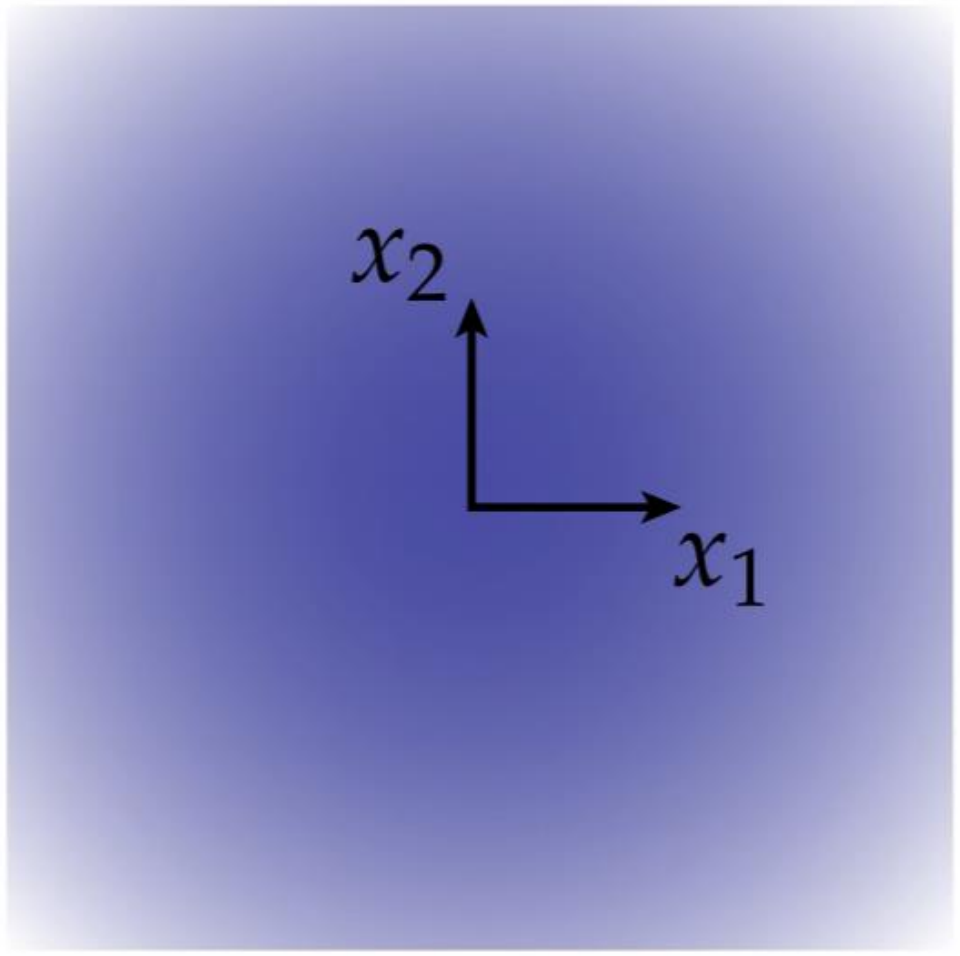
Gradient example

$$f(\mathbf{x}) := x_1^2 + x_2^2$$

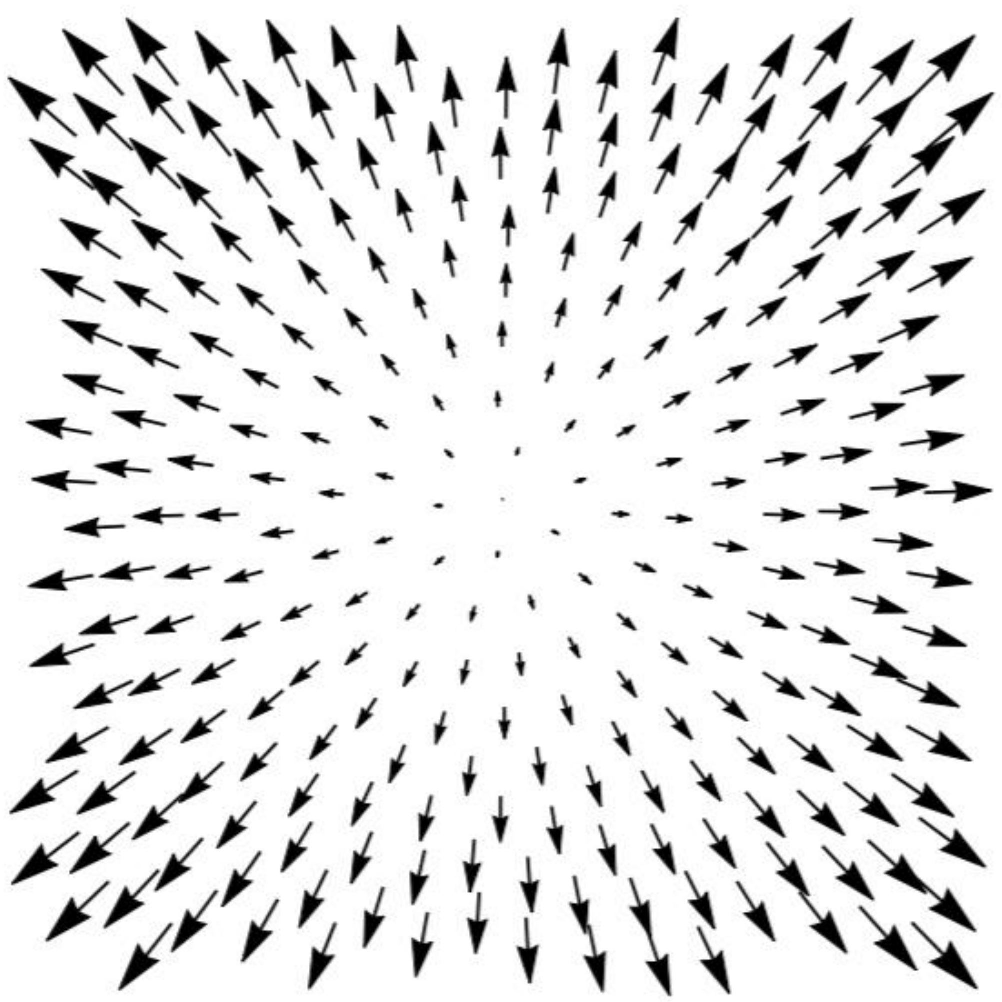
$$\frac{\partial f}{\partial x_1} = \frac{\partial}{\partial x_1} x_1^2 + \frac{\partial}{\partial x_1} x_2^2 = 2x_1 + 0$$

$$\frac{\partial f}{\partial x_2} = \frac{\partial}{\partial x_2} x_1^2 + \frac{\partial}{\partial x_2} x_2^2 = 0 + 2x_2$$

$$\nabla f(\mathbf{x}) = \begin{bmatrix} 2x_1 \\ 2x_2 \end{bmatrix} = 2\mathbf{x}$$



$f(\mathbf{x})$



$\nabla f(\mathbf{x})$

Gradients of Matrix-Valued Expressions

- **EXTREMELY useful to be able to differentiate matrix-valued expressions!**

For any two vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^n$ and **symmetric** matrix $\mathbf{A} \in \mathbb{R}^{n \times n}$:

Matrix Derivative	Looks Like
$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{y}) = \mathbf{y}$	$\frac{d}{dx}xy = y$
$\nabla_{\mathbf{x}}(\mathbf{x}^T\mathbf{x}) = 2\mathbf{x}$	$\frac{d}{dx}x^2 = 2x$
$\nabla_{\mathbf{x}}(\mathbf{x}^TA\mathbf{y}) = A\mathbf{y}$	$\frac{d}{dx}axy = ay$
$\nabla_{\mathbf{x}}(\mathbf{x}^TA\mathbf{x}) = 2A\mathbf{x}$	$\frac{d}{dx}ax^2 = 2ax$
...	...

Excellent resource: Petersen & Pedersen, "The Matrix Cookbook"

- **At least once in your life, work these out meticulously in coordinates!**

Hessian

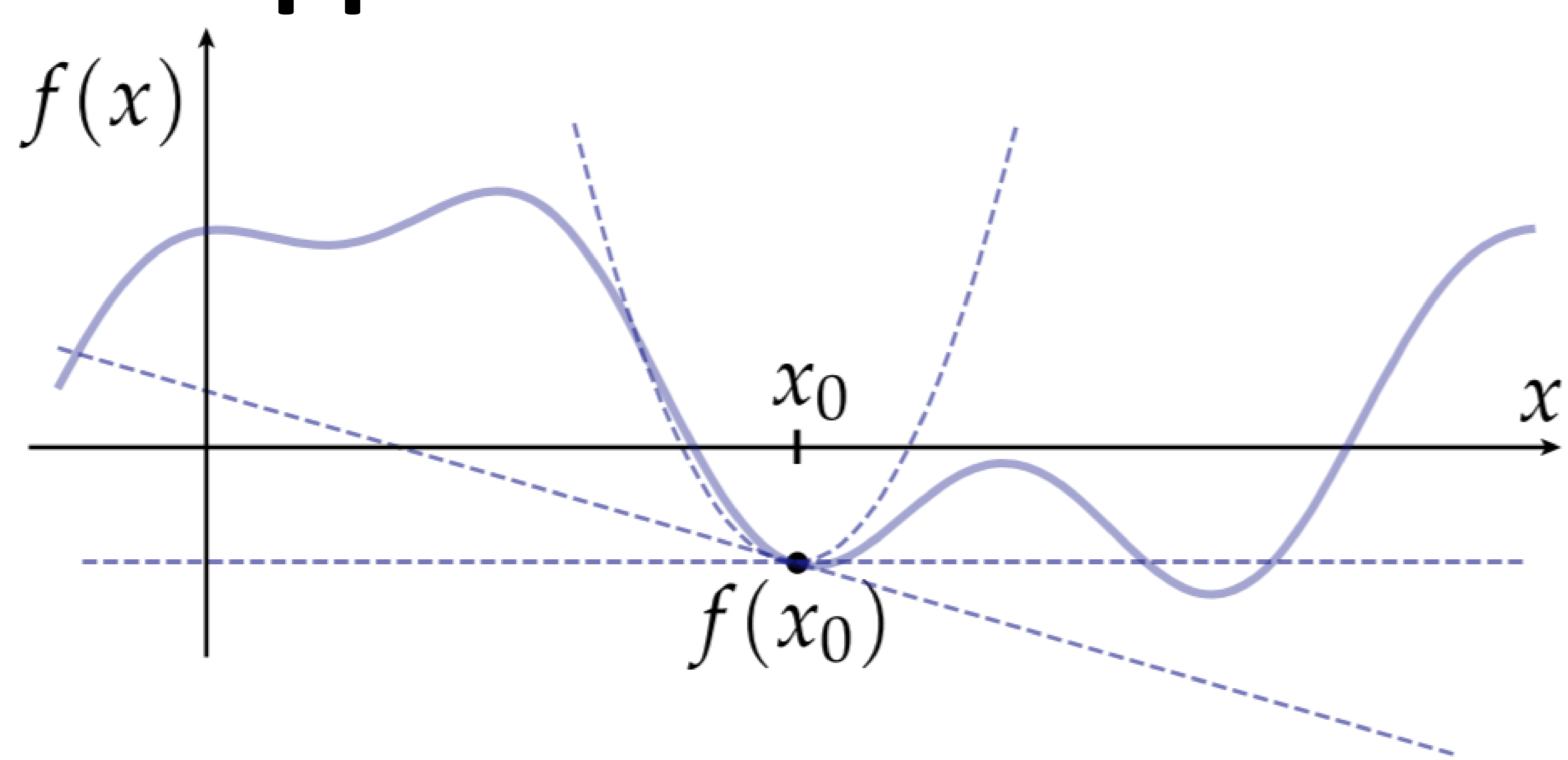
- **Jacobian of the gradient (matrix of second derivatives)**

$$\nabla^2 f := \begin{bmatrix} \frac{\partial^2 f}{\partial x_1 \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_1 \partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2 f}{\partial x_n \partial x_1} & \cdots & \frac{\partial^2 f}{\partial x_n \partial x_n} \end{bmatrix}$$

- **Recall Taylor series**

$$f(x) = f(x_0) + f'(x_0)(x - x_0) + \frac{(x-x_0)^2}{2!} f''(x_0) + \cdots$$

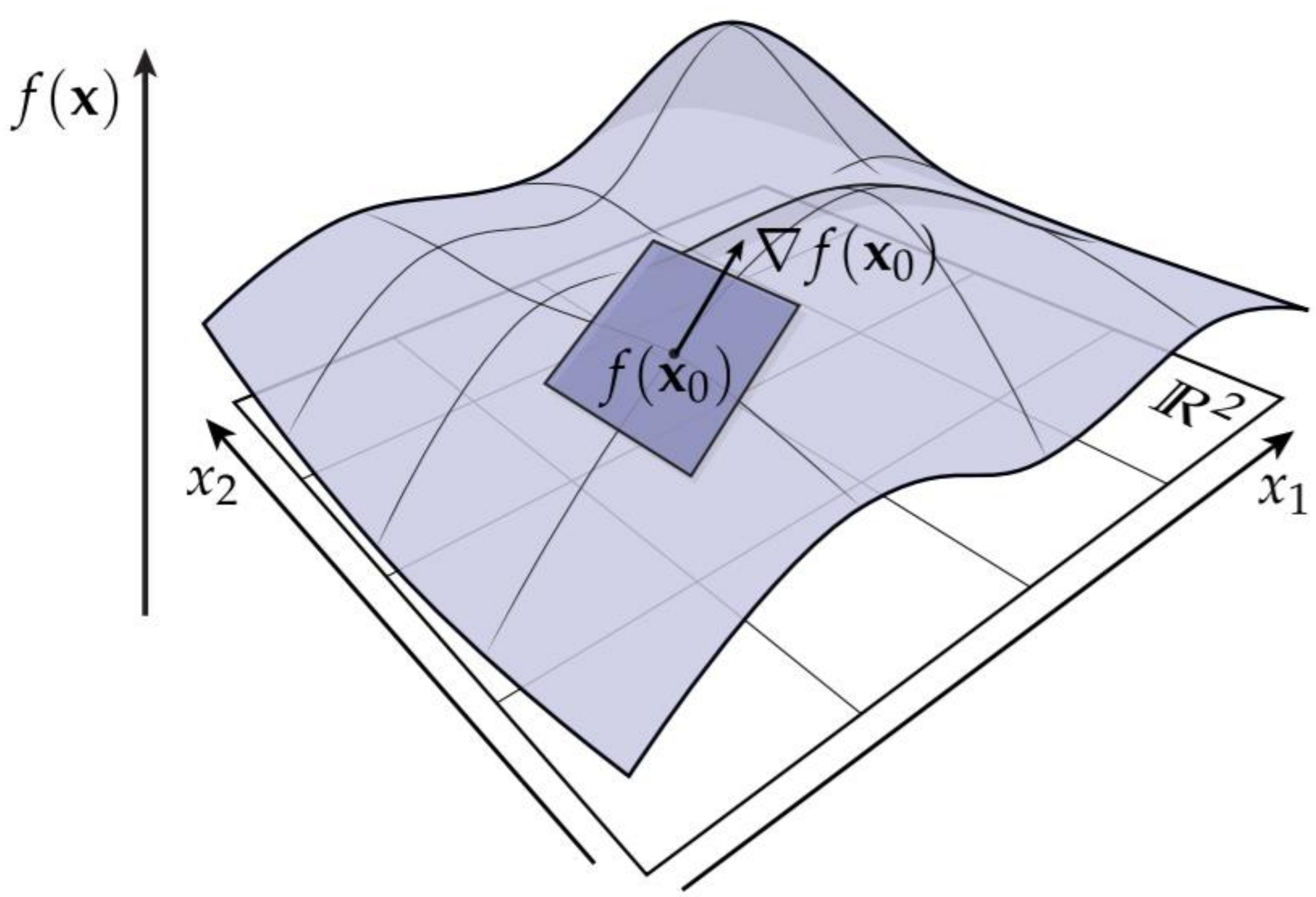
- **Gradient gives best linear approximation**
- **Hessian gives us quadratic approximation**



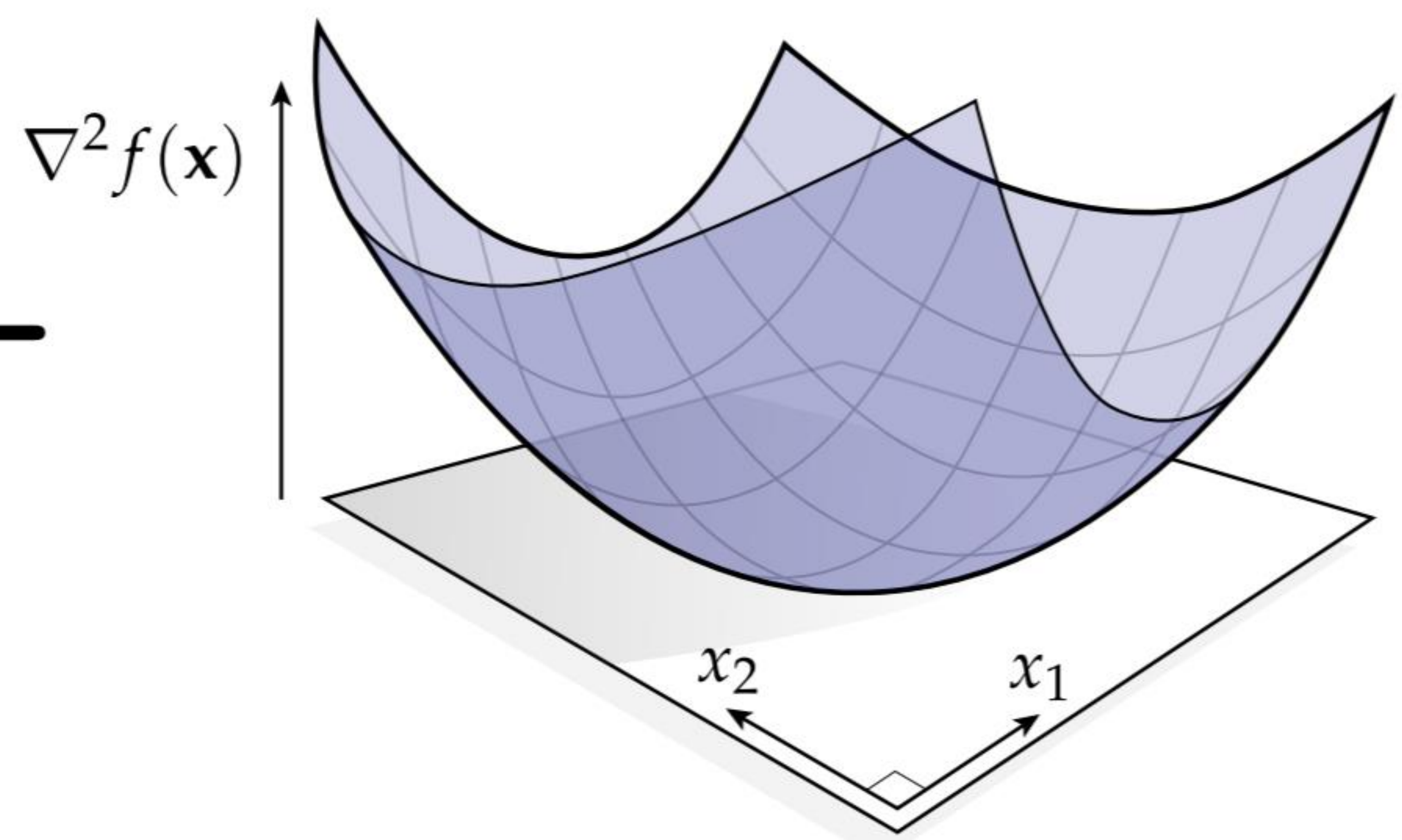
Hessian

- **Or in higher dimensions**
- **Taylor series...**

$$\underbrace{}_{\text{}}f(\mathbf{x}) \approx \underbrace{f(\mathbf{x}_0)}_{c\in\mathbb{R}}^{\text{constant}} + \langle \underbrace{\nabla f(\mathbf{x}_0)}_{\mathbf{b}\in\mathbb{R}^n}^{\text{linear}}, \mathbf{x}-\mathbf{x}_0\rangle + \langle \underbrace{\nabla^2 f(x_0)}_{\mathbf{A}\in\mathbb{R}^{n\times n}}^{\text{quadratic}}(\mathbf{x}-\mathbf{x}_0), \mathbf{x}-\mathbf{x}_0\rangle/2$$



+



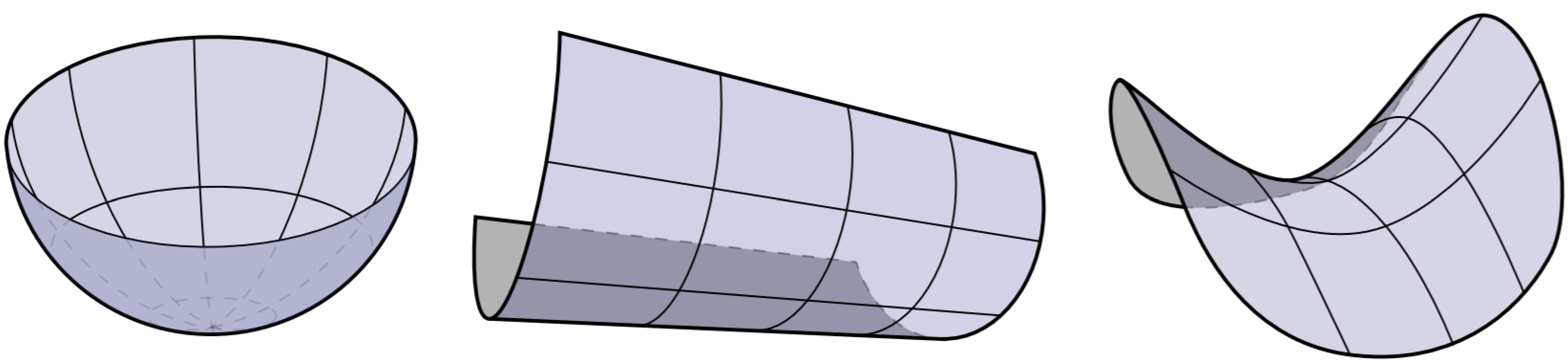
Hessian and Optimality conditions

- **Optimality conditions for multivariate optimization?**

positive semidefinite (PSD)
($u^T A u \geq 0$ for all u)

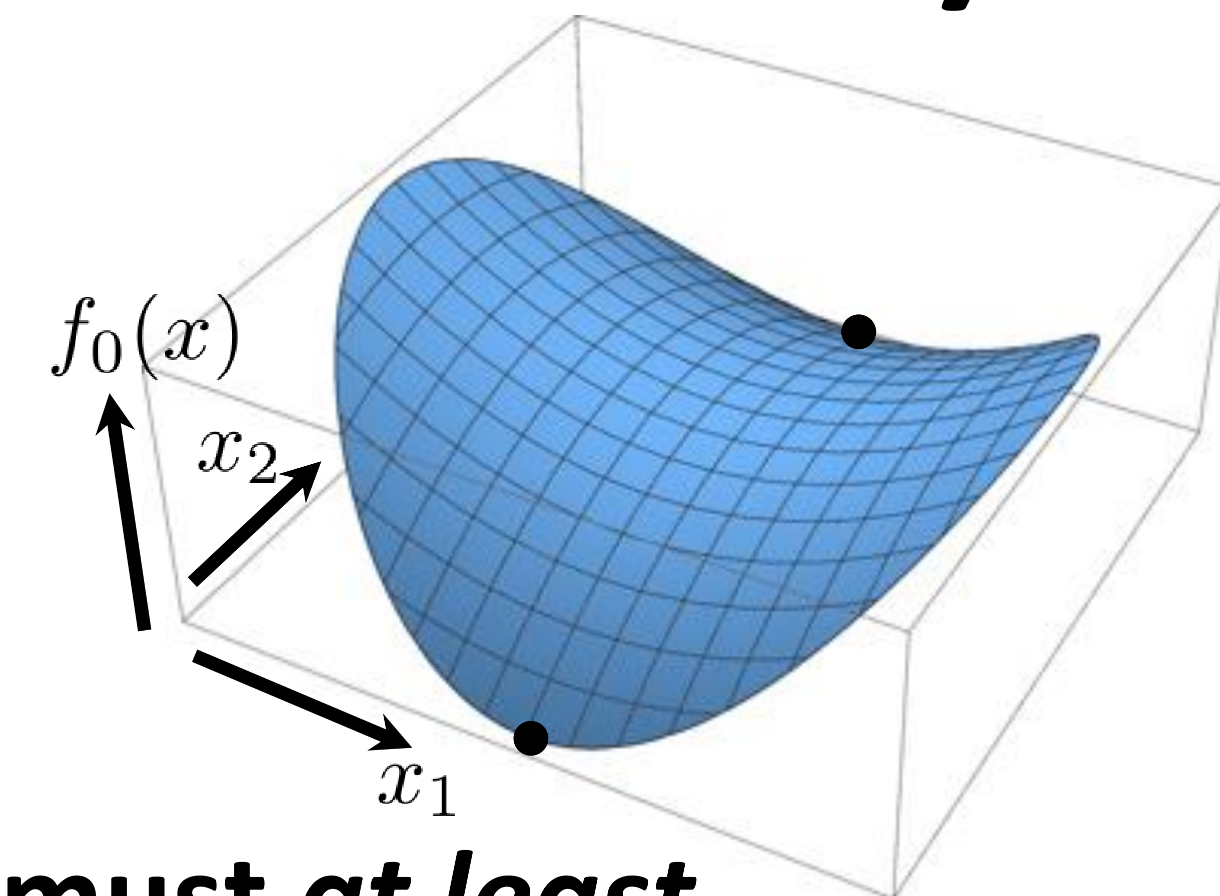
$$\nabla f_0(x^*) = 0 \qquad \nabla^2 f_0(x^*) \succeq 0$$

1st order — 2nd order



positive definite — positive semidefinite — indefinite

Optimality Conditions (Constrained)



- What if we have constraints?
- Is gradient at minimizer still zero?
- Is Hessian at minimizer still PSD?
- Not necessarily! (See example above)
- In general, any (local or global) minimizer must *at least* satisfy the *Karush–Kuhn–Tucker (KKT)* conditions:

$$\exists \lambda_i \text{ s.t.} \qquad \nabla f_0(x^*) = -\sum_{i=1}^{n} \lambda_i \nabla f_i(x^*) \quad \text{stationarity}$$

$$f_i(x^*) \leq 0, \; i = 1, \ldots, n \quad \text{primal feasibility}$$

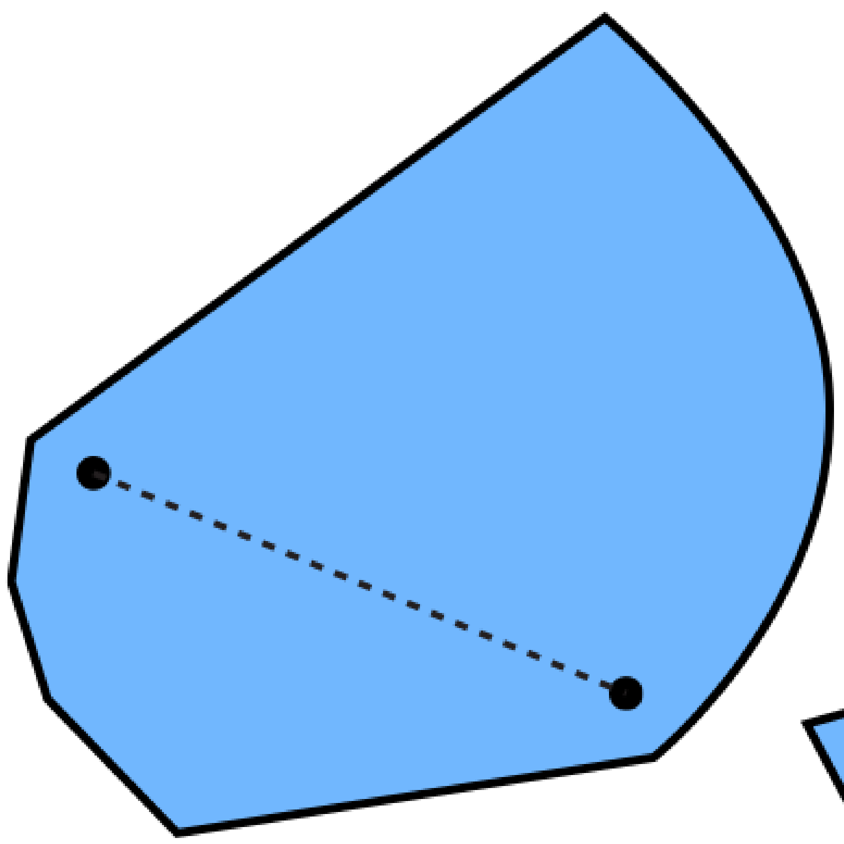
$$\lambda_i \geq 0, i = 1, \ldots, n \quad \text{dual feasibility}$$

$$\lambda_i f_i(x^*) = 0, \; i = 1, \ldots, n \quad \text{complementary slackness}$$

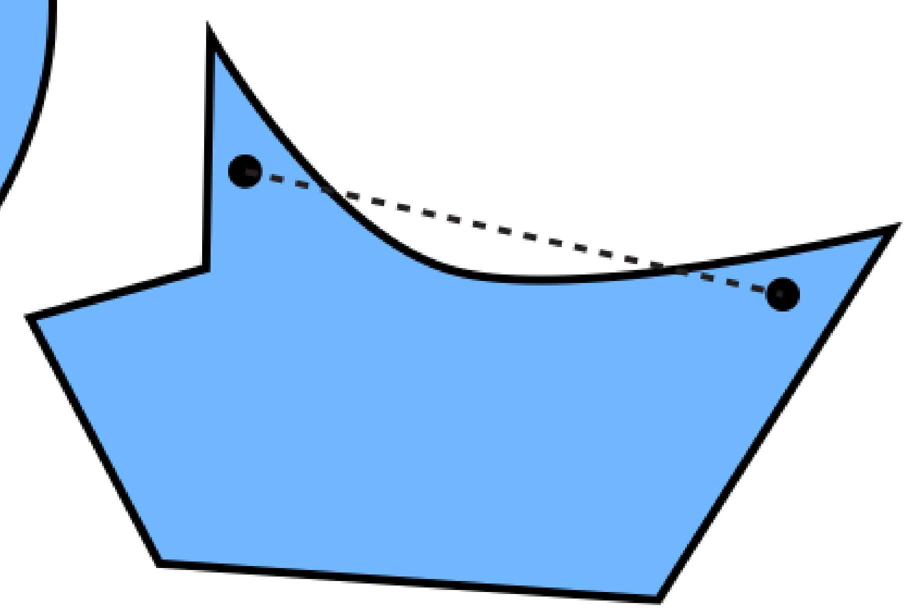
- ...we won't work with these in this class! (But good to get some intuition into how it works and to know where to look.)

Convex Optimization

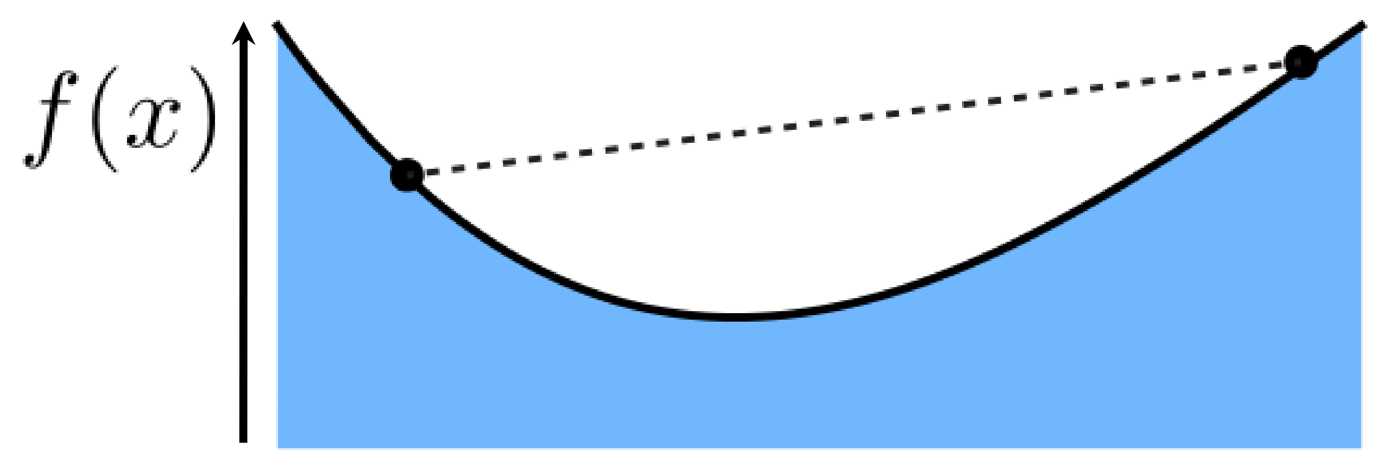
- Special class of problems that are almost always "easy" to solve (polynomial-time!)
- Problem is *convex* if it has a convex domain *and* convex objective



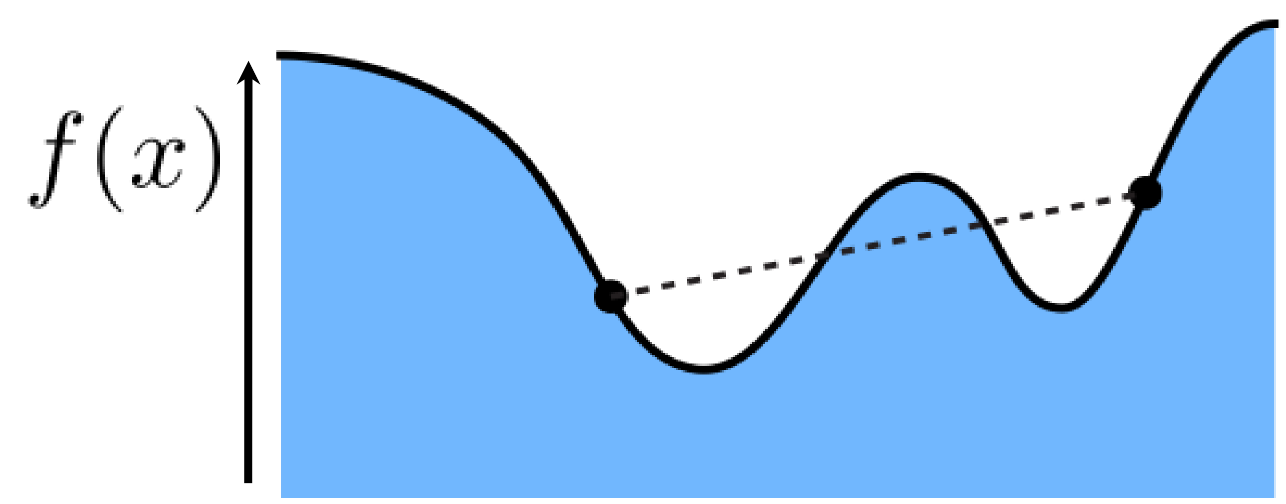
convex domain



noconvex domain



convex objective



nonconvex objective

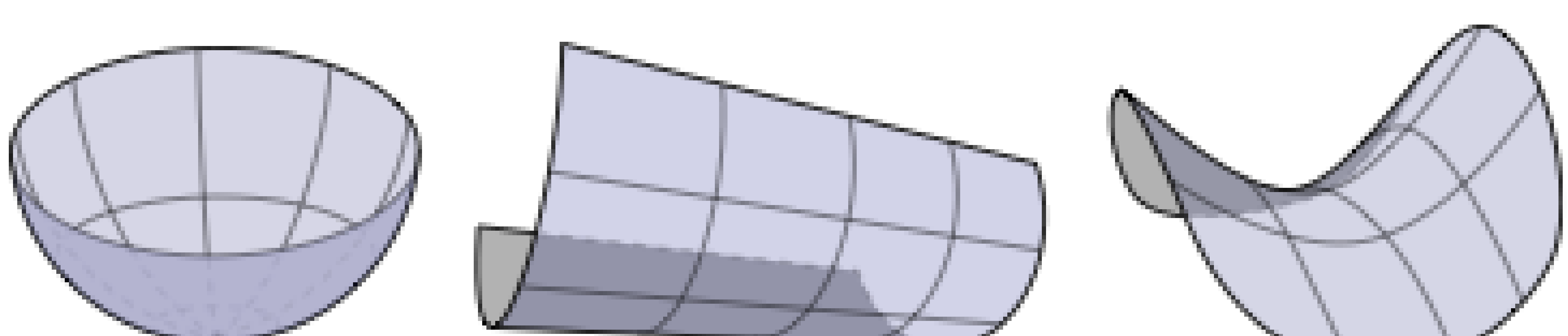
- Why care about convex problems in graphics?
  - can make guarantees about solution (always the best)
  - doesn't depend on initialization (strong convexity)
  - often quite efficient

Convex Quadratic Objectives & Linear Systems

- Very important example: convex *quadratic* objective
- Already saw this with, e.g., quadric error simplification
- Valuable *"variational"* way of looking at many common equations
- Can be expressed via positive-semidefinite (PSD) matrix:

$$f_0(x) = \frac{1}{2}x^T A x - x^T b, \;\; A \succeq 0$$

  - Q: 1st-order optimality condition? $Ax = b$ ← just solve a linear system!
  - Q: 2nd-order optimality condition? $A \succeq 0$ ← satisfied by definition



positive definite | positive semidefinite | indefinite

Sadly, life is not usually that easy.

How do we solve optimization problems in general?

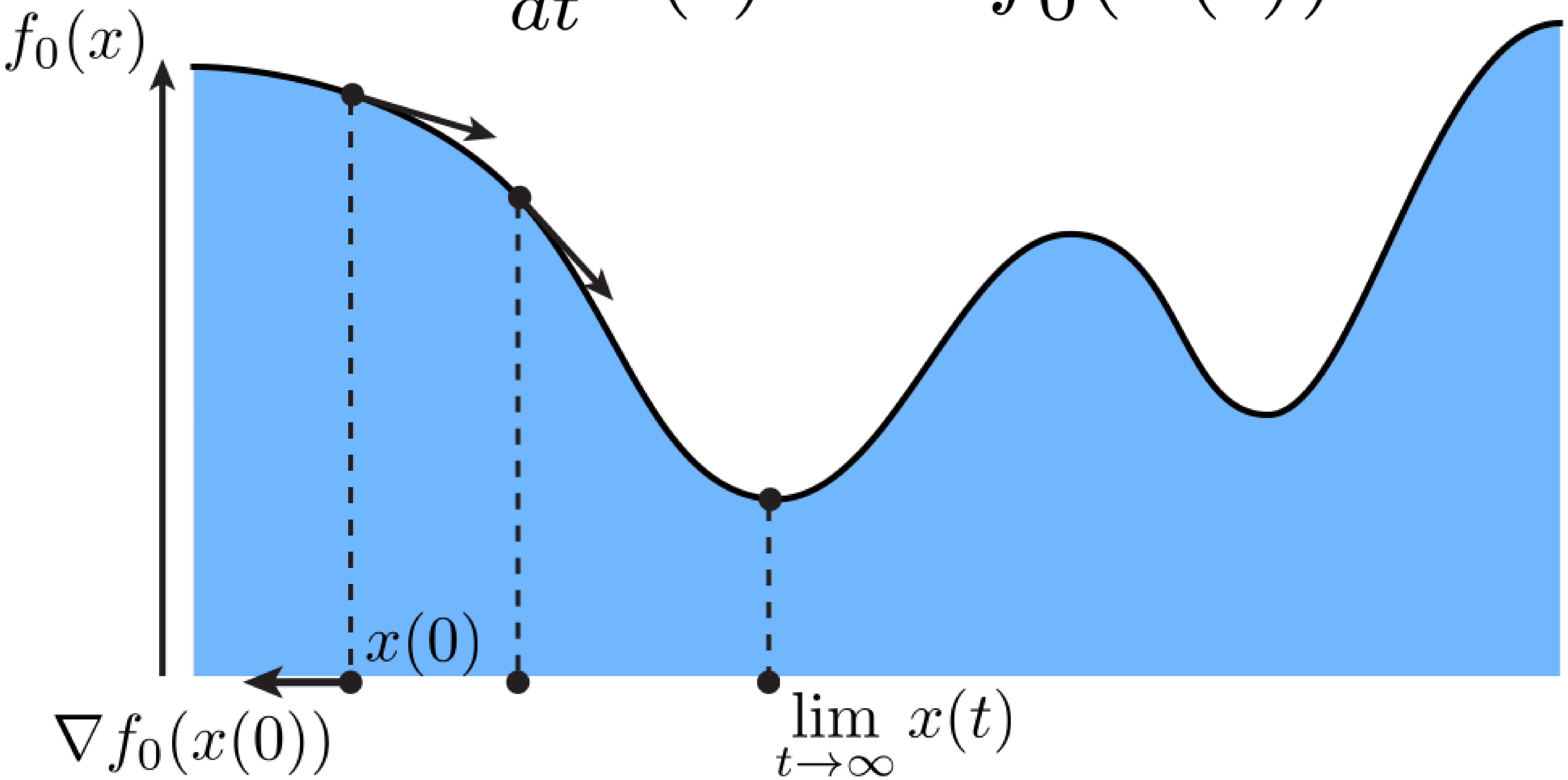
Descent Methods

An idea as old as the hills:

Gradient Descent (1D)

- **Basic idea: follow the gradient "downhill" until it's zero**
- **(Zero gradient was our 1st-order optimality condition)**

$$\frac{d}{dt}x(t) = -f_0'(x(t))$$

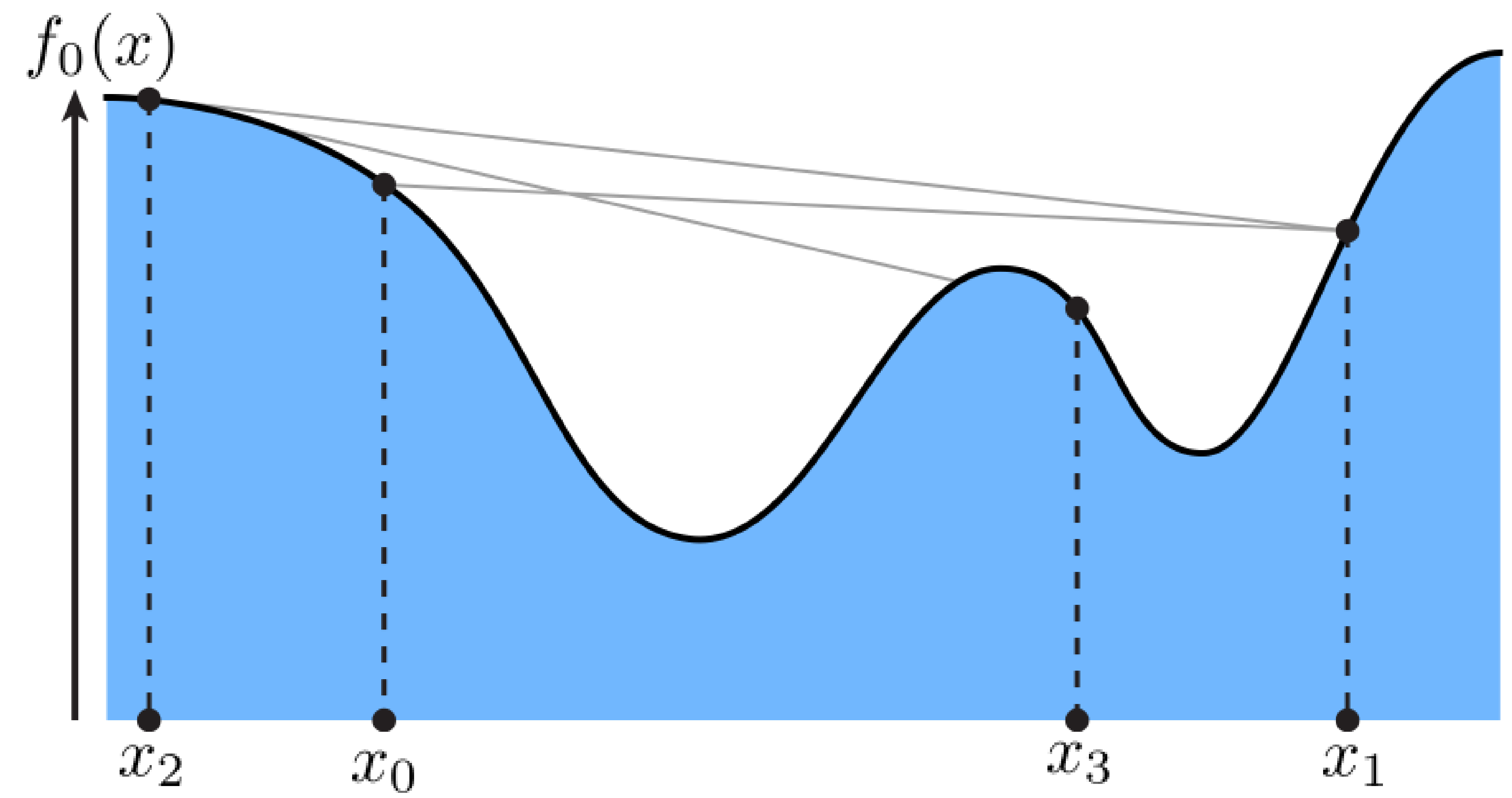


- **Do we always end up at a (global) minimum?**
- **How do we compute gradient descent in practice?**

Gradient Descent Algorithm (1D)

- **Simple update rule (go in direction that decreases objective):**

$$x_{k+1} = x_k - \tau f_0'(x_k)$$

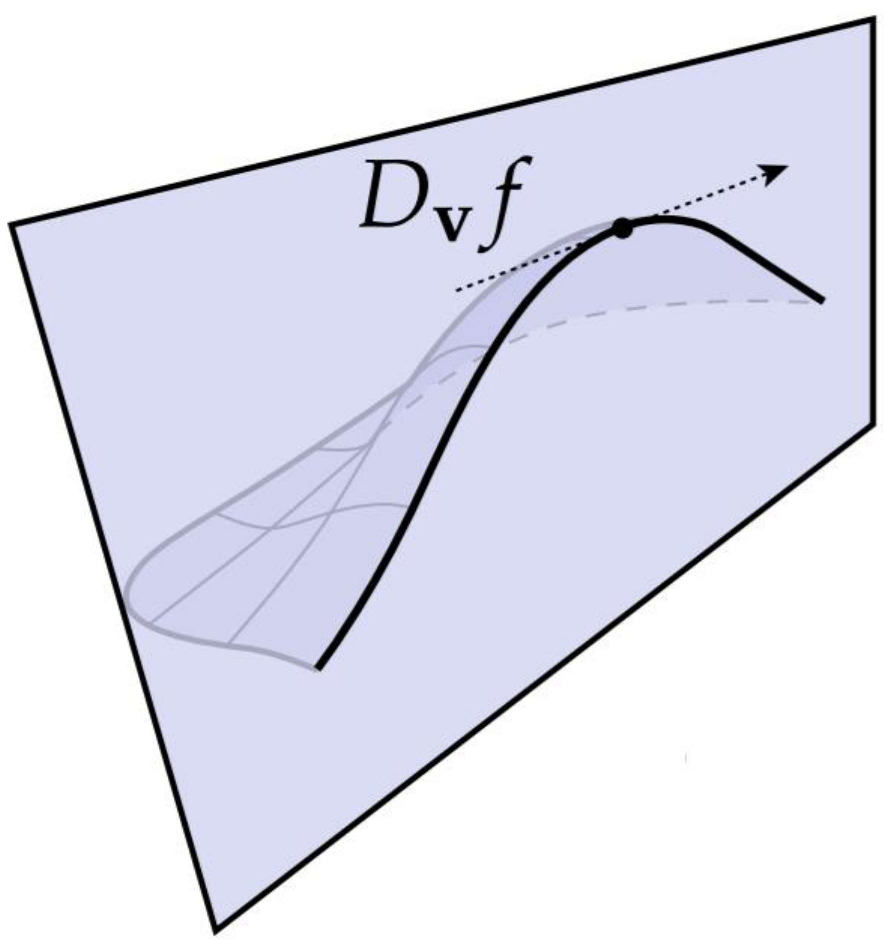


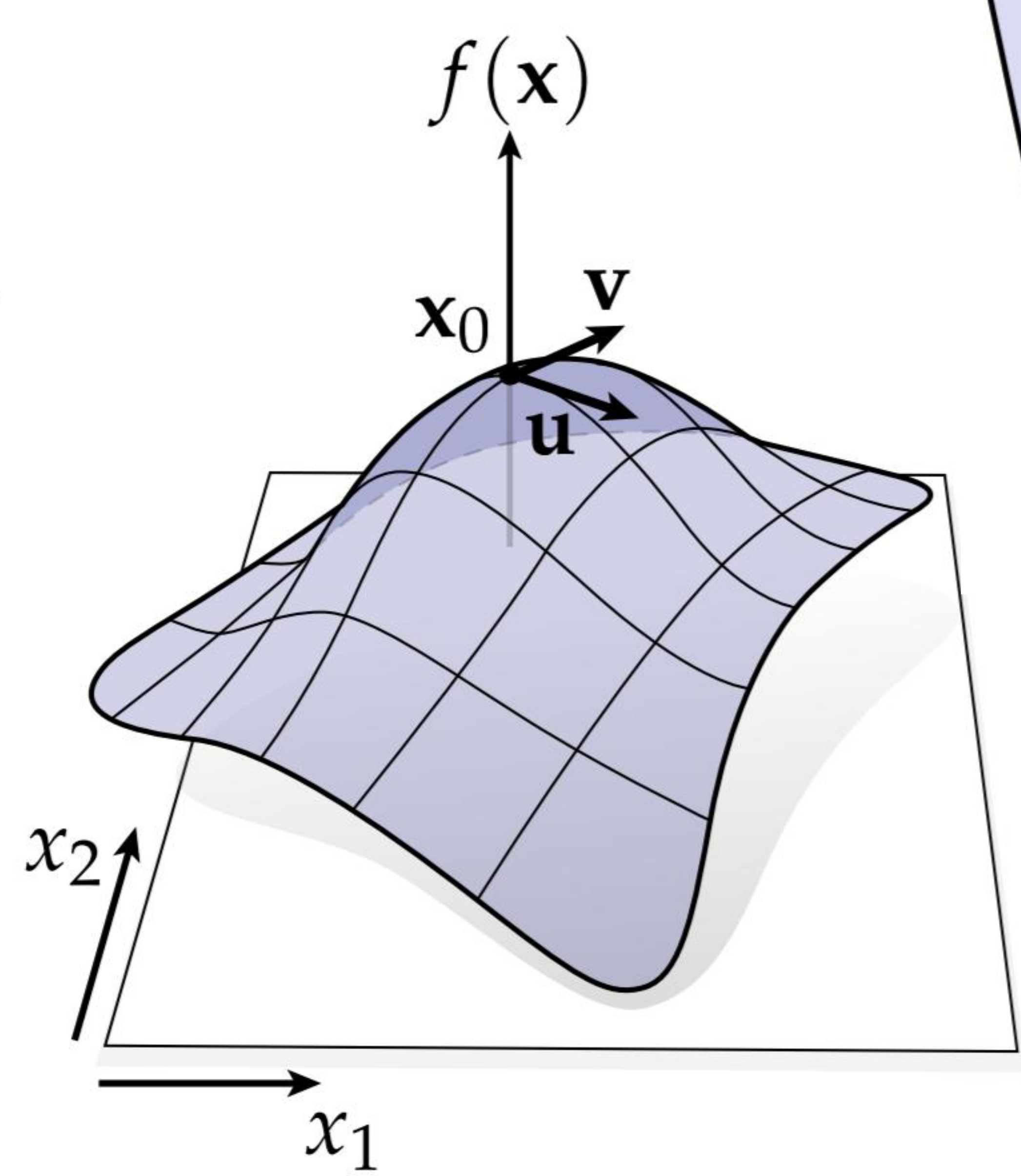
- **Q: How far should we go in that direction?**
- **If we're not careful, we'll be zipping all over the place!**
- **Basic idea: use *"step control"* to determine step size based on value of objective & derivatives.**
- **A careful strategy (e.g., Armijo-Wolfe) can guarantee convergence at least to a *local* minimum.**
- **For now we will do something simpler: make τ *really small!***

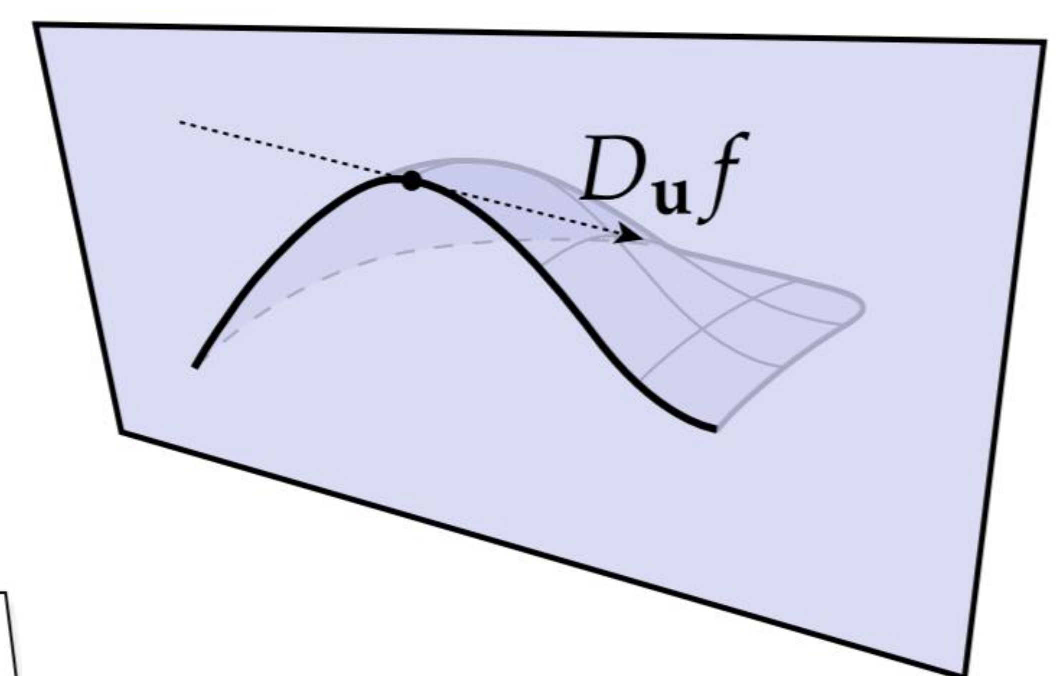
How do we go about optimizing a function of multiple variables?

Directional Derivative

- **Suppose we have a function f(x1, x2)**
  - **Take a slice through this function along some direction**
  - **Then apply the usual derivative concept!**
  - **This is called the directional derivative**







take a small step along u

$$D_{\mathbf{u}} f(\mathbf{x}_0) := \lim_{\varepsilon \to 0} \frac{f(\mathbf{x}_0 + \varepsilon \mathbf{u}) - f(\mathbf{x}_0)}{\varepsilon}$$

Directional Derivative

- **Starting from Taylor's series**

constant · linear · quadratic

$$f(\mathbf{x}) \approx \underbrace{f(\mathbf{x}_0)}_{c \in \mathbb{R}} + \langle \underbrace{\nabla f(\mathbf{x}_0)}_{\mathbf{b} \in \mathbb{R}^n}, \mathbf{x} - \mathbf{x}_0 \rangle + \langle \underbrace{\nabla^2 f(x_0)}_{\mathbf{A} \in \mathbb{R}^{n \times n}} (\mathbf{x} - \mathbf{x}_0), \mathbf{x} - \mathbf{x}_0 \rangle / 2$$

easy to show that

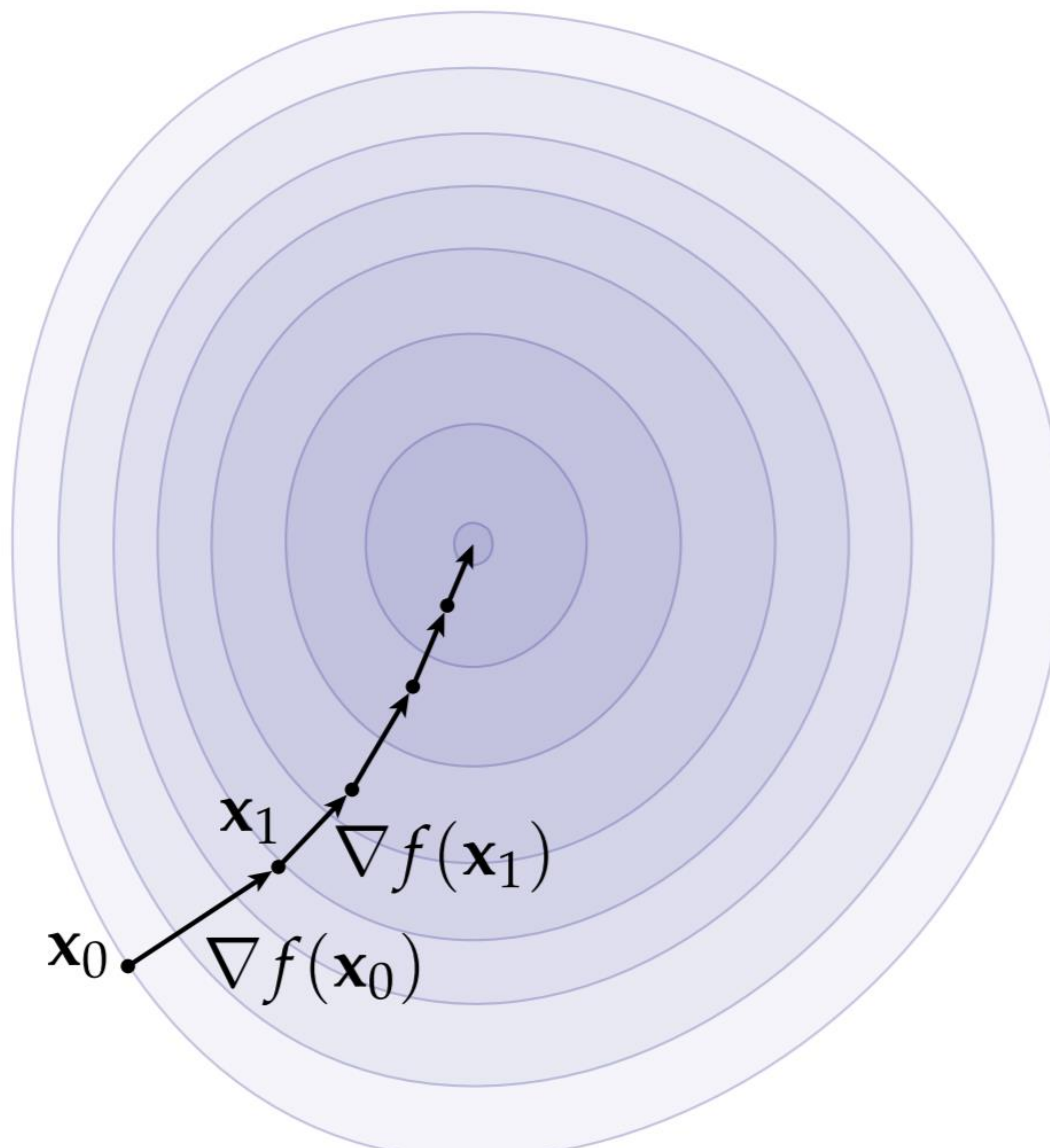
take a small step along u

$$D_{\mathbf{u}} f(\mathbf{x}_0) := \lim_{\varepsilon \to 0} \frac{f(\mathbf{x}_0 + \varepsilon \mathbf{u}) - f(\mathbf{x}_0)}{\varepsilon} = \langle \nabla f(x), \mathbf{u} \rangle$$

Q: What does this mean?

Gradient is direction of steepest ascent

- **Function value**
  - gets largest if we move in direction of gradient
  - doesn't change if we move orthogonally
  - decreases *fastest* if we move exactly in opposite direction

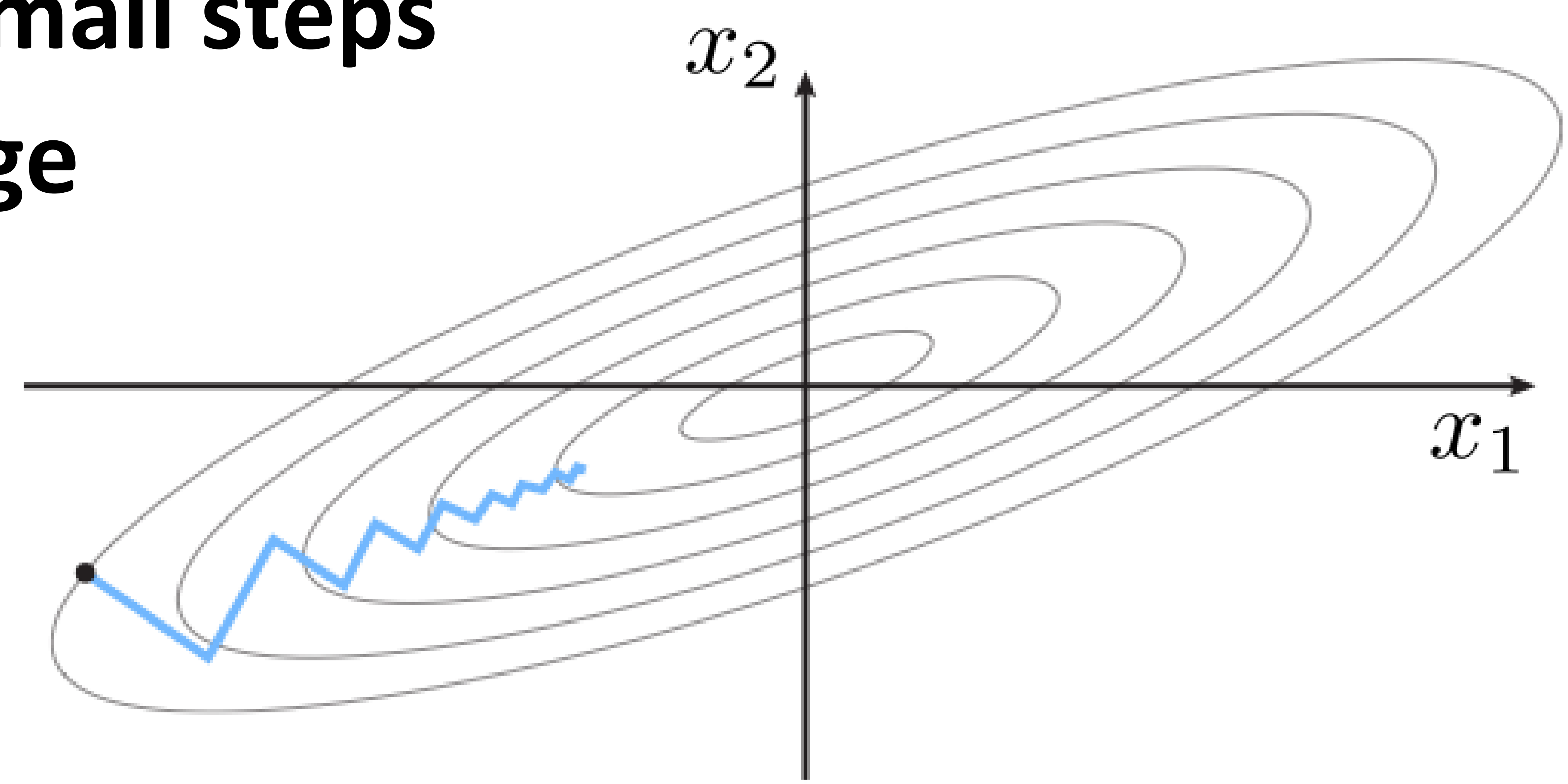


Gradient Descent Algorithm (nD)

Q: What's the corresponding update in higher dimensions?

$$x_{k+1} = x_k - \tau \nabla f_0(x_k)$$

- **Basic challenge in nD:**
  - **solution can "oscillate"**
  - **takes many, many small steps**
  - **very slow to converge**



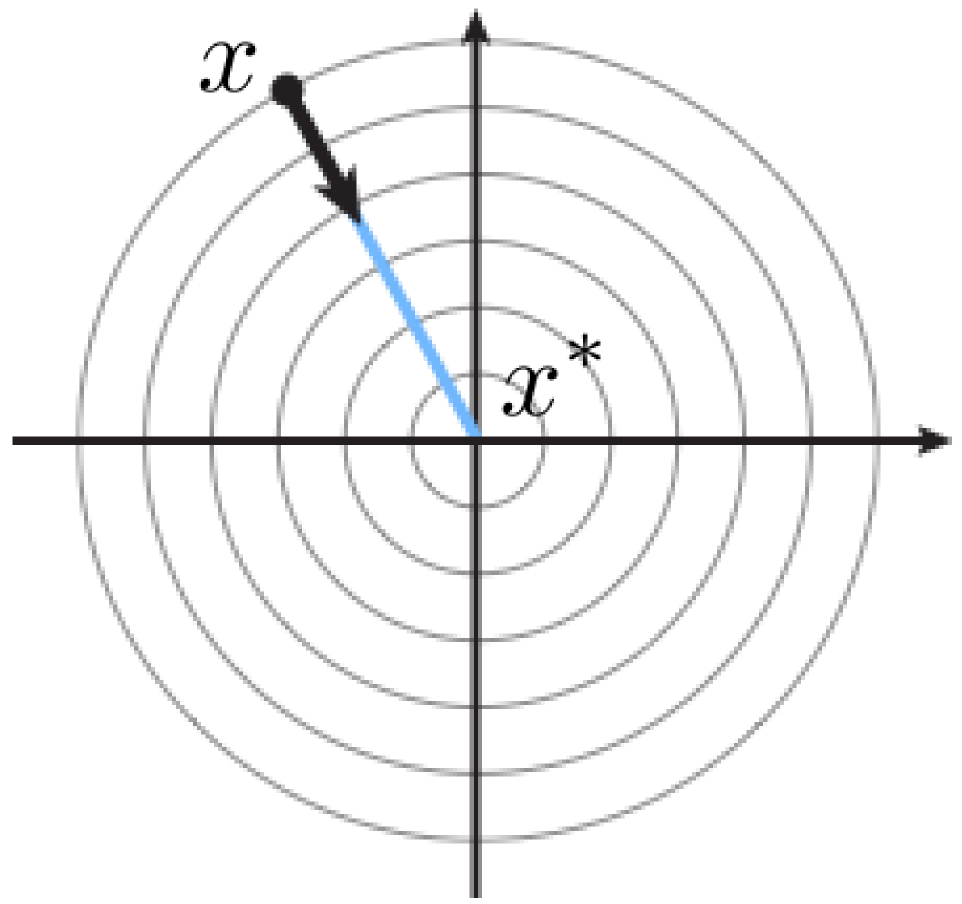
Higher Order Descent

- General idea: apply a coordinate transformation so that the local energy landscape looks more like a "round bowl"
- Gradient now points directly toward nearby minimizer
- Most basic strategy: Newton's method:

gradient

$$x_{k+1} = x_k - \tau(\nabla^2 f_0(x_k))^{-1} \nabla f_0(x_k)$$

Hessian inverse



- Great for convex problems (even proofs about # of steps!)
- For nonconvex problems, need to be more careful
- In general, nonconvex optimization is a *BLACK ART*
- Meta-strategy: try lots of solvers, see what works!
  - quasi-Newton, trust region, L-BFGS, ...

An example: optimization-based IK

- **Basic idea behind IK algorithm:**
  - **write down distance between final point and "target" and set up objective**
  - **compute gradient with respect to angles**
  - **apply gradient descent**

- **Objective?**

$$\boldsymbol{f_0(\theta) = \frac{1}{2}(p(\theta) - \tilde{p})^T(p(\theta) - \tilde{p})}$$

- **Constraints?**
  - **We could limit joint angles**