Metodi Numerici dell'Informatica

Discesa del gradiente

Emanuele Rodolà
rodola@di.uniroma1.it

SAPIENZA
UNIVERSITÀ DI ROMA

2nd semester a.y. 2022/2023 · May 2, 2023

Is there a general recipe to solve problems of this form, with f differentiable?

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

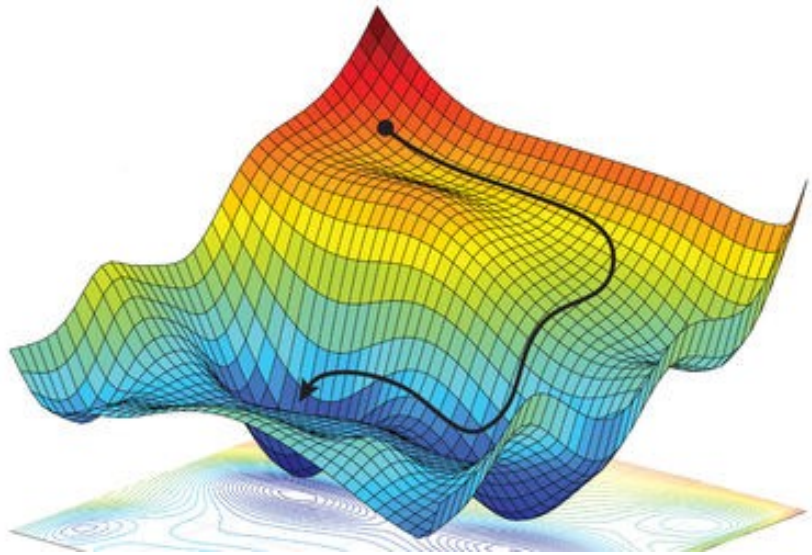
Gradient descent: Intuition

Gradient descent is a first-order iterative minimization algorithm.

Gradient descent: Intuition

Gradient descent is a first-order iterative minimization algorithm.

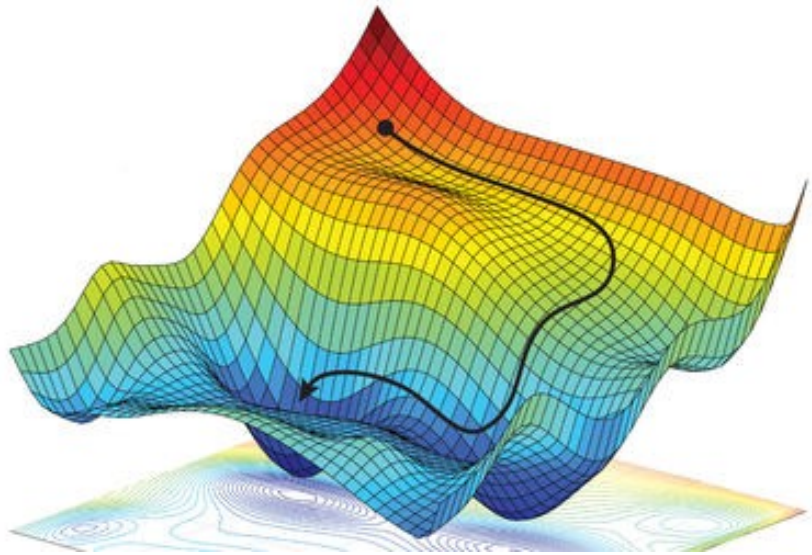
Example of an energy $f : \mathbb{R}^2 \to \mathbb{R}$:



Gradient descent: Intuition

Gradient descent is a first-order iterative minimization algorithm.

Example of an energy $f : \mathbb{R}^2 \to \mathbb{R}$:



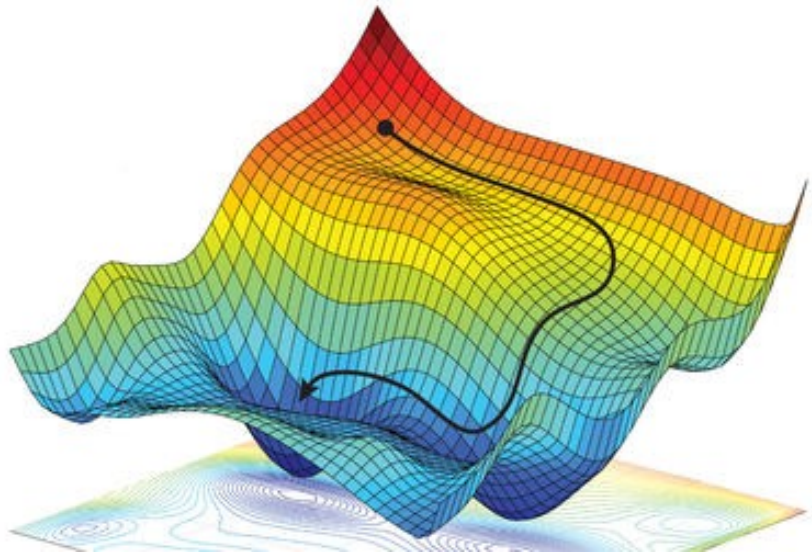
Overall idea: Move where the function decreases the most.

1. Start from some point $\mathbf{x}^{(0)} \in \mathbb{R}^2$.

Gradient descent: Intuition

Gradient descent is a first-order iterative minimization algorithm.

Example of an energy $f : \mathbb{R}^2 \to \mathbb{R}$:



Overall idea: Move where the function decreases the most.

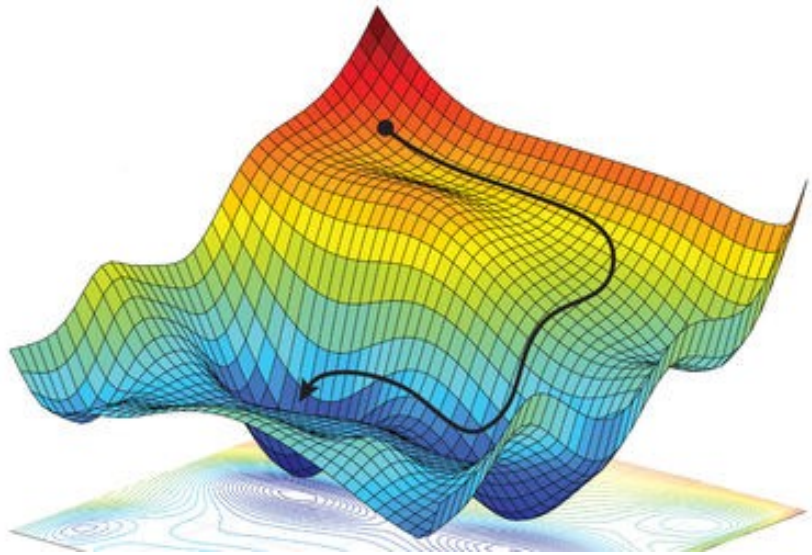
1. Start from some point $\mathbf{x}^{(0)} \in \mathbb{R}^2$.
2. Iteratively compute:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

Gradient descent: Intuition

Gradient descent is a first-order iterative minimization algorithm.

Example of an energy $f : \mathbb{R}^2 \to \mathbb{R}$:



Overall idea: Move where the function decreases the most.

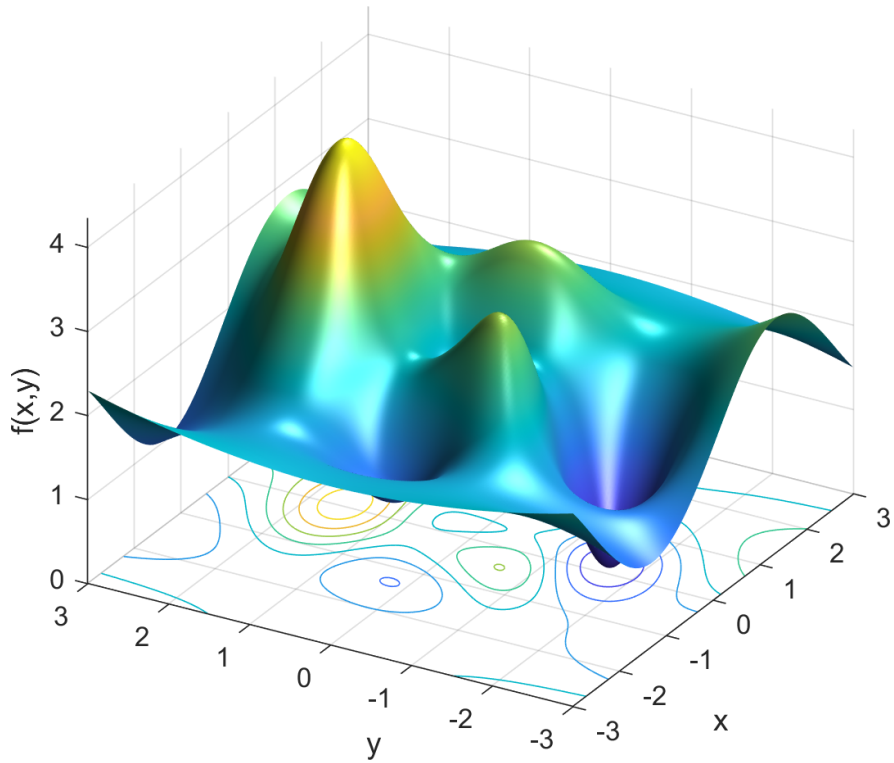
1. Start from some point $\mathbf{x}^{(0)} \in \mathbb{R}^2$.
2. Iteratively compute:
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$
3. Stop when a minimum is reached.

Gradient descent

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

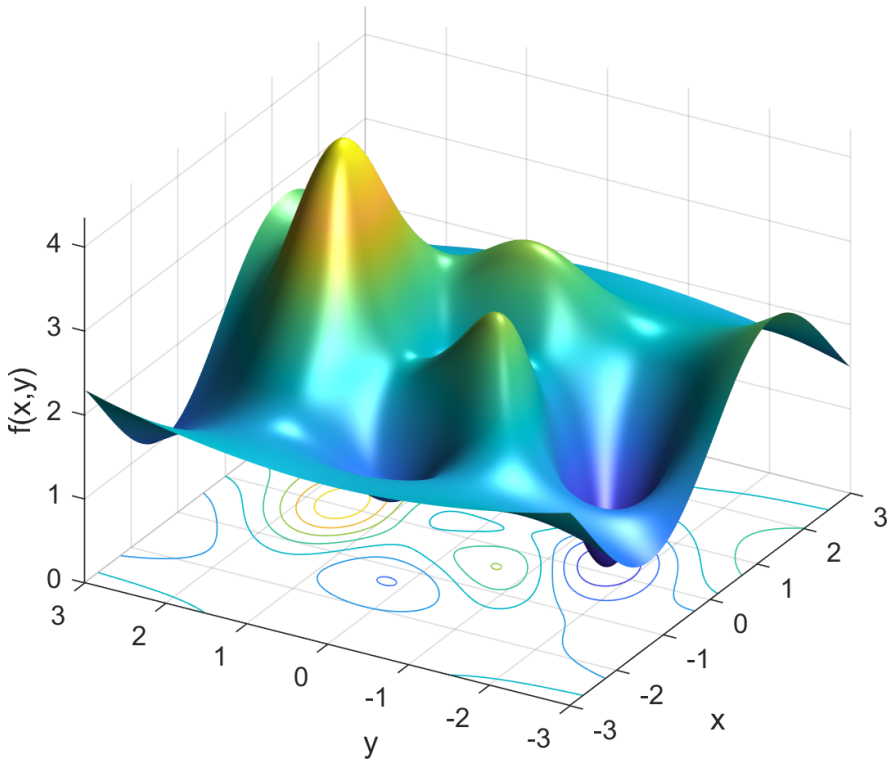
Gradient descent

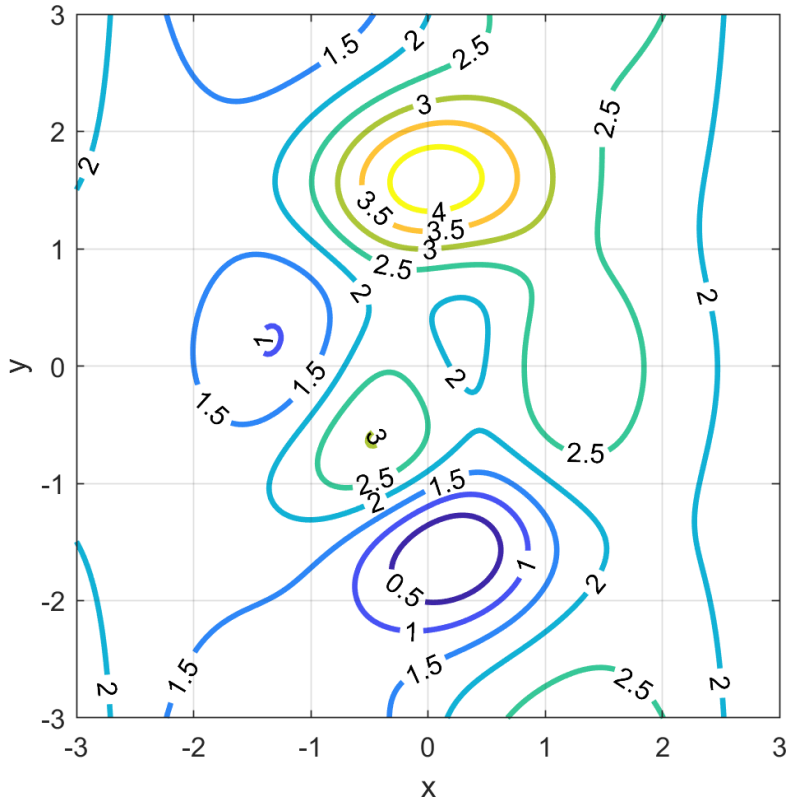
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$



Gradient descent

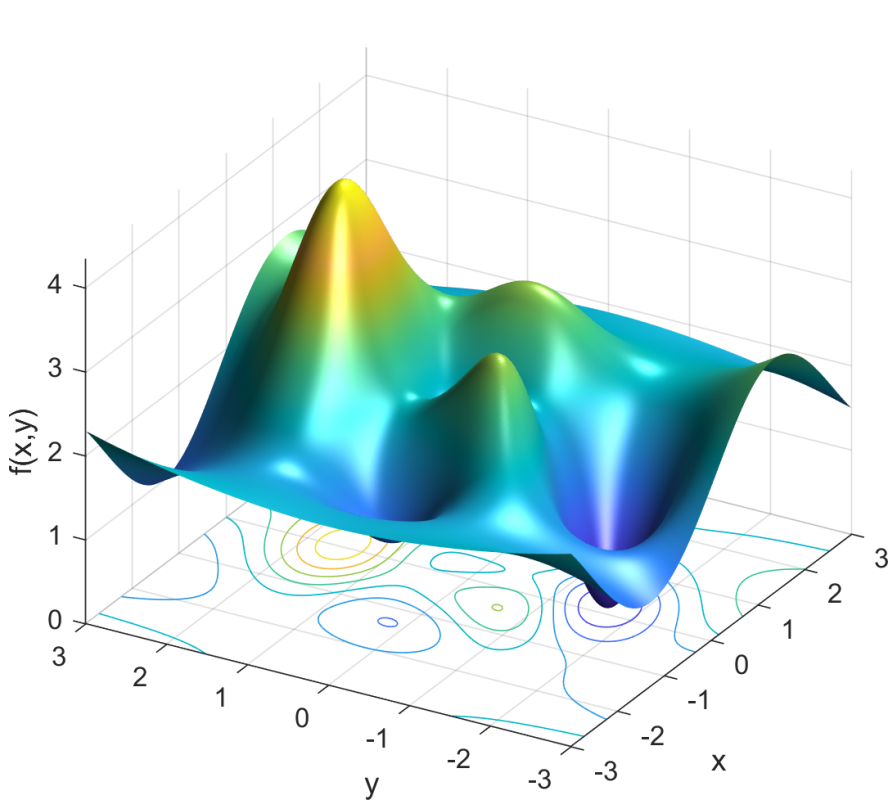
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

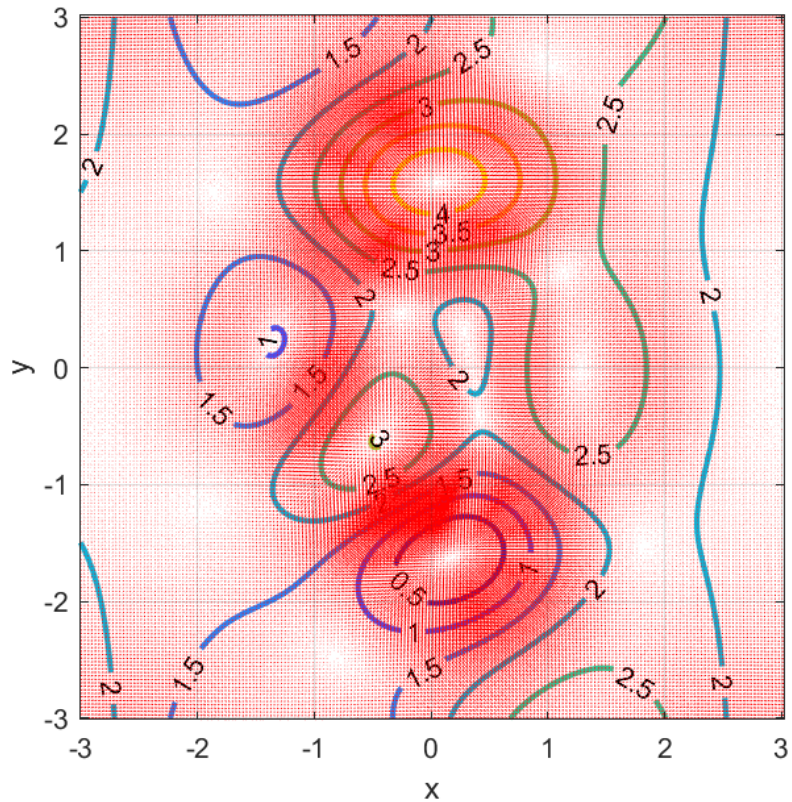




Gradient descent

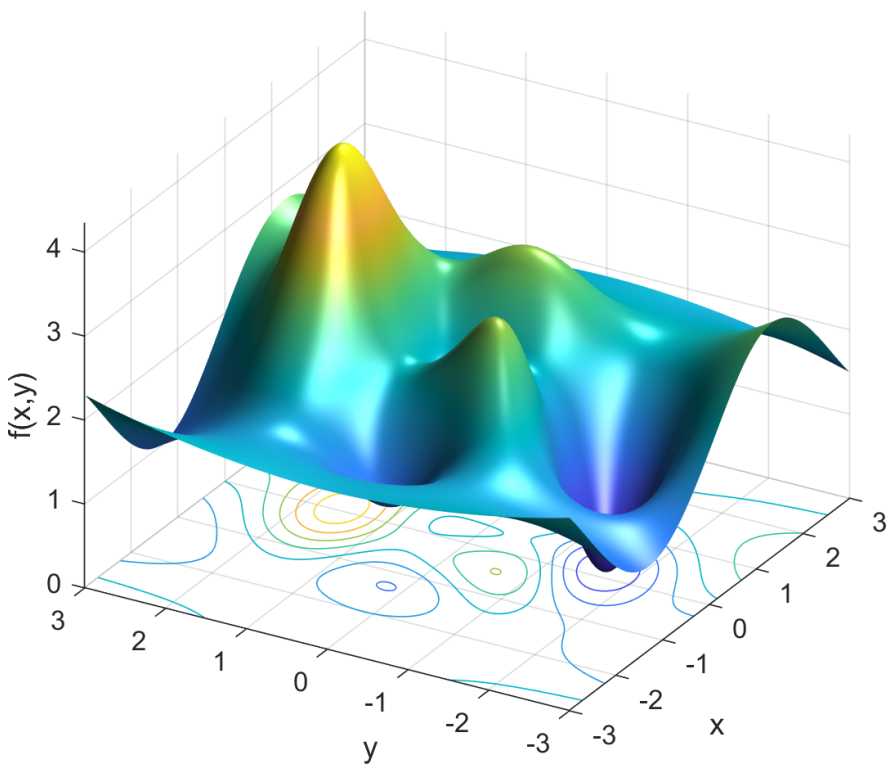
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

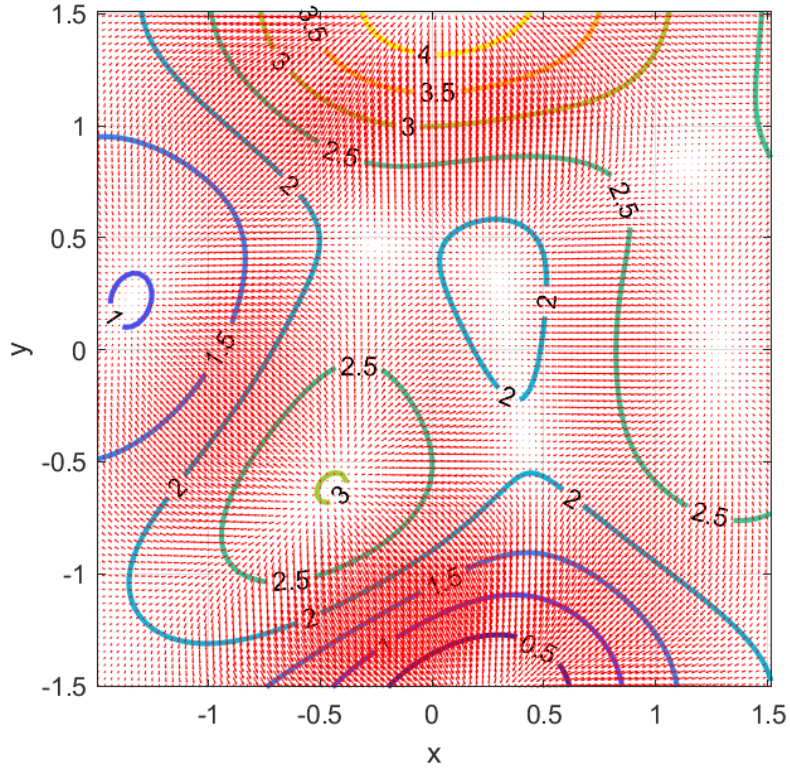




Gradient descent

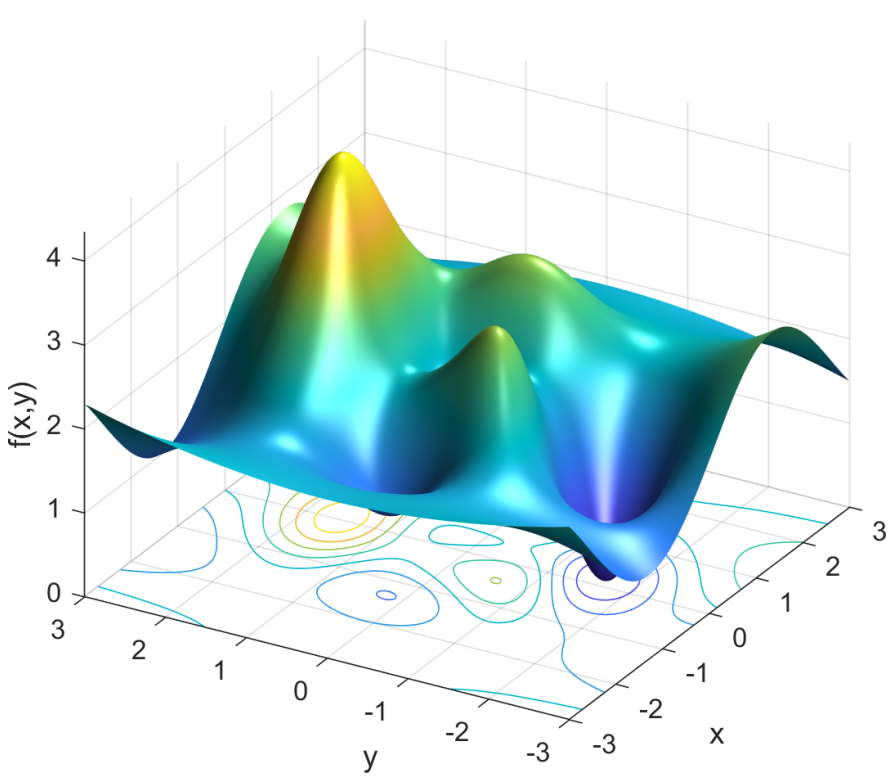
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

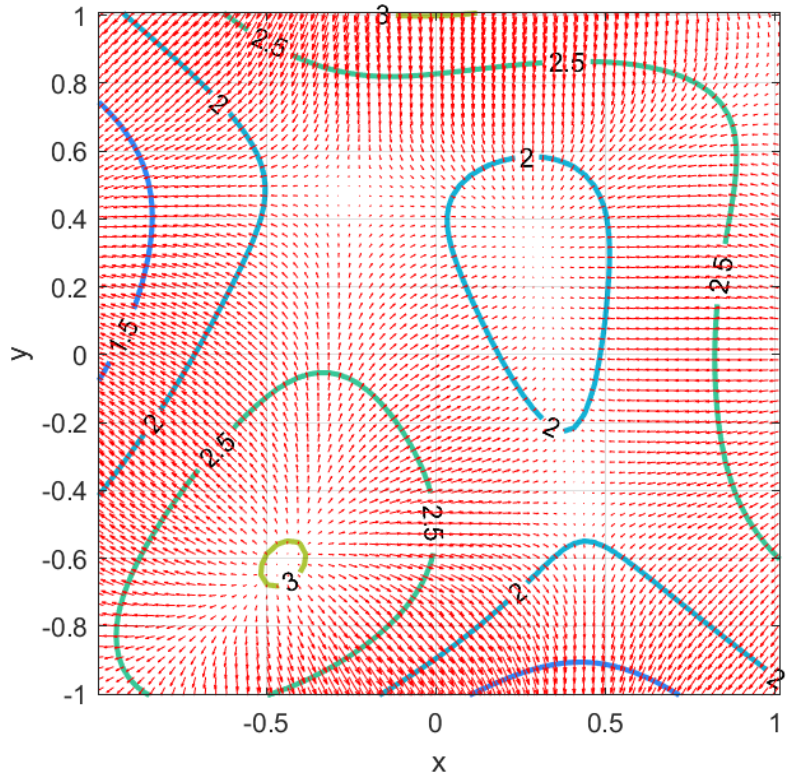




Gradient descent

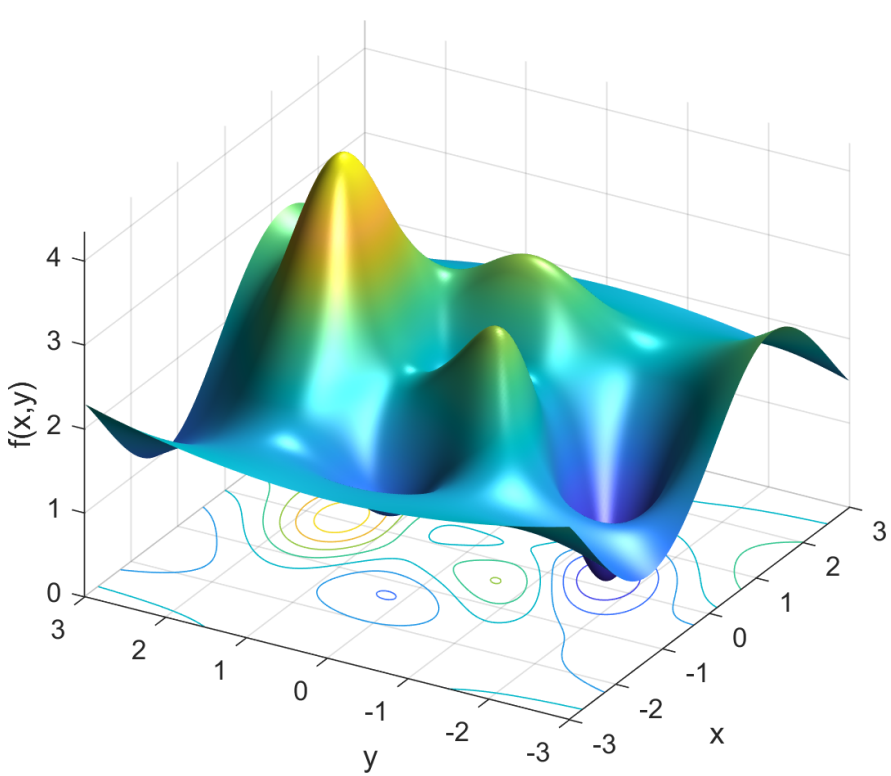
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

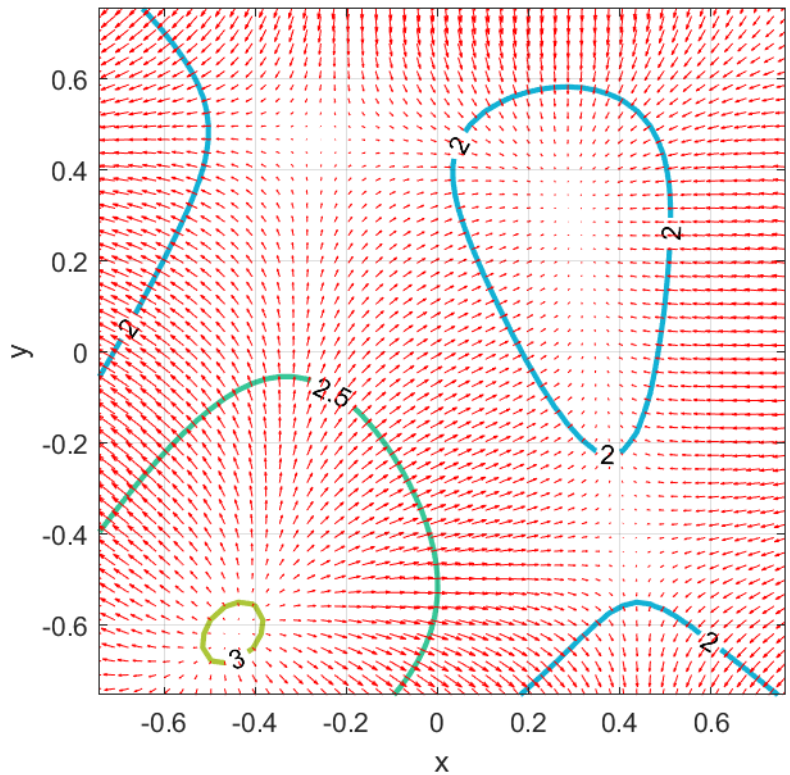




Gradient descent

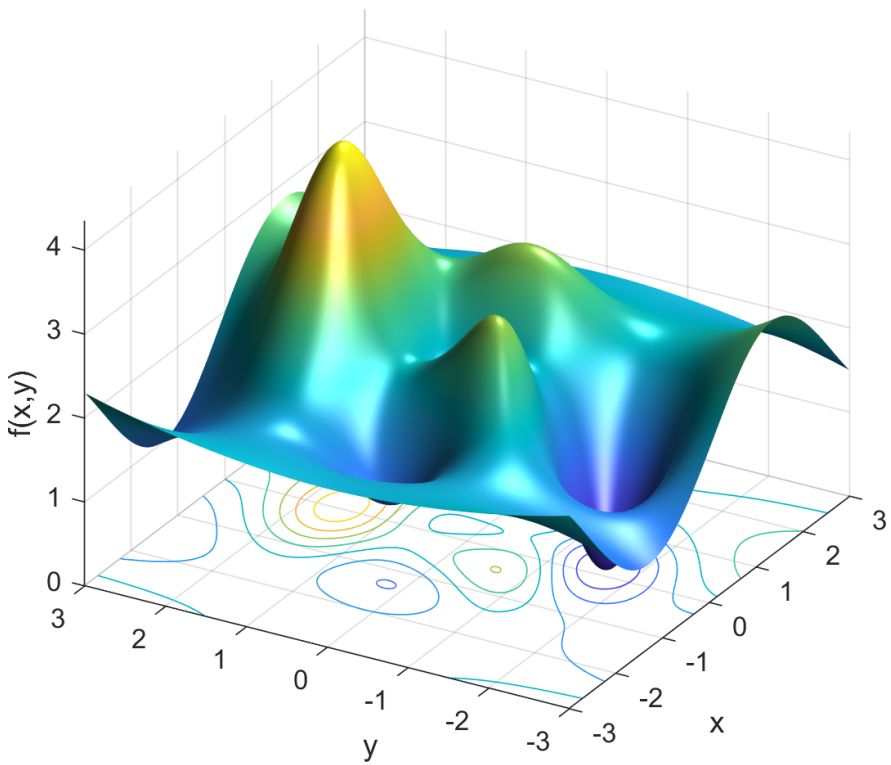
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

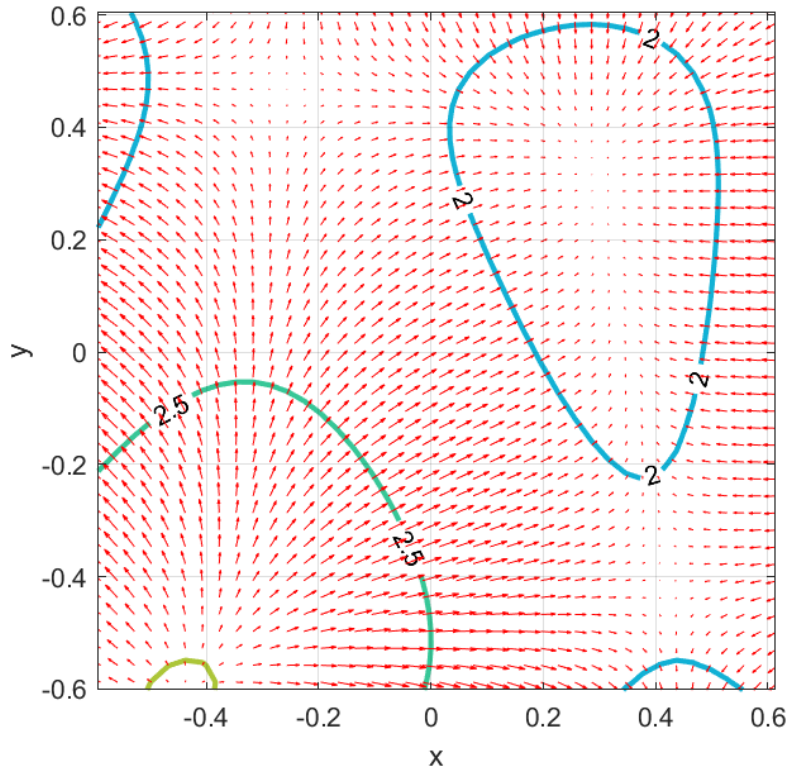




Gradient descent

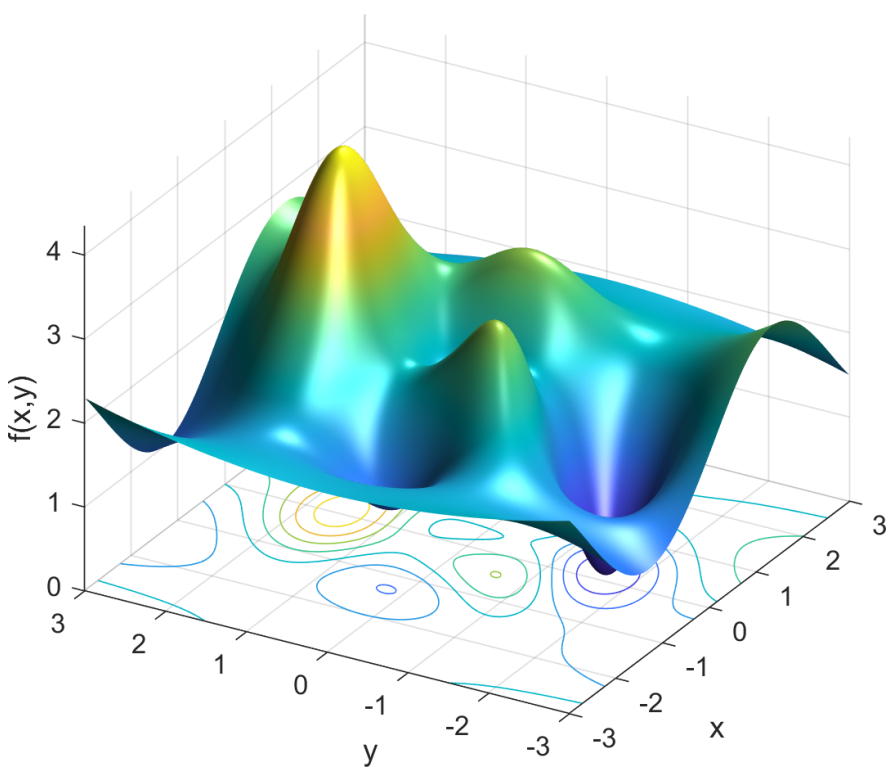
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

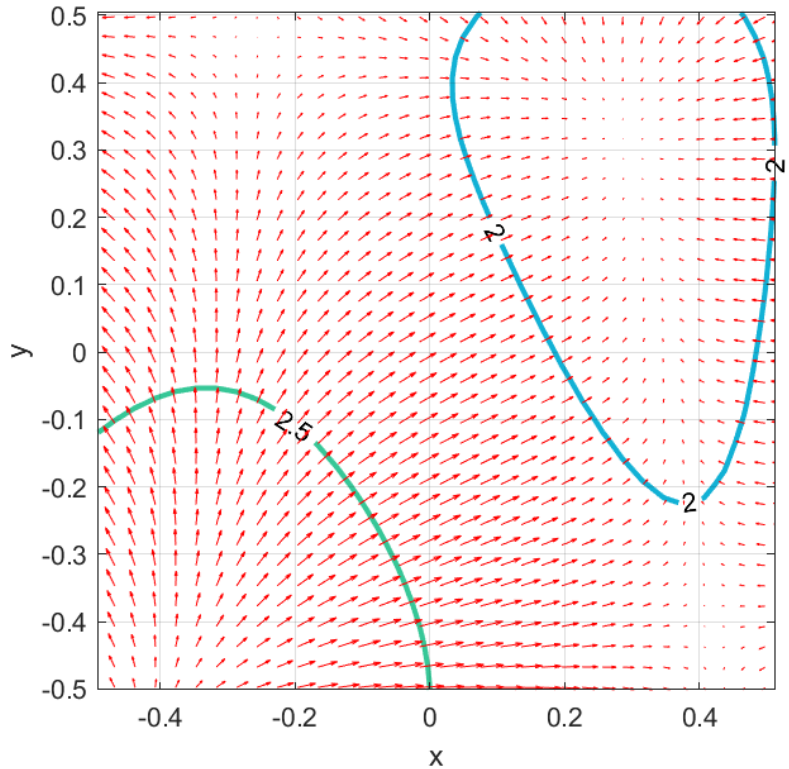




Gradient descent

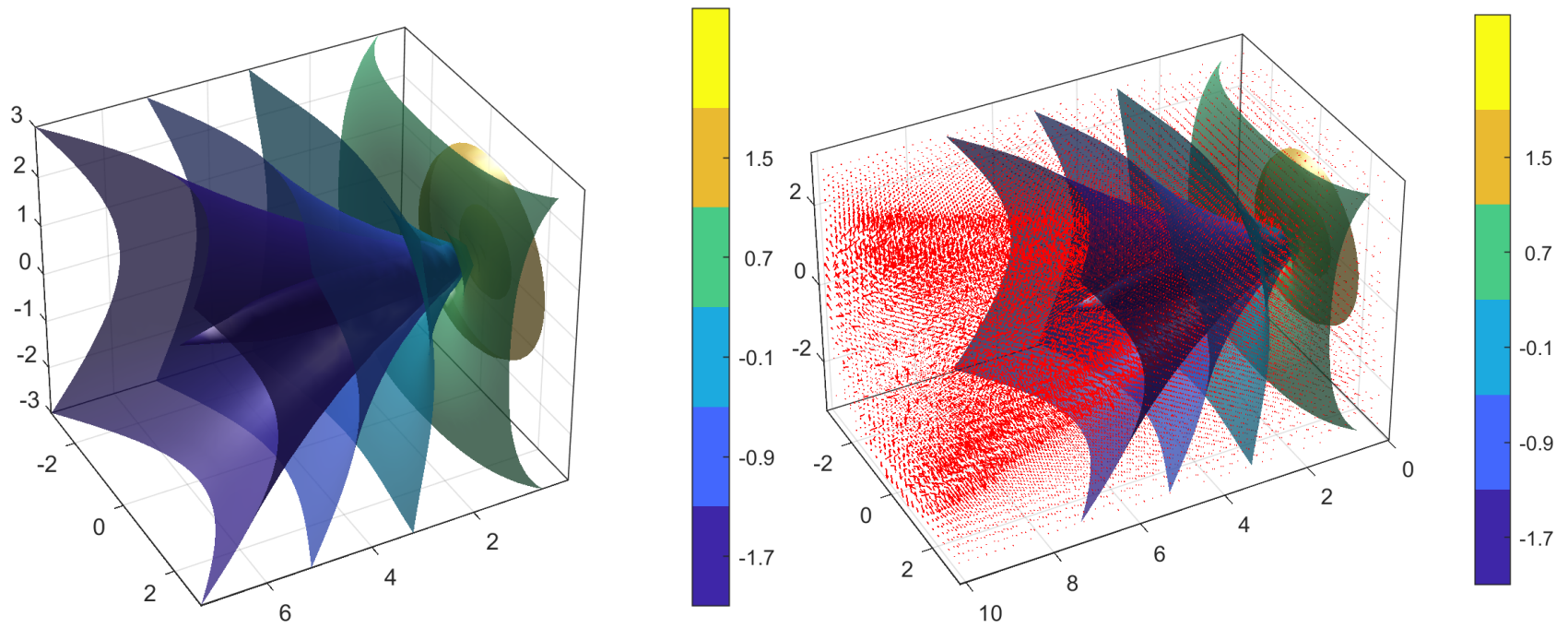
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$





Gradient descent: High dimensions

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

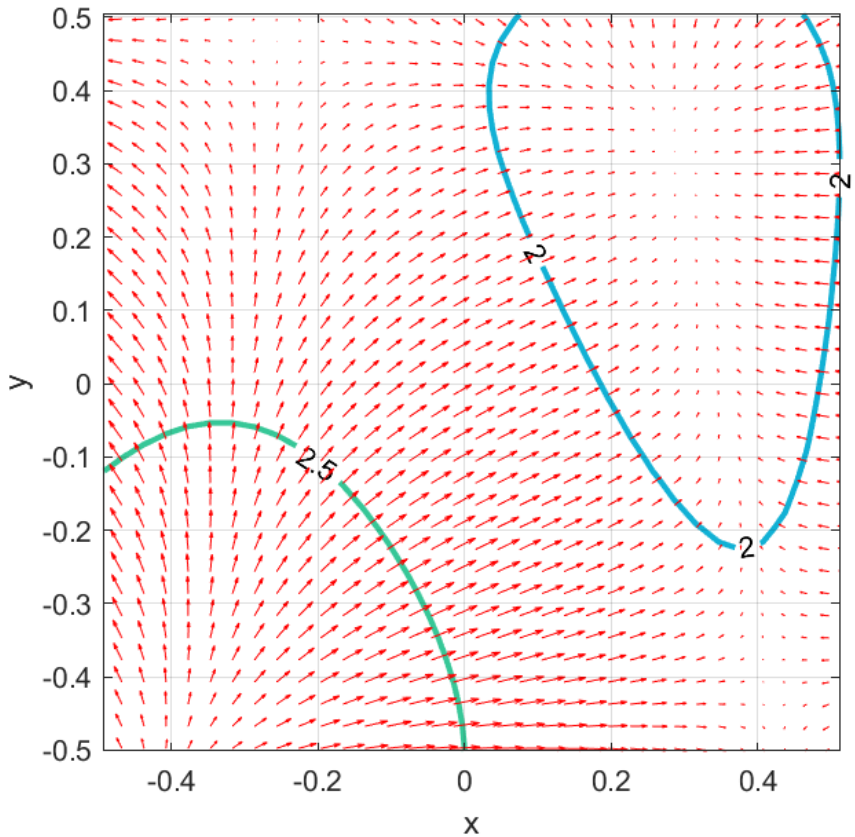


All we say is also valid in $\gg 2$ dimensions.

Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

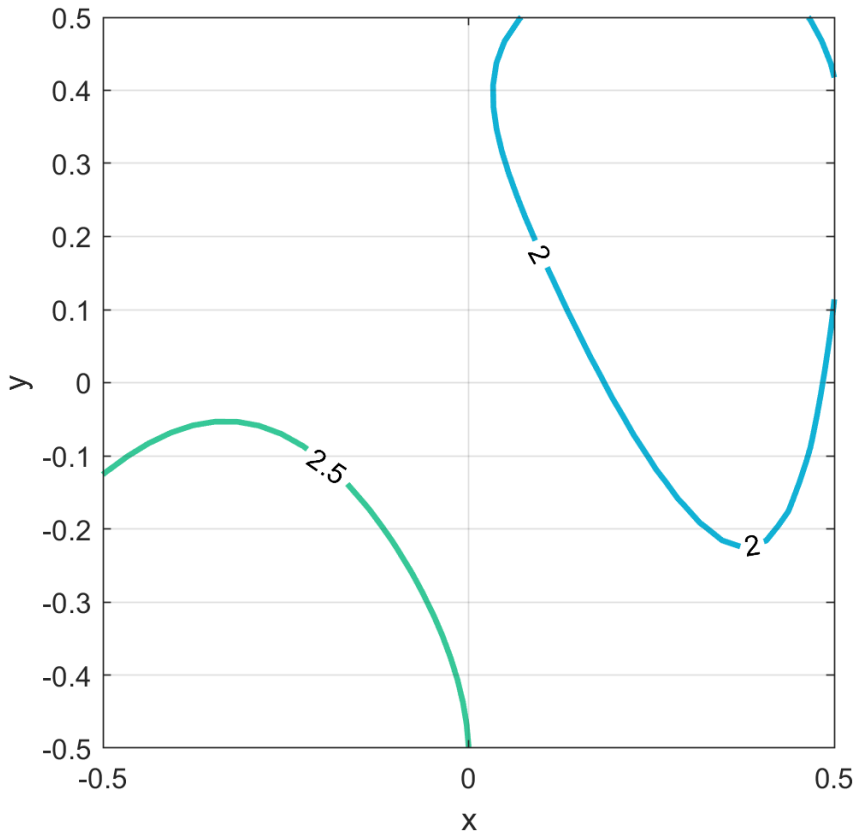
The gradient is orthogonal to level curves / level surfaces.



Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

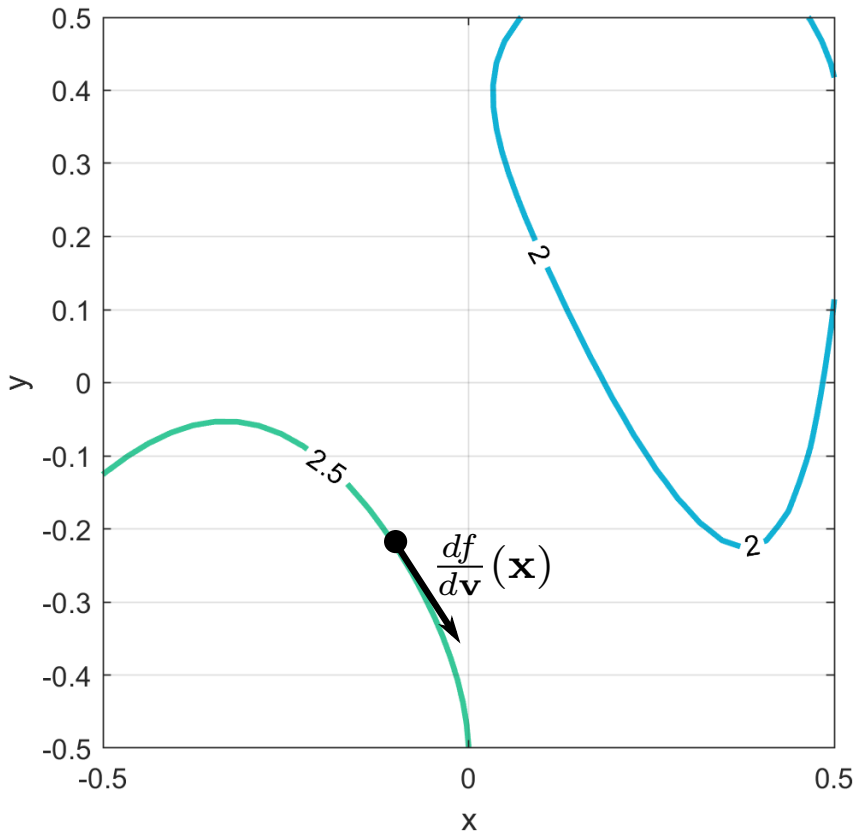
The gradient is orthogonal to level curves / level surfaces.



Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

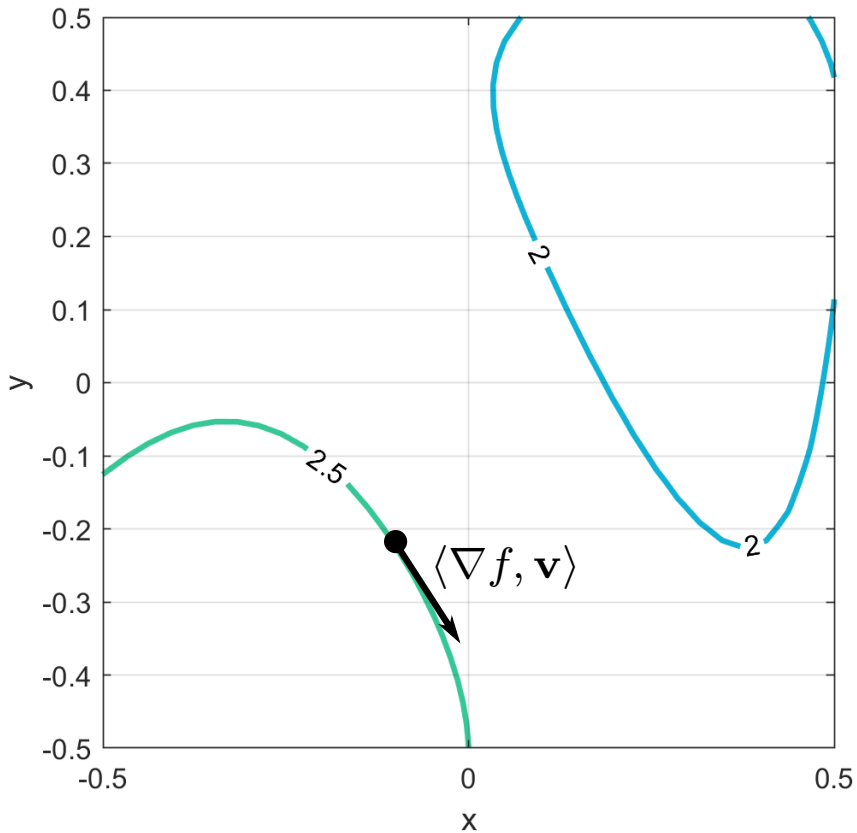
The gradient is orthogonal to level curves / level surfaces.



Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

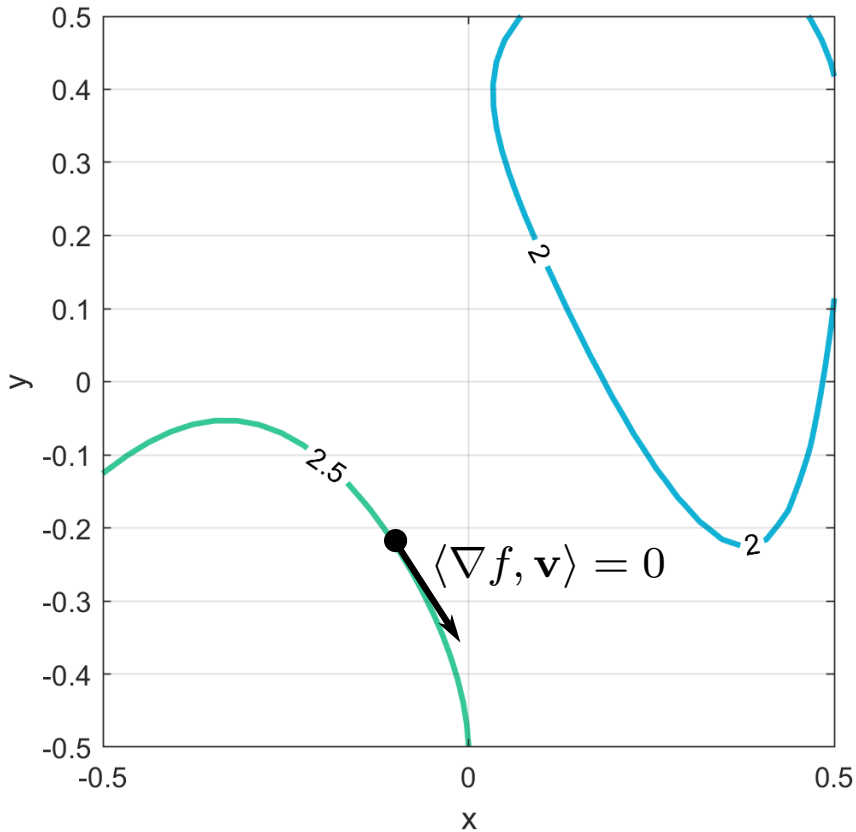
The gradient is orthogonal to level curves / level surfaces.



Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The gradient is orthogonal to level curves / level surfaces.

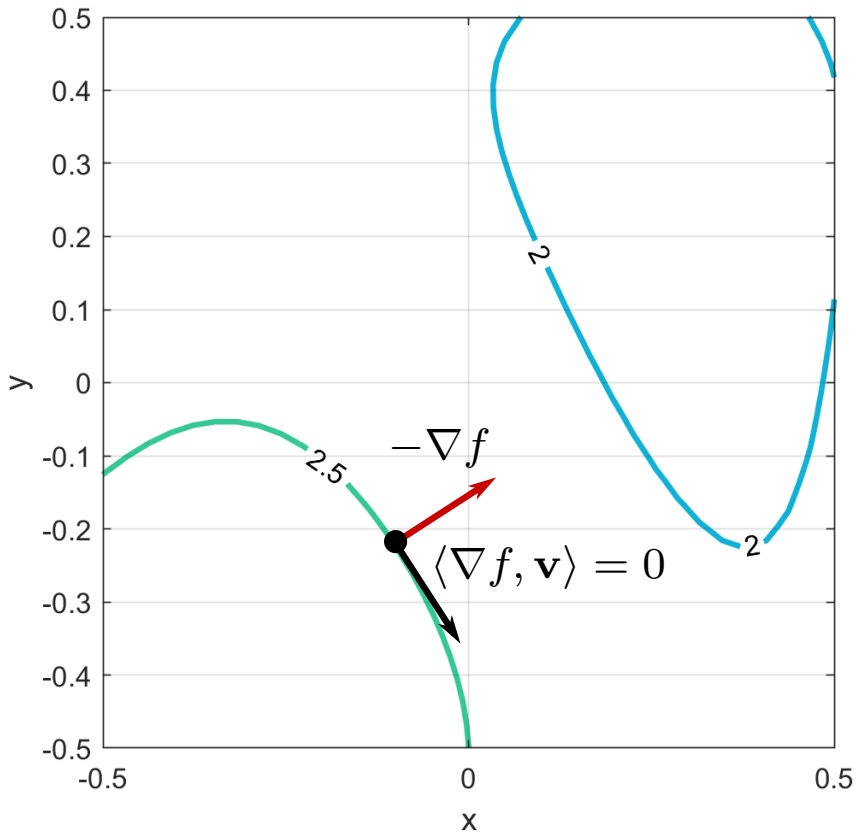


The directional derivative is zero along isocurves.

Gradient descent: Orthogonality

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The gradient is orthogonal to level curves / level surfaces.



The directional derivative is zero along isocurves.

Gradient descent: Differentiability

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

Gradient descent requires f to be differentiable at all points.

Warning:

Gradient descent: Differentiability

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

Gradient descent requires f to be differentiable at all points.

Warning:

f has partial (or even directional) derivatives $\not\Rightarrow$ f is differentiable

Gradient descent: Differentiability

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

Gradient descent requires f to be differentiable at all points.

Warning:

f has partial (or even directional) derivatives $\not\Rightarrow$ f is differentiable

f has continuous gradient $\Rightarrow$ f is differentiable

Gradient descent: Differentiability

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

Gradient descent requires f to be differentiable at all points.

Warning:

f has partial (or even directional) derivatives $\not\Rightarrow$ f is differentiable

f has continuous gradient $\Rightarrow$ f is differentiable

See examples at: `https://mathinsight.org/differentiability_multivariable_subtleties`

Gradient descent: Stationary points

A stationary point is such that:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \cancelto{0}{\nabla f(\mathbf{x}^{(t)})}$$

Gradient descent "gets stuck" at stationary points.

Gradient descent: Stationary points

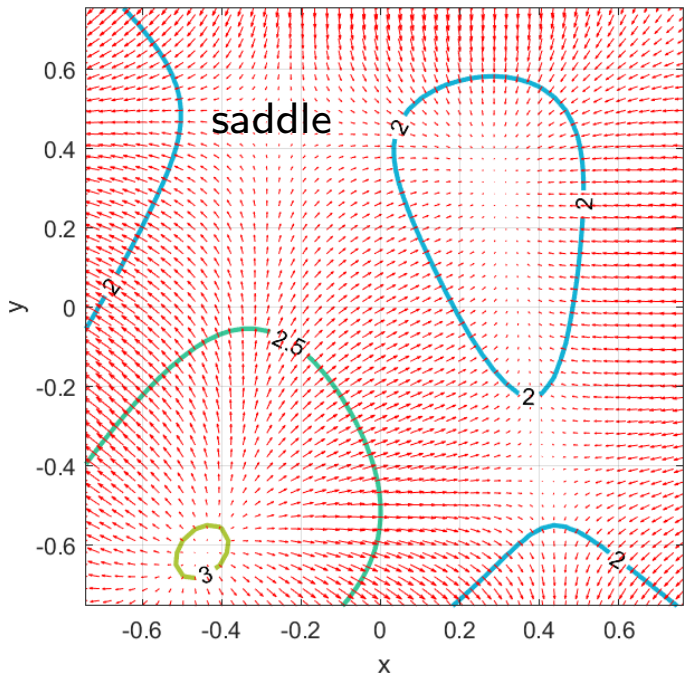
A stationary point is such that:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \cancelto{0}{\nabla f(\mathbf{x}^{(t)})}$$

Gradient descent "gets stuck" at stationary points.

However:

- Stationary point $\not\Rightarrow$ local minimum



Gradient descent: Stationary points

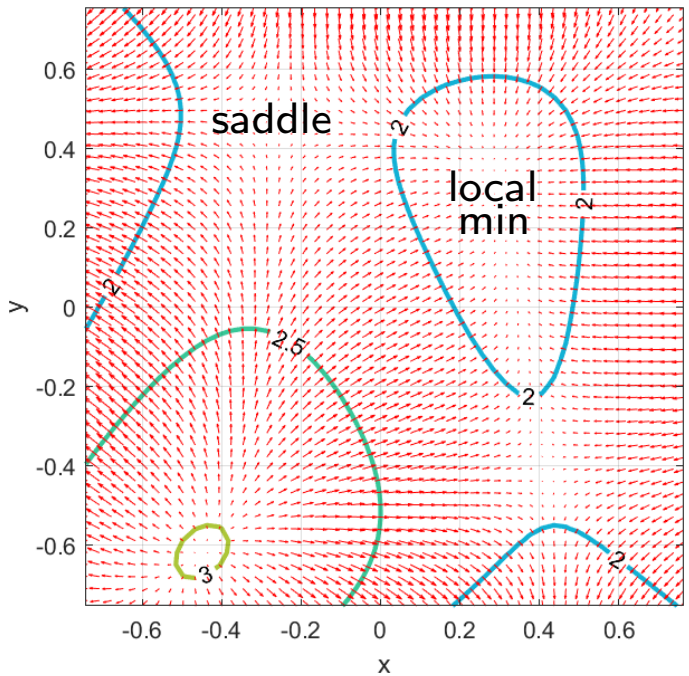
A stationary point is such that:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \cancelto{0}{\nabla f(\mathbf{x}^{(t)})}$$

Gradient descent "gets stuck" at stationary points.

However:

- Stationary point $\not\Rightarrow$ local minimum $\not\Rightarrow$ global minimum.



Gradient descent: Stationary points

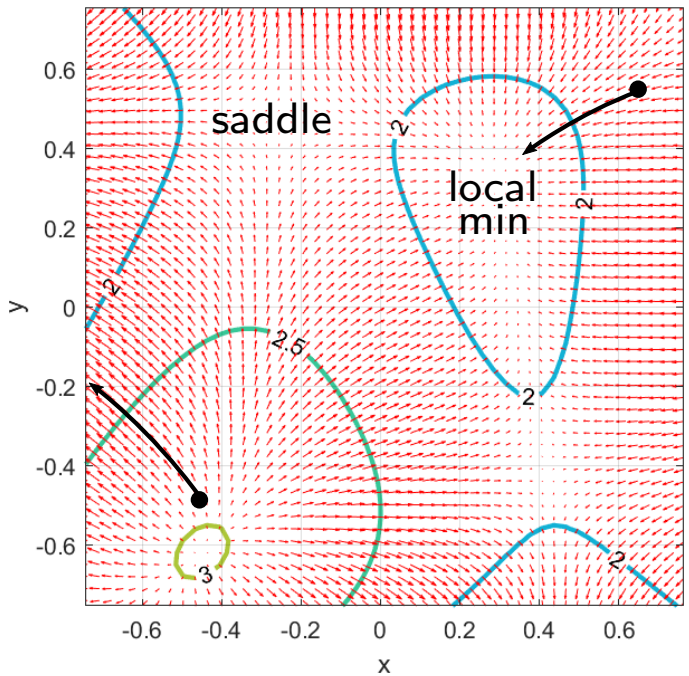
A stationary point is such that:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \cancelto{0}{\nabla f(\mathbf{x}^{(t)})}$$

Gradient descent "gets stuck" at stationary points.

However:

- Stationary point $\not\Rightarrow$ local minimum $\not\Rightarrow$ global minimum.
- Which stationary point depends on the initialization.



Gradient descent: Step length

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The parameter $\alpha > 0$ affects the step length.

Gradient descent: Step length

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The parameter $\alpha > 0$ affects the step length.

Remark: The length of a step is not simply α, but $\alpha \|\nabla f\|$.

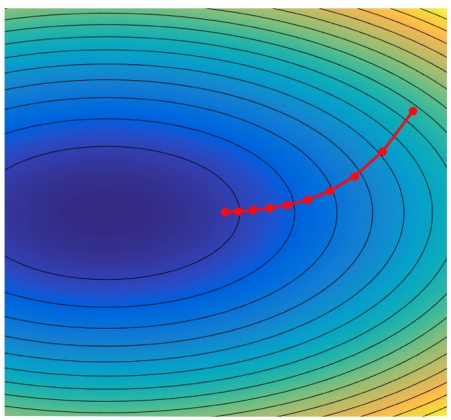
Gradient descent: Step length

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The parameter $\alpha > 0$ affects the step length.

Remark: The length of a step is not simply α, but $\alpha \|\nabla f\|$.

- Too small: slow convergence speed



small α

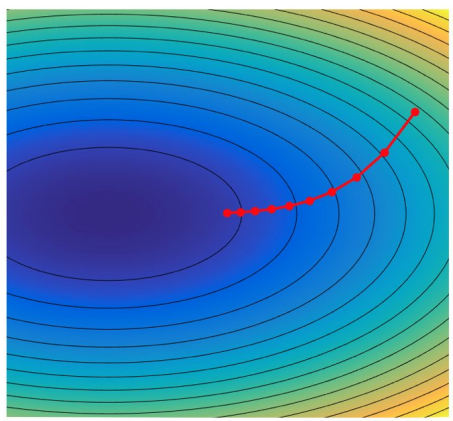
Gradient descent: Step length

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

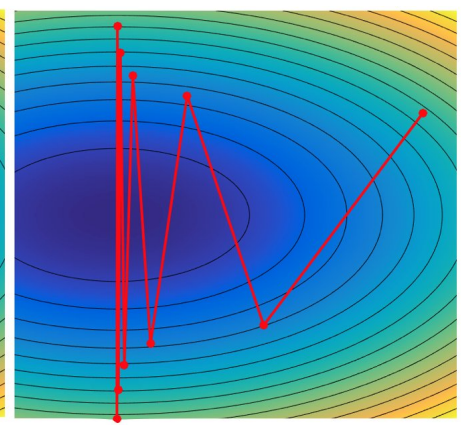
The parameter $\alpha > 0$ affects the step length.

Remark: The length of a step is not simply α, but $\alpha \|\nabla f\|$.

- Too small: slow convergence speed
- Too big: risk of overshooting



small α



large α

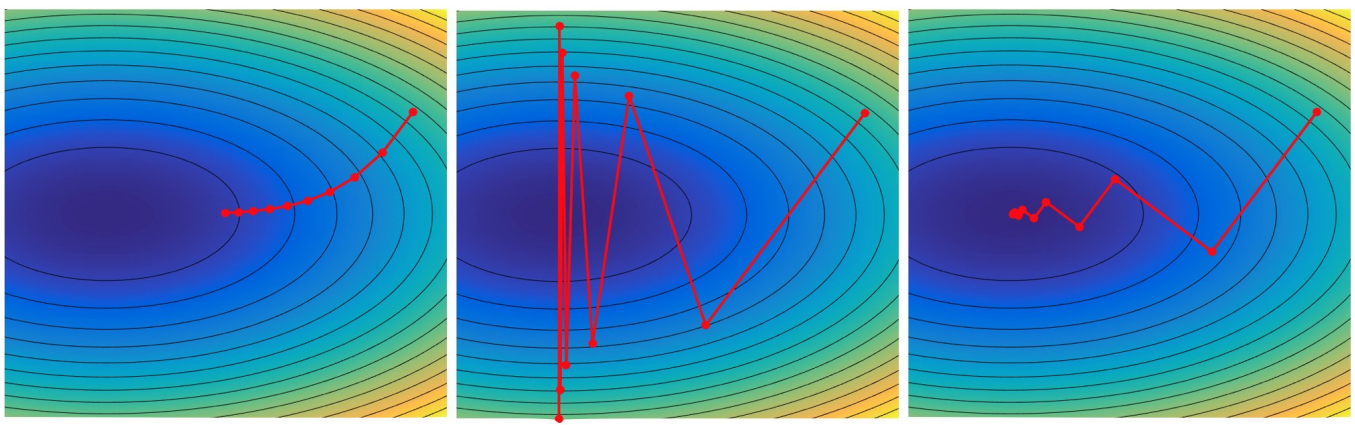
Gradient descent: Step length

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

The parameter $\alpha > 0$ affects the step length.

Remark: The length of a step is not simply α, but $\alpha \|\nabla f\|$.

- Too small: slow convergence speed
- Too big: risk of overshooting
- Optimal values can be found via line search algorithms



small α | large α | optimal α

$\arg\min_\alpha f(\mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)}))$

Decay and momentum

The parameter α can be adaptive or follow a schedule.

Decay and momentum

The parameter α can be adaptive or follow a schedule.

- Decrease α according to a decay parameter ρ:
  Examples:

$$\alpha^{(t+1)} = (1-\frac{t}{\rho})\alpha^{(0)} + \frac{t}{\rho}\alpha^{(\rho)}\,, \quad \alpha^{(t+1)} = \frac{\alpha^{(t)}}{1+\rho t}\,, \quad \alpha^{(t+1)} = \alpha^{(0)} e^{-\rho t}$$

Decay and momentum

The parameter α can be adaptive or follow a schedule.

- Decrease α according to a decay parameter ρ:
  Examples:

$$\alpha^{(t+1)} = (1-\frac{t}{\rho})\alpha^{(0)} + \frac{t}{\rho}\alpha^{(\rho)}\,, \quad \alpha^{(t+1)} = \frac{\alpha^{(t)}}{1+\rho t}\,, \quad \alpha^{(t+1)} = \alpha^{(0)} e^{-\rho t}$$

- Accumulate past gradients and keep moving in their direction:

$$\mathbf{v}^{(t+1)} = \lambda \mathbf{v}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)}) \quad \text{momentum}$$
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{v}^{(t+1)}$$

Decay and momentum

The parameter α can be adaptive or follow a schedule.

- Decrease α according to a decay parameter ρ:
  Examples:

$$\alpha^{(t+1)} = (1-\frac{t}{\rho})\alpha^{(0)} + \frac{t}{\rho}\alpha^{(\rho)}\,, \quad \alpha^{(t+1)} = \frac{\alpha^{(t)}}{1+\rho t}\,, \quad \alpha^{(t+1)} = \alpha^{(0)} e^{-\rho t}$$

- Accumulate past gradients and keep moving in their direction:

$$\mathbf{v}^{(t+1)} = \lambda \mathbf{v}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)}) \quad \text{momentum}$$
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{v}^{(t+1)}$$

  Step length $\propto$ how aligned is the sequence of gradients.

$$\frac{1}{1-\lambda}\alpha\|\nabla f\|$$

Decay and momentum

The parameter α can be adaptive or follow a schedule.

- Decrease α according to a decay parameter ρ:
  Examples:

$$\alpha^{(t+1)} = (1-\frac{t}{\rho})\alpha^{(0)} + \frac{t}{\rho}\alpha^{(\rho)}\,, \quad \alpha^{(t+1)} = \frac{\alpha^{(t)}}{1+\rho t}\,, \quad \alpha^{(t+1)} = \alpha^{(0)} e^{-\rho t}$$

- Accumulate past gradients and keep moving in their direction:

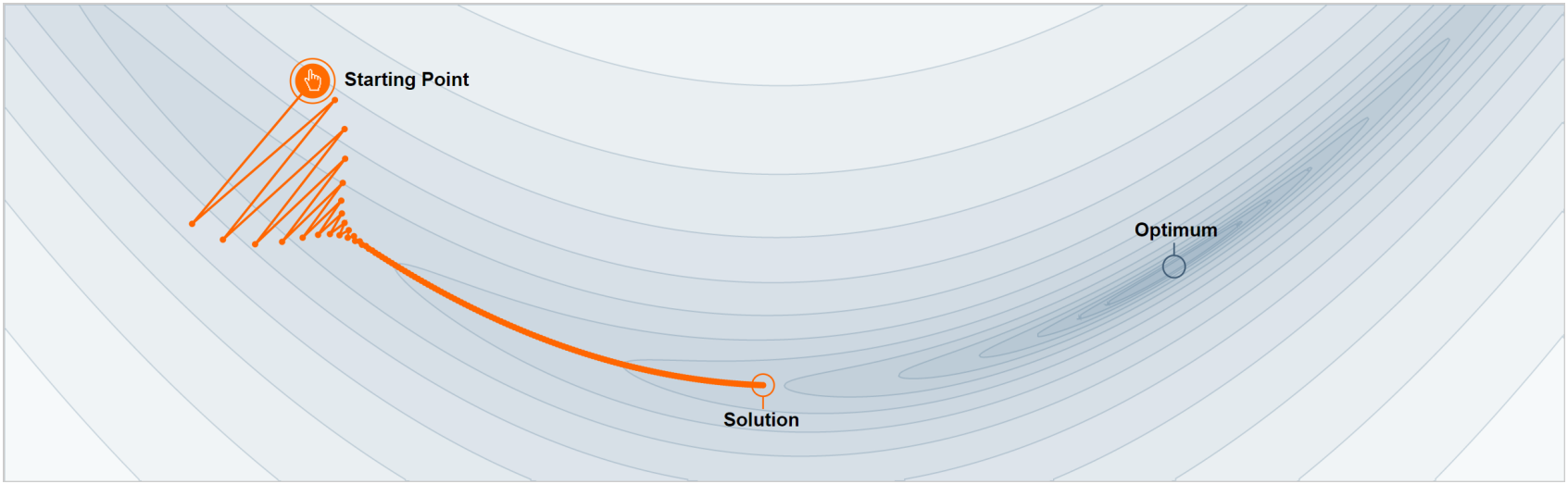
$$\mathbf{v}^{(t+1)} = \lambda \mathbf{v}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)}) \quad \text{momentum}$$
$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} + \mathbf{v}^{(t+1)}$$

  Step length $\propto$ how aligned is the sequence of gradients.

$$\frac{1}{1-\lambda}\alpha\|\nabla f\|$$

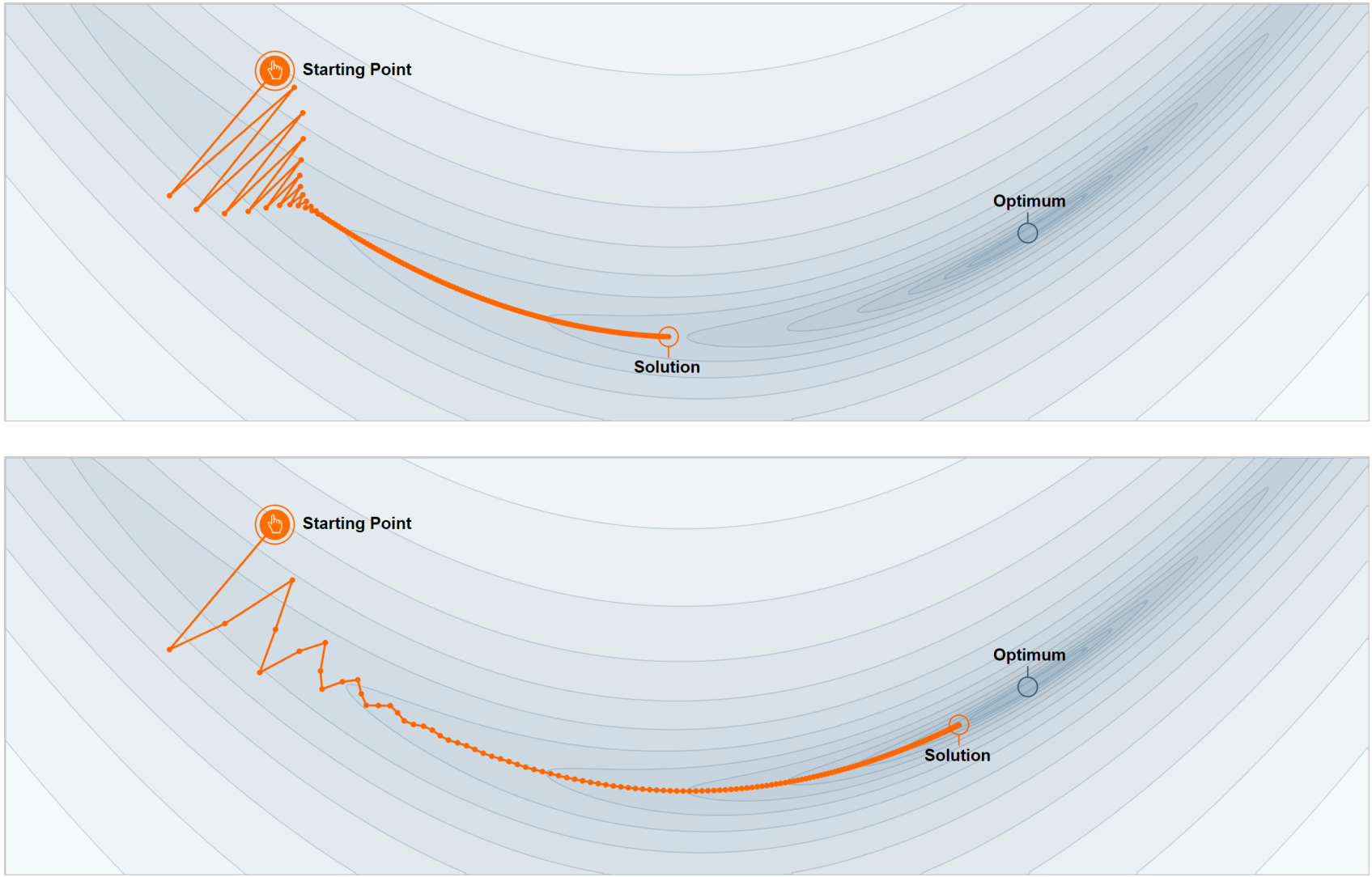
  Acceleration effect for big λ + escape from local minima.

Momentum

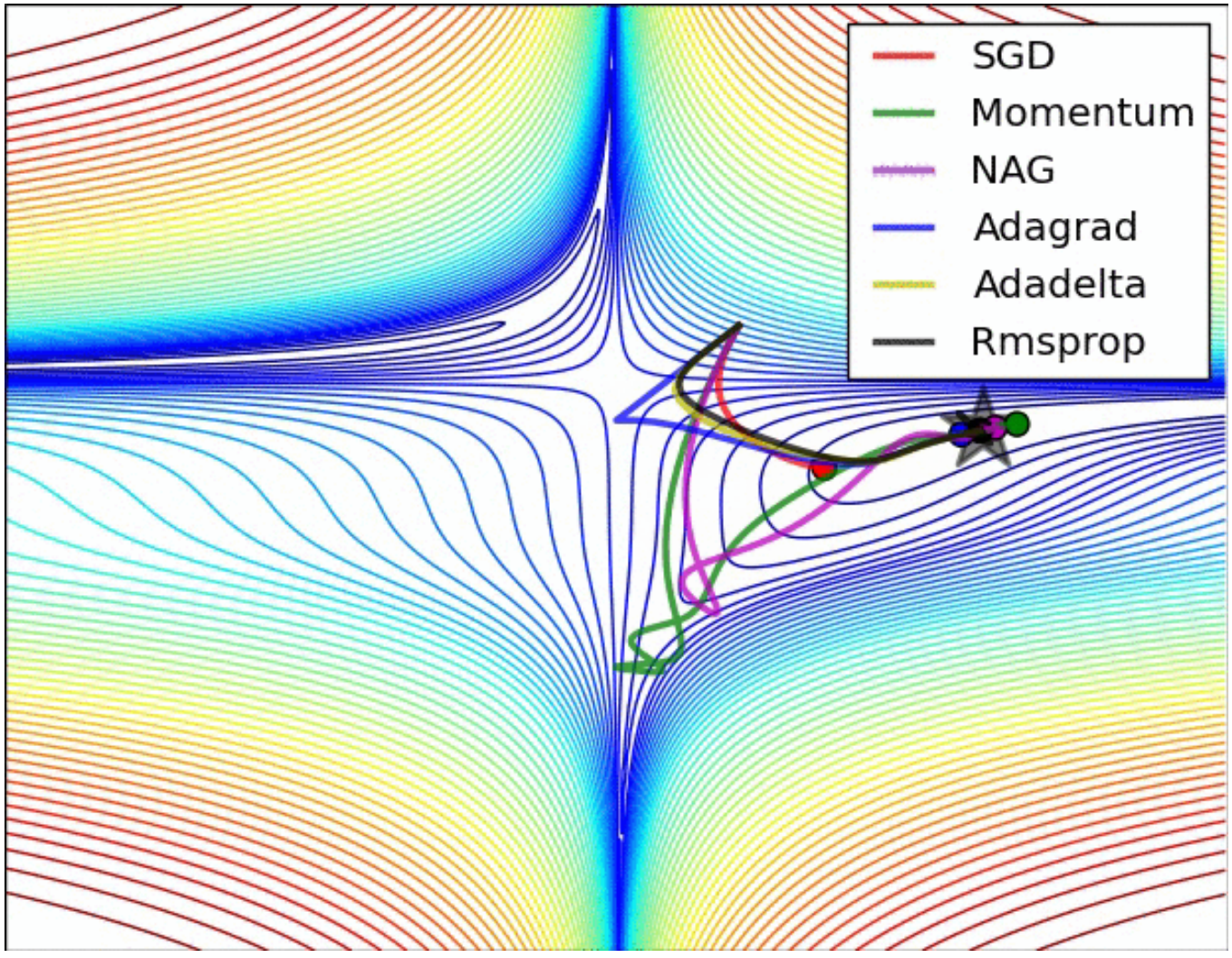


Goh, "Why momentum really works", Distill 2017

Momentum



Goh, "Why momentum really works", Distill 2017


SGD
Momentum
NAG
Adagrad
Adadelta
Rmsprop

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(t)} - \alpha \nabla f(\mathbf{x}^{(t)})$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(1)} = \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)})$$
$$\mathbf{x}^{(2)} = \mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)})$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\begin{aligned}\mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - \alpha\nabla f(\mathbf{x}^{(0)}) \\ \mathbf{x}^{(2)} &= \mathbf{x}^{(1)} - \alpha\nabla f(\mathbf{x}^{(1)}) \\ &= \mathbf{x}^{(0)} - \alpha\nabla f(\mathbf{x}^{(0)}) - \alpha\nabla f(\mathbf{x}^{(1)})\end{aligned}$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$
\begin{aligned}
\mathbf{x}^{(1)} &= \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)}) \\
\mathbf{x}^{(2)} &= \mathbf{x}^{(1)} - \alpha \nabla f(\mathbf{x}^{(1)}) \\
&= \mathbf{x}^{(0)} - \alpha \nabla f(\mathbf{x}^{(0)}) - \alpha \nabla f(\mathbf{x}^{(1)}) \\
&\vdots \\
\mathbf{x}^{(t+1)} &= \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})
\end{aligned}
$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})$$

With momentum:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \frac{1 - \lambda^{t+1-i}}{1 - \lambda} \nabla f(\mathbf{x}^{(i)})$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})$$

With momentum:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \frac{1 - \lambda^{t+1-i}}{1 - \lambda} \nabla f(\mathbf{x}^{(i)})$$

The more general form:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} + \alpha \sum_{i=1}^{t} \gamma_i^t \nabla f(\mathbf{x}^{(i)}) \quad \text{for some } \gamma_i$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})$$

With momentum:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \frac{1-\lambda^{t+1-i}}{1-\lambda} \nabla f(\mathbf{x}^{(i)})$$

The more general form:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} + \alpha \sum_{i=1}^{t} \Gamma_i^t \nabla f(\mathbf{x}^{(i)}) \quad \text{for some diag. matrix } \Gamma_i$$

First-order acceleration methods

Let us try to unroll gradient descent:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \nabla f(\mathbf{x}^{(i)})$$

With momentum:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} - \alpha \sum_{i=1}^{t} \frac{1-\lambda^{t+1-i}}{1-\lambda} \nabla f(\mathbf{x}^{(i)})$$

The more general form:

$$\mathbf{x}^{(t+1)} = \mathbf{x}^{(0)} + \alpha \sum_{i=1}^{t} \Gamma_i^t \nabla f(\mathbf{x}^{(i)}) \quad \text{for some diag. matrix } \Gamma_i$$

generalizes optimization algorithms like ADAM, AdaGrad, etc.

Gradient descent as a general tool

Gradient descent can be applied to nonconvex problems, without optimality guarantees.

Gradient descent as a general tool

Gradient descent can be applied to nonconvex problems, without optimality guarantees.

Even for convex problems like:

- Linear regression
- Logistic regression

We get more efficient and numerically stable solutions.

Gradient descent as a general tool

Gradient descent can be applied to nonconvex problems, without optimality guarantees.

Even for convex problems like:

- Linear regression ($\mathbf{X}$ can be huge and must be inverted/factorized)
- Logistic regression

We get more efficient and numerically stable solutions.

Gradient descent as a general tool

Gradient descent can be applied to nonconvex problems, without optimality guarantees.

Even for convex problems like:

- Linear regression ($\mathbf{X}$ can be huge and must be inverted/factorized)
- Logistic regression (no closed form solution)

We get more efficient and numerically stable solutions.