ORF522 – Linear and Nonlinear Optimization

14. Gradient descent

Bartolomeo Stellato — Fall 2022

Ed Forum

- Can you explain more why KKT for convex problems is sufficient condition?
- What's the advantage of using normal cone to characterize optimality?

Recap

Strong duality theorem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \ldots, m \\ & h_i(x) = 0, \quad i = 1, \ldots, p \end{array}$$

Theorem

If the problem is convex and there exists at least a strictly feasible x, *i.e.*,

$g_i(x) \leq 0,$ (for all affine g_i)

$g_i(x) < 0,$ (for all non-affine g_i)

$h_i(x) = 0, \quad i = 1, \ldots, p$

Slater's condition

then $p^\star = d^\star$ (**strong duality holds**)

Strong duality theorem

$$\begin{array}{ll} \text{minimize} & f(x) \\ \text{subject to} & g_i(x) \leq 0, \quad i = 1, \dots, m \\ & h_i(x) = 0, \quad i = 1, \dots, p \end{array}$$

Theorem

If the problem is convex and there exists at least a strictly feasible x, *i.e.*,

$$\begin{array}{ll} g_i(x) \leq 0, & \text{(for all affine } g_i) \\ g_i(x) < 0, & \text{(for all non-affine } g_i) \\ h_i(x) = 0, & i = 1, \dots, p \end{array}$$

Slater's condition

then $p^\star = d^\star$ (**strong duality holds**)

Remarks

- For nonconvex optimization, we need harder conditions
- Generalizes LP conditions [Lecture 7]

KKT for convex problems

Always sufficient

For $x^\star, y^\star, v^\star$ that satisfy the KKT conditions

$$f(x^\star) = f(x^\star) + \sum_{i=1}^{m} y_i^\star g_i(x^\star) + \sum_{i=1}^{p} v_i^\star h_i(x^\star) = L(x^\star, y^\star v^\star) \qquad \text{(compl slackness)}$$

$$\nabla f(x^\star) + \sum_{i=1}^{m} y_i^\star \nabla g_i(x^\star) + \sum_{i=1}^{p} v_i^\star \nabla h_i(x^\star) = 0 \quad \Rightarrow \quad g(y^\star, v^\star) = L(x^\star, y^\star, v^\star) \qquad \text{(convexity)}$$

Therefore, $f(x^\star) = g(y^\star, v^\star)$ and $x^\star, y^\star, v^\star$ are primal-dual optimal

KKT for convex problems

Always sufficient

For $x^\star, y^\star, v^\star$ that satisfy the KKT conditions

$$f(x^\star) = f(x^\star) + \sum_{i=1}^{m} y_i^\star g_i(x^\star) + \sum_{i=1}^{p} v_i^\star h_i(x^\star) = L(x^\star, y^\star v^\star) \qquad \text{(compl slackness)}$$

$$\nabla f(x^\star) + \sum_{i=1}^{m} y_i^\star \nabla g_i(x^\star) + \sum_{i=1}^{p} v_i^\star \nabla h_i(x^\star) = 0 \quad \Rightarrow \quad g(y^\star, v^\star) = L(x^\star, y^\star, v^\star) \qquad \text{(convexity)}$$

Therefore, $f(x^\star) = g(y^\star, v^\star)$ and $x^\star, y^\star, v^\star$ are primal-dual optimal

Necessary when constraint qualifications (Slater's) condition holds

If $x^\star$ strictly primal feasible (Slater's), then strong duality $f(x^\star) = g(y^\star, v^\star)$

Therefore, dual optimum attained and KKT conditions satisfied

Today's lecture

[Chapter 1 and 2, ILCO][Chapter 9, CO][Chapter 5, FMO]

Gradient descent algorithms

- Optimization algorithms and convergence rates
- Gradient descent
- Fixed step size:
  - quadratic functions, smooth and strongly convex, only smooth
- Line search: can we adapt the step size?
- Issues with gradient descent

Optimization algorithms and convergence rates

Iterative solution idea

1. Start from initial point x^0
2. Generate sequence $\{x^k\}$ by applying an operator

$$x^{k+1} = T(x^k)$$

3. Converge to fixed-point $x^\star = T(x^\star)$ for which **necessary optimality conditions** hold

Note: typically, we have $f(x^{k+1}) \leq f(x^k)$

Convergence rates

Rank methods by how fast they converge

Error function $e(x) \geq 0$ such that $e(x^\star) = 0$

- Cost function distance: $e(x) = f(x) - f(x^\star)$

- Solution distance: $e(x) = \|x - x^\star\|_2$

Convergence rates

Rank methods by how fast they converge

Error function $e(x) \geq 0$ such that $e(x^\star) = 0$

- Cost function distance: $e(x) = f(x) - f(x^\star)$
- Solution distance: $e(x) = \|x - x^\star\|_2$

Convergence rate

A sequence converges with order p and factor c if

$$\lim_{k \to \infty} \frac{e(x^{k+1})}{e(x^k)^p} = c$$

Convergence rates types

Linear convergence (geometric) ($c \in (0, 1)$)

$$e(x^{k+1}) \leq ce(x^k)$$

Examples

$$e(x^k) = 0.6^k$$

Convergence rates types

Linear convergence (geometric) ($c \in (0,1)$)

$$e(x^{k+1}) \leq c e(x^k)$$

Sublinear convergence (slower than linear)

$$e(x^{k+1}) \leq \frac{M}{(k+1)^q}, \quad \text{with} \quad q = 0.5, 1, 2, ...$$

Examples

$$e(x^k) = 0.6^k$$

$$e(x^k) = \frac{1}{\sqrt{k}}$$

Convergence rates types

Linear convergence (geometric) ($c \in (0,1)$)

$$e(x^{k+1}) \leq c e(x^k)$$

Sublinear convergence (slower than linear)

$$e(x^{k+1}) \leq \frac{M}{(k+1)^q}, \quad \text{with} \quad q = 0.5, 1, 2, ...$$

Superlinear convergence (faster than linear)

If it converges linearly $p = 1$ for any factor $c \in (0,1)$

Examples

$$e(x^k) = 0.6^k$$

$$e(x^k) = \frac{1}{\sqrt{k}}$$

$$e(x^k) = \frac{1}{k^k}$$

Convergence rates types

Linear convergence (geometric) ($c \in (0,1)$)

$$e(x^{k+1}) \leq c e(x^k)$$

Sublinear convergence (slower than linear)

$$e(x^{k+1}) \leq \frac{M}{(k+1)^q}, \quad \text{with} \quad q = 0.5, 1, 2, ...$$

Superlinear convergence (faster than linear)

If it converges linearly $p = 1$ for any factor $c \in (0,1)$

Quadratic convergence (c can be > 1)

$$e(x^{k+1}) \leq c e(x^k)^2$$

Examples

$$e(x^k) = 0.6^k$$

$$e(x^k) = \frac{1}{\sqrt{k}}$$

$$e(x^k) = \frac{1}{k^k}$$

$$e(x^k) = 0.9^{(2^k)}$$

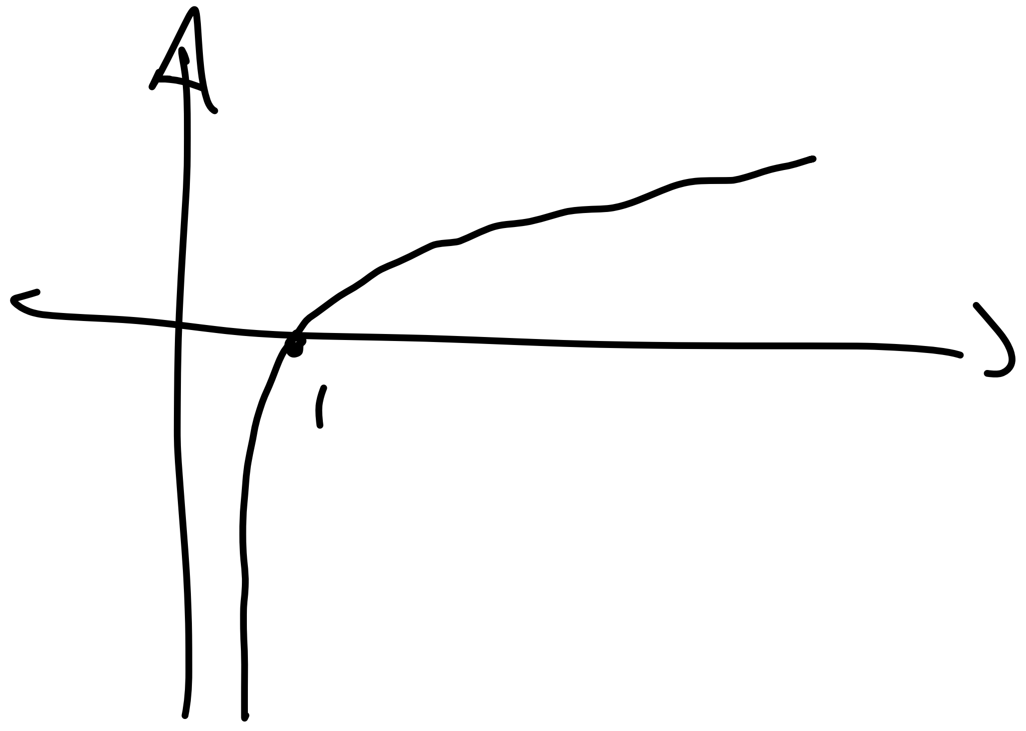
Convergence rates

Number of iterations

Solve inequality for k

Convergence rates

Number of iterations



Solve inequality for k

Example: linear convergence ($c \in (0,1)$)

$$e(x^{k+1}) \leq c e(x^k)$$

$$e(x^{k+1}) \leq \epsilon \quad \Longrightarrow \quad c^k e(x^0) \leq \epsilon \quad \Longrightarrow \quad k \geq O(\log(1/\epsilon))$$

$$K \log c + \log(e(x^0)) \leq \log(\varepsilon)$$

$$K \log c \leq \log(\varepsilon) - \log(e(x^0))$$

$$K \geq \frac{1}{\log(c)} \log(\varepsilon) \ldots$$

$$\geq -\left|\frac{1}{\log(c)}\right| \log(\varepsilon)$$

$$\hookrightarrow -\log(\varepsilon) \Rightarrow \log(\varepsilon^{-1})$$

Convergence rates

Number of iterations

Solve inequality for k

Example: linear convergence ($c \in (0, 1)$)

$$e(x^{k+1}) \leq ce(x^k)$$

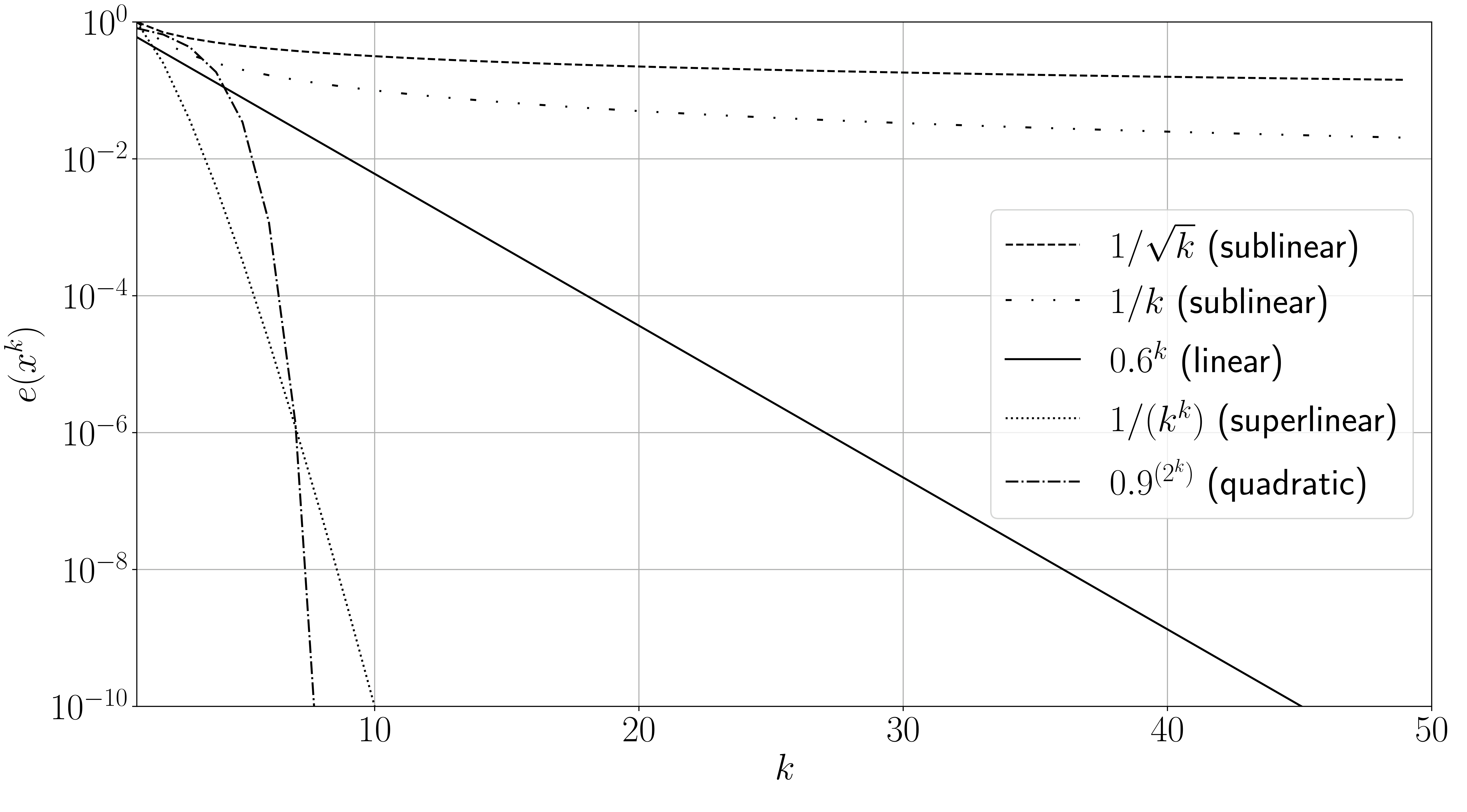
$$e(x^{k+1}) \leq \epsilon \quad \Longrightarrow \quad c^k e(x^0) \leq \epsilon \quad \Longrightarrow \quad k \geq O(\log(1/\epsilon))$$

Example: sublinear convergence

$$e(x^{k+1}) \leq \frac{M}{k+1} \quad \Longrightarrow \quad k \geq O(1/\epsilon)$$

Convergence rates

Examples



Optimization methods overview

Zero order. They rely only on $f(x)$. Not possible to evaluate the curvature. Extremely slow.

Examples: Random search, genetic algorithms, particle swarm optimization, simulated annealing, etc.

Optimization methods overview

Zero order. They rely only on $f(x)$. Not possible to evaluate the curvature. Extremely slow.

Examples: Random search, genetic algorithms, particle swarm optimization, simulated annealing, etc.

First order. They use $f(x)$ and $\nabla f(x)$ or $\partial f(x)$. Inexpensive iterations make them extremely popular in large-scale optimization and machine learning

Examples: Gradient descent, stochastic gradient descent, coordinate descent, proximal algorithms, ADMM.

Optimization methods overview

Zero order. They rely only on $f(x)$. Not possible to evaluate the curvature. Extremely slow.

Examples: Random search, genetic algorithms, particle swarm optimization, simulated annealing, etc.

First order. They use $f(x)$ and $\nabla f(x)$ or $\partial f(x)$. Inexpensive iterations make them extremely popular in large-scale optimization and machine learning

Examples: Gradient descent, stochastic gradient descent, coordinate descent, proximal algorithms, ADMM.

Second order. They use $f(x)$, $\nabla f(x)$ and $\nabla^2 f(x)$. Expensive iterations but very fast convergence

Examples: Newton method, interior-point methods.

Optimization methods overview

Zero order. They rely only on $f(x)$. Not possible to evaluate the curvature. Extremely slow.

Examples: Random search, genetic algorithms, particle swarm optimization, simulated annealing, etc.

First order. They use $f(x)$ and $\nabla f(x)$ or $\partial f(x)$. Inexpensive iterations make them extremely popular in large-scale optimization and machine learning

(our focus)

Examples: Gradient descent, stochastic gradient descent, coordinate descent, proximal algorithms, ADMM.

Second order. They use $f(x)$, $\nabla f(x)$ and $\nabla^2 f(x)$. Expensive iterations but very fast convergence

Examples: Newton method, interior-point methods.

Iterative descent algorithms

Problem setup

Unconstrained smooth optimization

$$\text{minimize} \quad f(x) \qquad x \in \mathbf{R}^n$$

f is differentiable

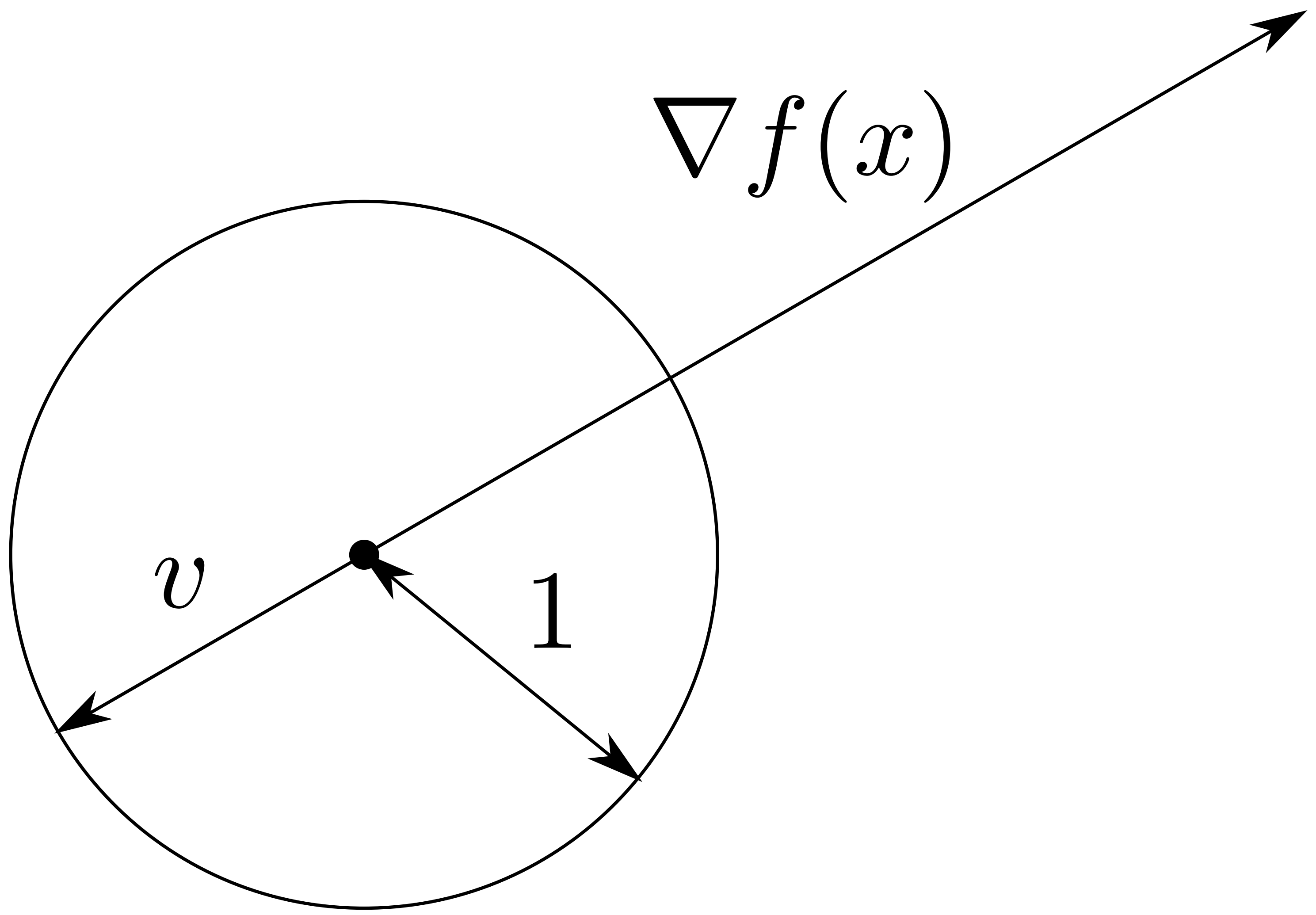
General descent scheme

Iterations

- Pick descent direction d^k, i.e., $\nabla f(x^k)^T d^k < 0$
- Pick step size t_k
- $x^{k+1} = x^k + t^k d^k, \quad k = 0, 1, \ldots$

Gradient descent
[Cauchy 1847]

Choose $d_k = -\nabla f(x^k)$



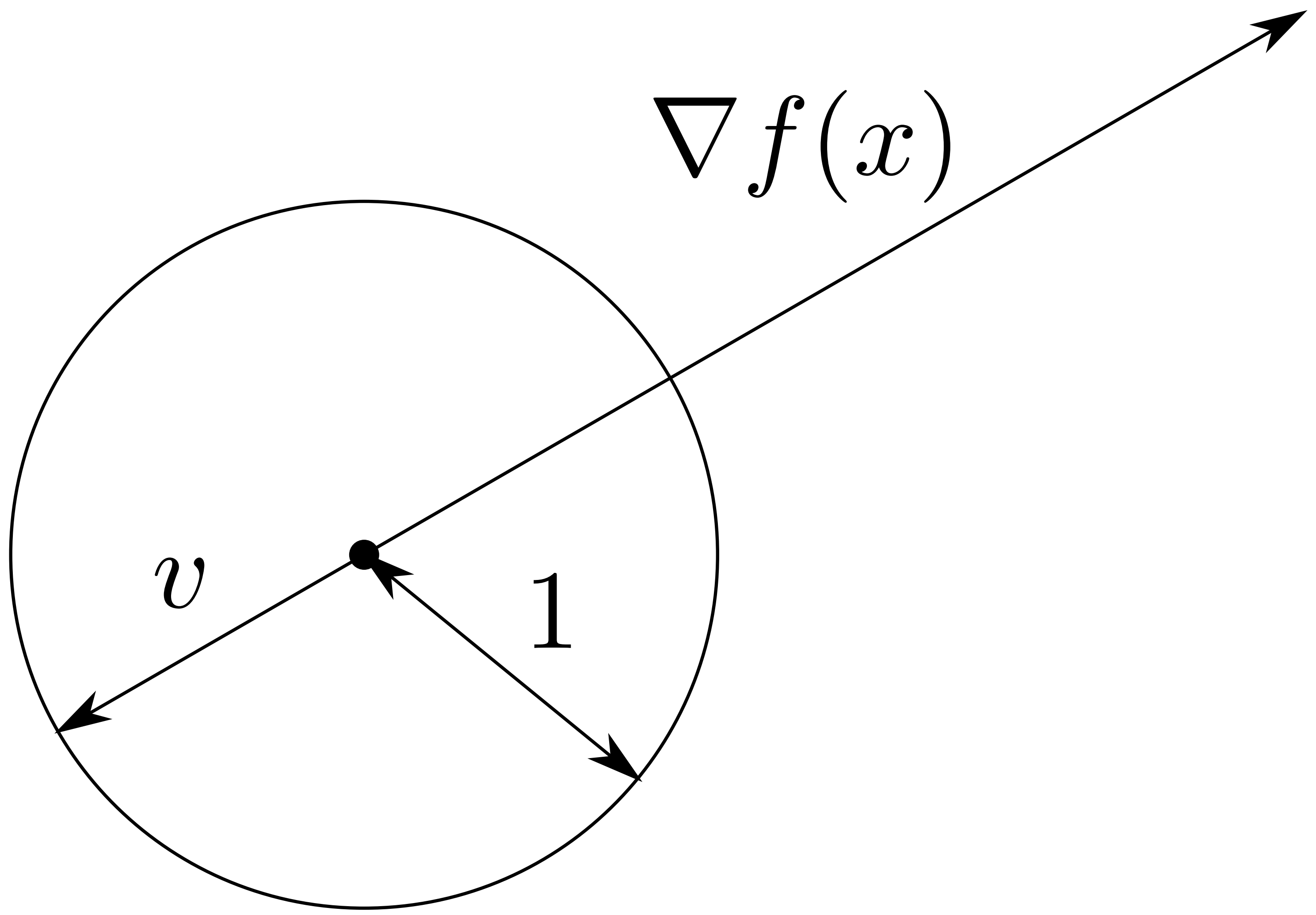
Interpretation: steepest descent (Cauchy-Schwarz)

$$\underset{\|v\|_2 \le 1}{\text{argmin}}\ \nabla f(x)^T v = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \longrightarrow d = v\|v\|_2$$

Gradient descent

[Cauchy 1847]

Choose $d_k = -\nabla f(x^k)$



Interpretation: steepest descent (Cauchy-Schwarz)

$$\underset{\|v\|_2 \leq 1}{\text{argmin}}\ \nabla f(x)^T v = -\frac{\nabla f(x)}{\|\nabla f(x)\|_2} \longrightarrow$$

$d = v \|\nabla f(x)\|_2$

Iterations

$$x^{k+1} = x^k - t_k \nabla f(x^k), \quad k = 0, 1, \ldots$$

(very cheap iterations)

Quadratic function interpretation

Quadratic approximation, replacing Hessian $\nabla^2 f(x^k)$ with $\dfrac{1}{t_k} I$

$$x^{k+1} = \underset{y}{\text{argmin}}\ f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2t_k} \|y - x^k\|_2^2$$

Set gradient with respect to y to 0...

$$x^{k+1} = x^k - t_k \nabla f(x^k)$$

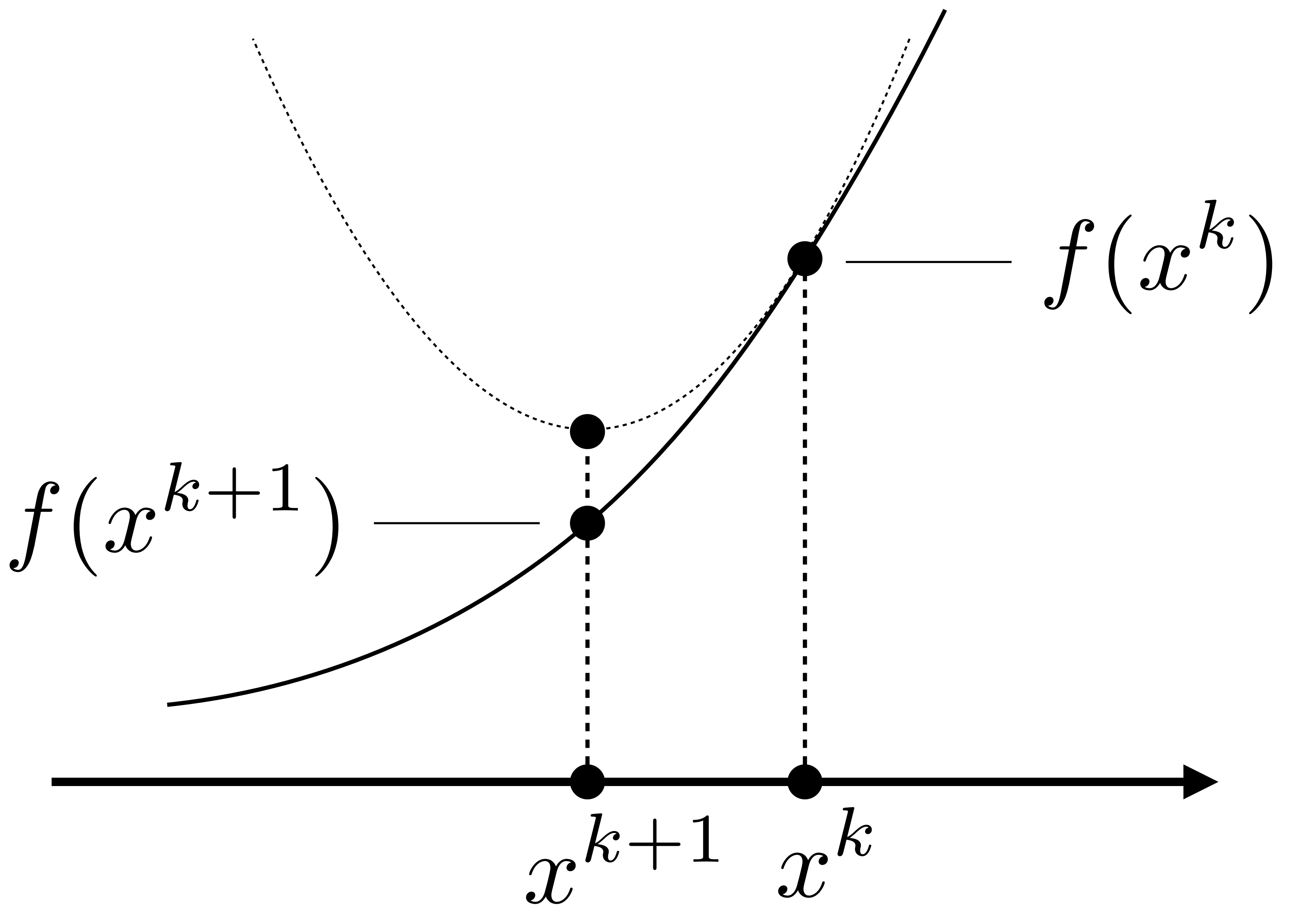
Quadratic function interpretation

Quadratic approximation, replacing Hessian $\nabla^2 f(x^k)$ with $\dfrac{1}{t_k} I$

$$x^{k+1} = \underset{y}{\text{argmin}}\ f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2t_k}\|y - x^k\|_2^2 \quad \text{(proximity to } x^k\text{)}$$

Set gradient with respect to y to 0...

$$x^{k+1} = x^k - t_k \nabla f(x^k)$$

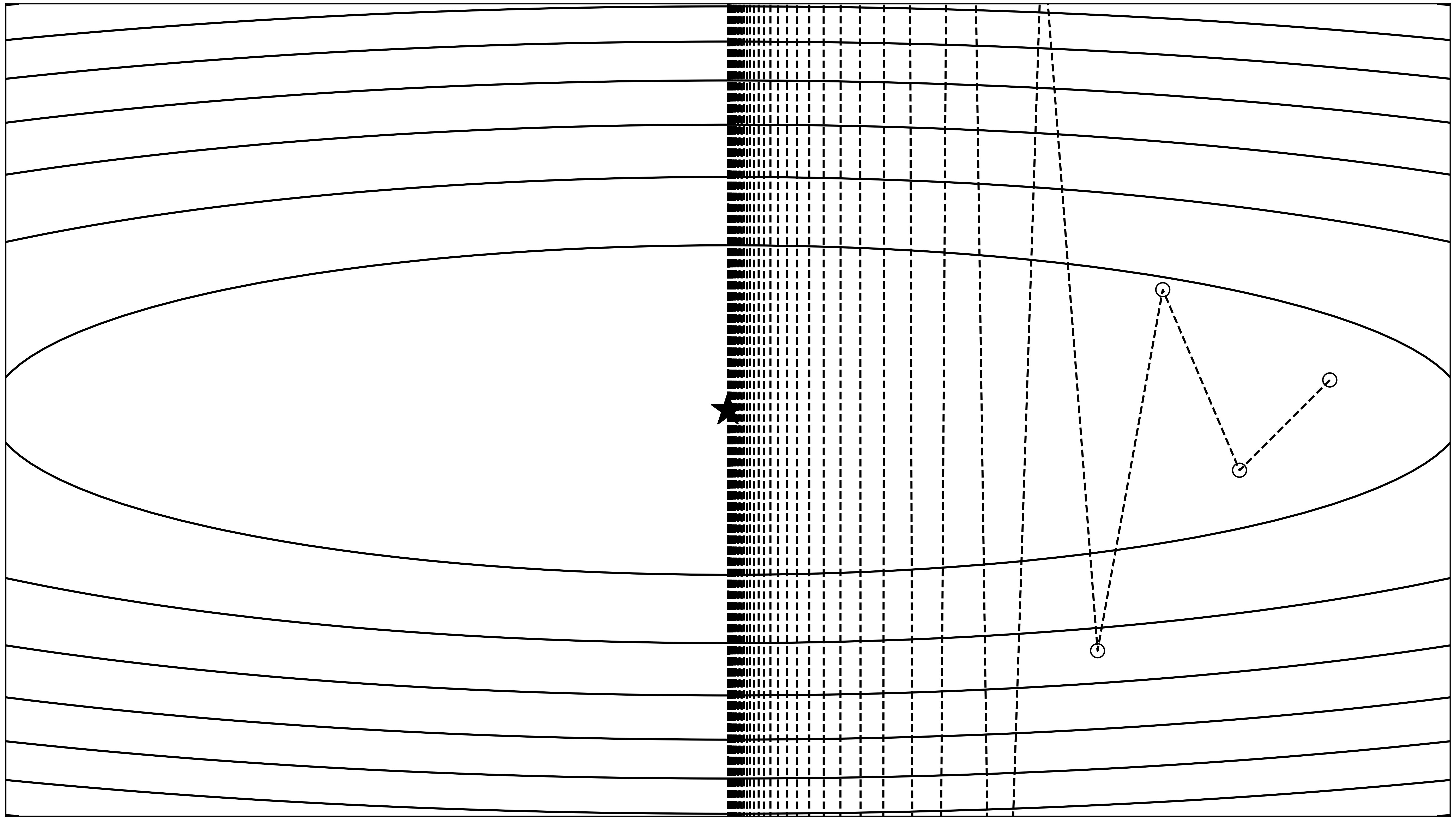


Fixed step size

Fixed step size

$$t_k = t \text{ for all } k = 0, 1, \ldots$$

$$f(x) = (x_1^2 + 20x_2^2)/2$$



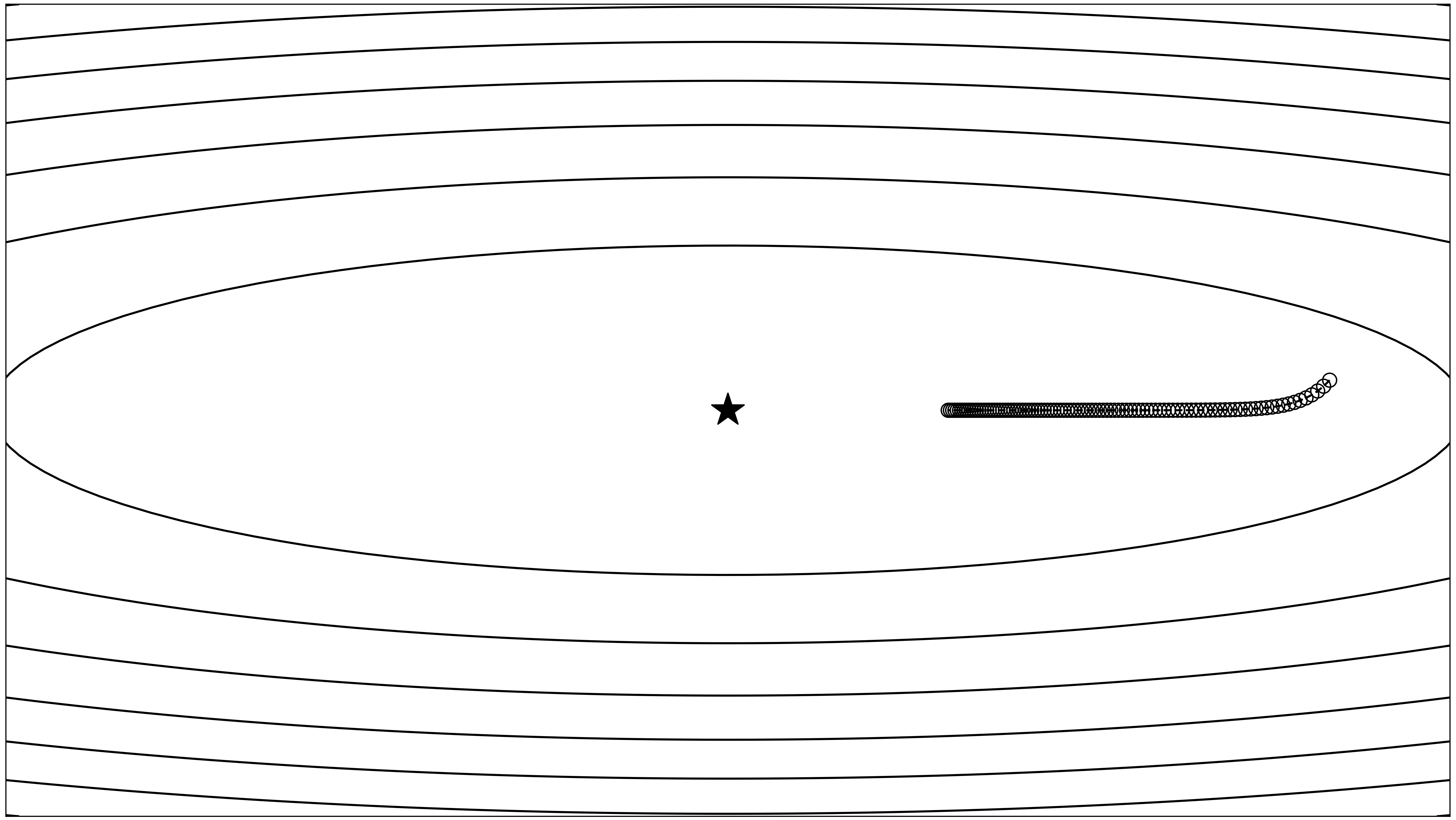
$$x^0 = (20, 1)$$
$$t = 0.15$$

It diverges

Fixed step size

$$t_k = t \text{ for all } k = 0, 1, \ldots$$

$$f(x) = (x_1^2 + 20x_2^2)/2$$



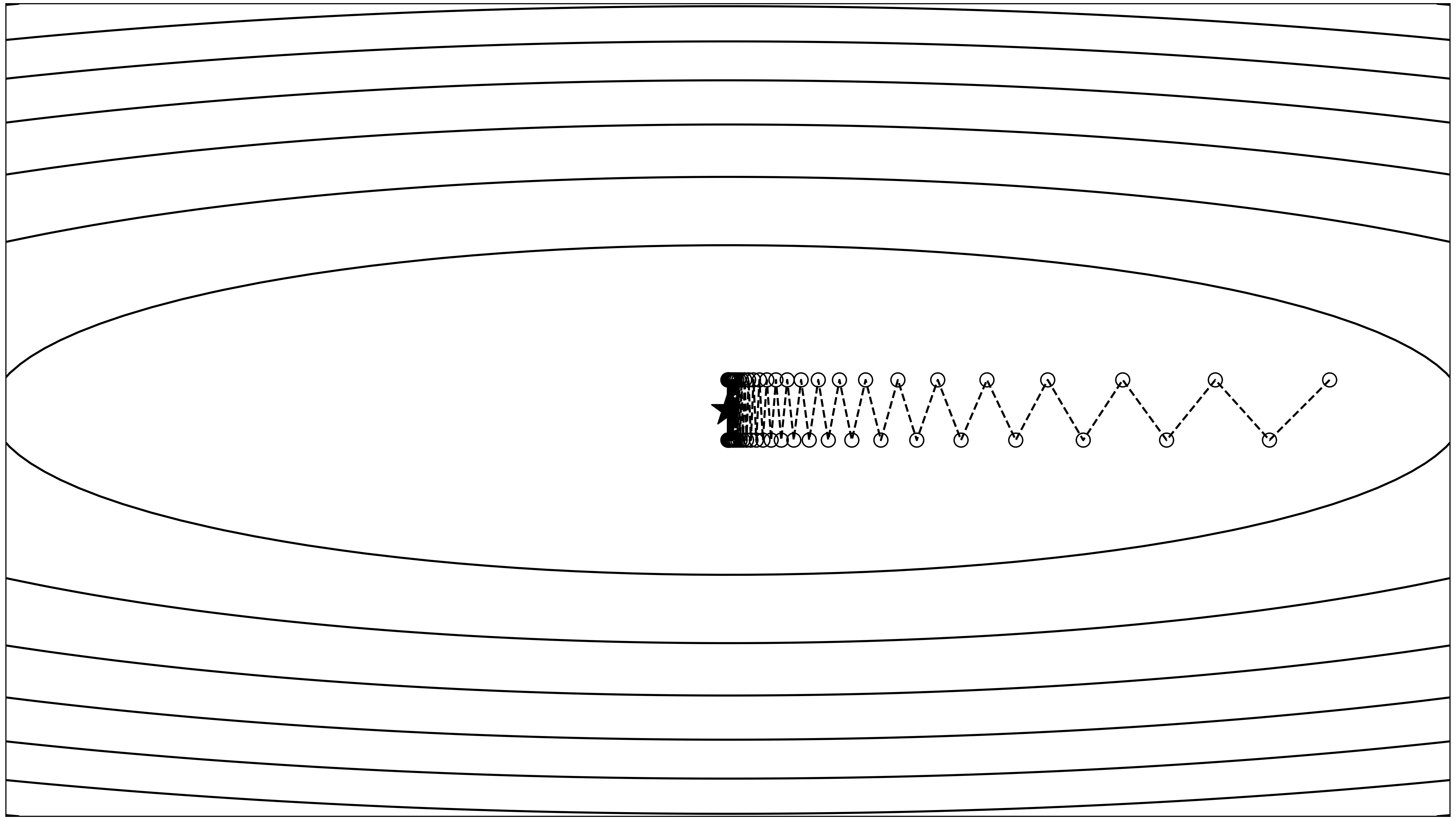
$$x^0 = (20, 1)$$
$$t = 0.01$$

too slow

Fixed step size

$t_k = t$ for all $k = 0, 1, \ldots$

$f(x) = (x_1^2 + 20x_2^2)/2$



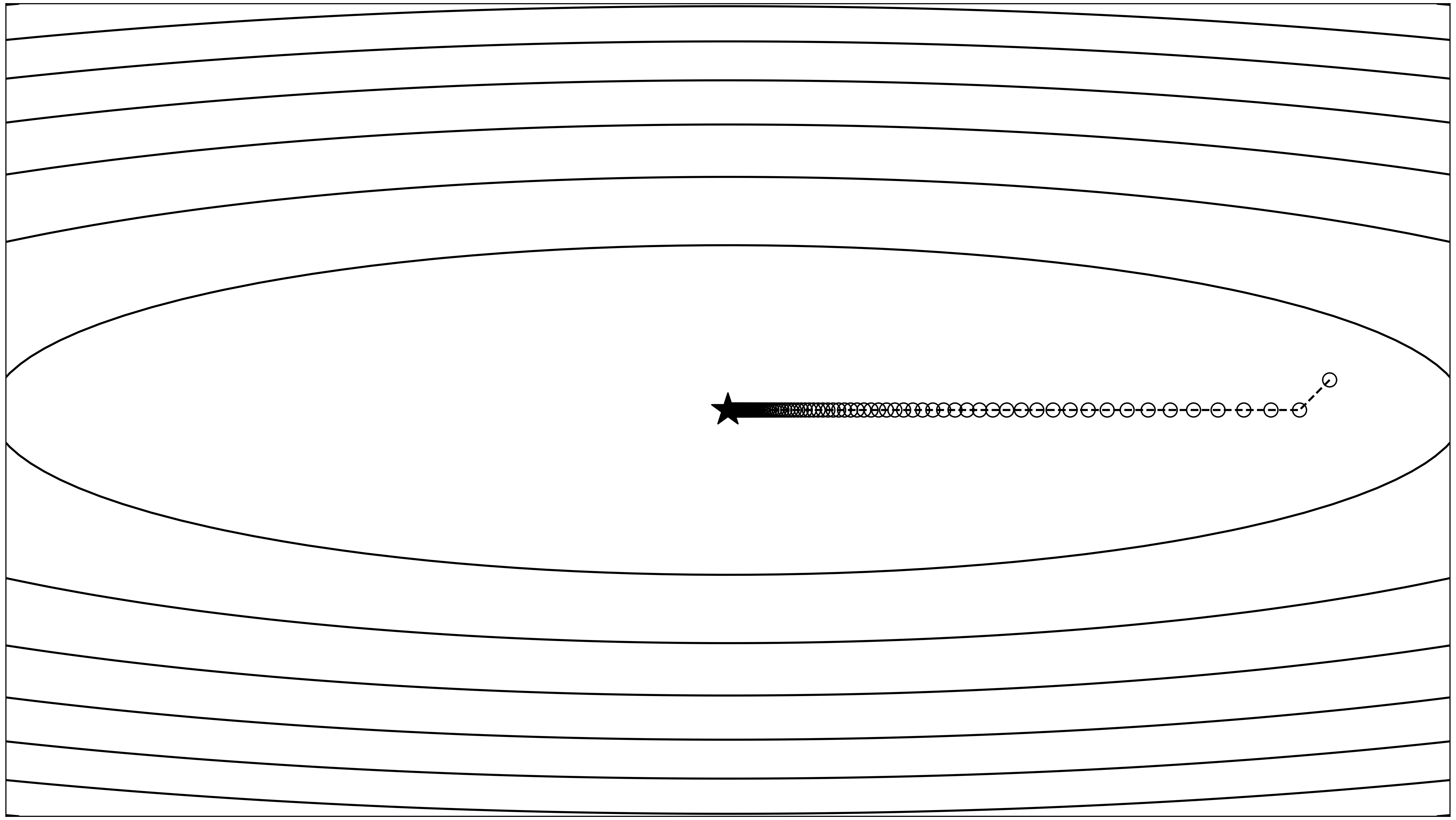
$x^0 = (20, 1)$
$t = 0.10$

it oscillates

Fixed step size

$$t_k = t \text{ for all } k = 0, 1, \ldots$$

$$f(x) = (x_1^2 + 20x_2^2)/2$$



$$x^0 = (20, 1)$$

$$t = 0.05$$

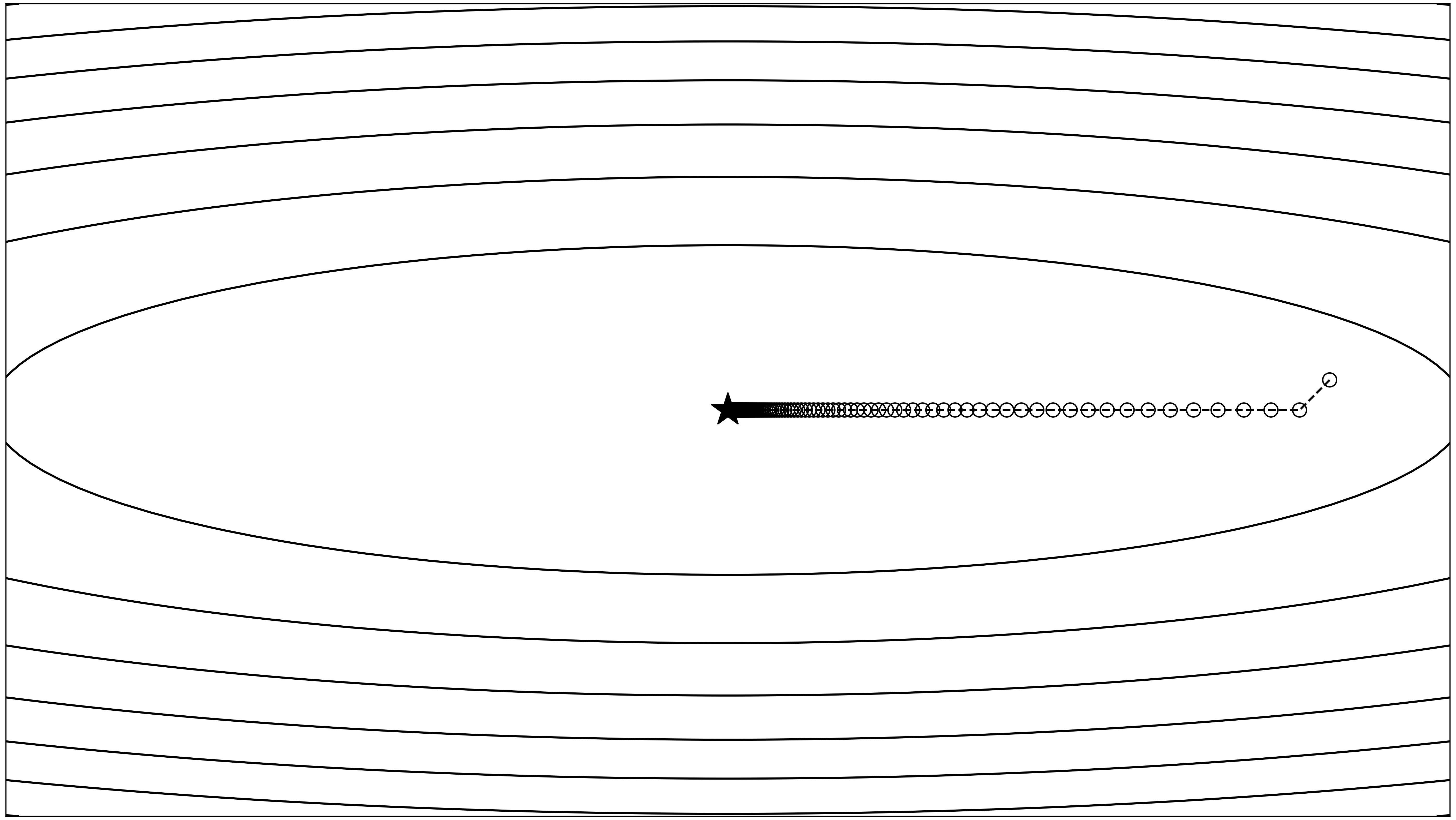
just right!

It converges in 149 iterations

Fixed step size

$$t_k = t \text{ for all } k = 0, 1, \ldots$$

$$f(x) = (x_1^2 + 20x_2^2)/2$$



$$x^0 = (20, 1)$$

$$t = 0.05$$

just right!

It converges in 149 iterations

How do we find the best one?

Quadratic optimization

Quadratic optimization

$\frac{1}{2}x^TPx + q^Tx + r$

minimize $\quad f(x) = \frac{1}{2}(x - x^\star)^T P(x - x^\star)$

where $P \succ 0$

$$\nabla f(x) = P(x - x^\star)$$

Study behavior of

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

Quadratic optimization

minimize $\quad f(x) = \dfrac{1}{2}(x - x^\star)^T P(x - x^\star)$

where $P \succ 0$

$$\nabla f(x) = P(x - x^\star)$$

Study behavior of

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

Remarks

- Always possible to write QPs in this form
- Important for smooth nonlinear programming. Close to $x^\star$, $\nabla f(x^\star) = 0$ and $\nabla^2 f(x^\star)$ dominates other terms of the Taylor expansion.

Quadratic optimization convergence

Theorem

If $t_k = t = \dfrac{2}{\lambda_{\min}(P) + \lambda_{\max}(P)}$, then

$$\|x^k - x^\star\|_2 \leq \left(\frac{\mathbf{cond}(P) - 1}{\mathbf{cond}(P) + 1}\right)^k \|x^0 - x^\star\|_2$$

Quadratic optimization convergence

Theorem

If $t_k = t = \dfrac{2}{\lambda_{\min}(P) + \lambda_{\max}(P)}$, then

$$\|x^k - x^\star\|_2 \leq \left(\frac{\mathbf{cond}(P) - 1}{\mathbf{cond}(P) + 1}\right)^k \|x^0 - x^\star\|_2$$

Remarks

- Linear (geometric) convergence rate: $O(\log(1/\epsilon))$ iterations
- It depends on the **condition number** of P: $\mathbf{cond}(P) = \dfrac{\lambda_{\max}(P)}{\lambda_{\min}(P)}$

Quadratic optimization convergence

Proof

Rewrite iterations using $\nabla f(x^k) = P(x^k - x^\star)$

$$x^{k+1} - x^\star = x^k - x^\star - t\nabla f(x^k) = (I - tP)(x^k - x^\star)$$

Therefore $\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$

Quadratic optimization convergence

Proof

Rewrite iterations using $\nabla f(x^k) = P(x^k - x^\star)$

$$x^{k+1} - x^\star = x^k - x^\star - t\nabla f(x^k) = (I - tP)(x^k - x^\star)$$

Therefore $\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$

Let's rewrite $\|I - tP\|_2$:

Matrix norm: $\|M\|_2 = \max_i |\lambda_i(M)|$

Quadratic optimization convergence

Proof

Rewrite iterations using $\nabla f(x^k) = P(x^k - x^\star)$

$$x^{k+1} - x^\star = x^k - x^\star - t\nabla f(x^k) = (I - tP)(x^k - x^\star)$$

Therefore $\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$

Let's rewrite $\|I - tP\|_2$:

Matrix norm: $\|M\|_2 = \max_i |\lambda_i(M)|$

Decomposition: $I - tP = U\mathbf{diag}(\mathbf{1} - t\lambda)U^T$ where $P = U\mathbf{diag}(\lambda)U^T$

Quadratic optimization convergence

Proof

Rewrite iterations using $\nabla f(x^k) = P(x^k - x^\star)$

$$x^{k+1} - x^\star = x^k - x^\star - t\nabla f(x^k) = (I - tP)(x^k - x^\star)$$

Therefore $\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$

Let's rewrite $\|I - tP\|_2$:

Matrix norm: $\|M\|_2 = \max_i |\lambda_i(M)|$

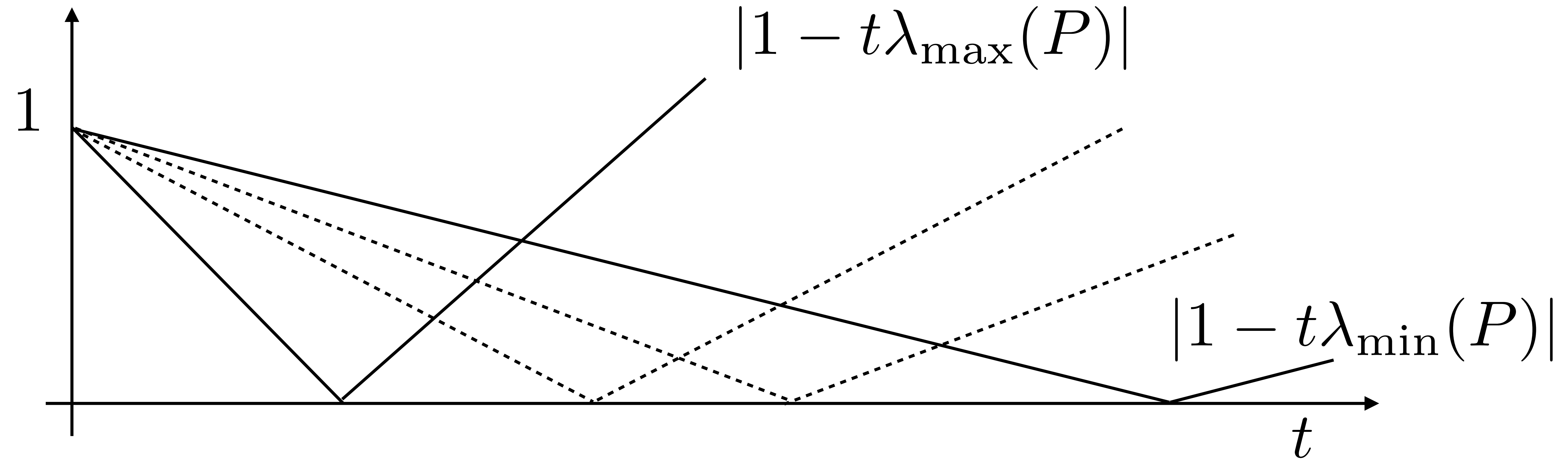
Decomposition: $I - tP = U\mathbf{diag}(\mathbf{1} - t\lambda)U^T$ where $P = U\mathbf{diag}(\lambda)U^T$

Therefore, $\|I - tP\|_2 = \max_i |1 - t\lambda_i(P)|$

Quadratic optimization convergence

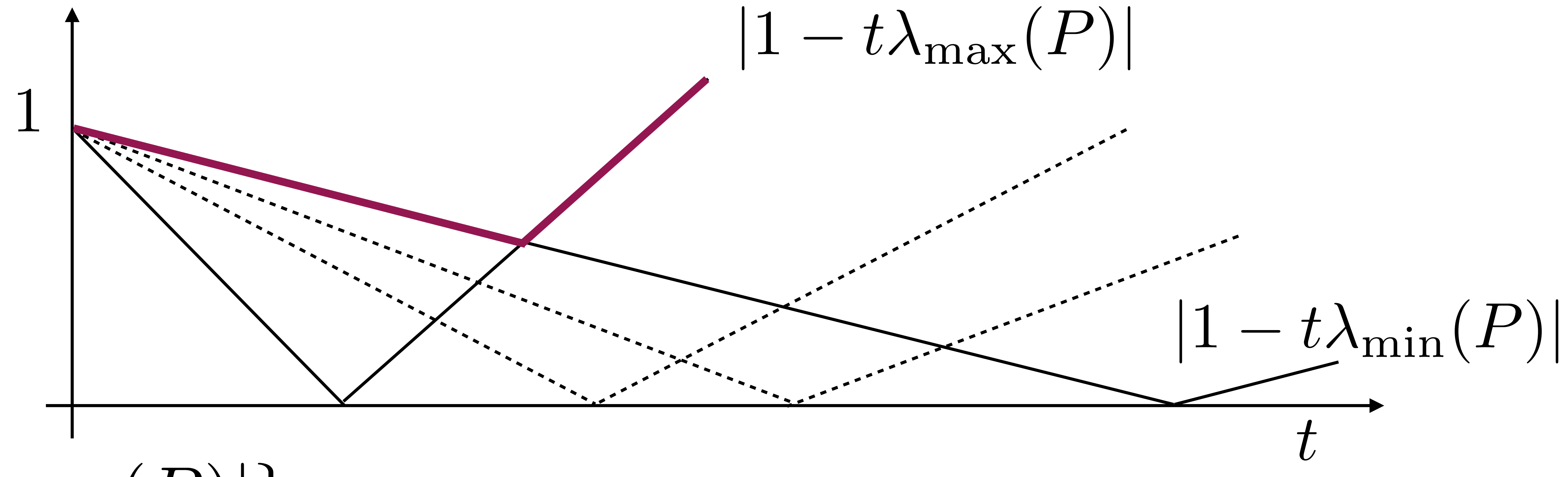
Proof (continued)

$$\|I - tP\|_2 = \max_i |1 - t\lambda_i(P)|$$



Quadratic optimization convergence

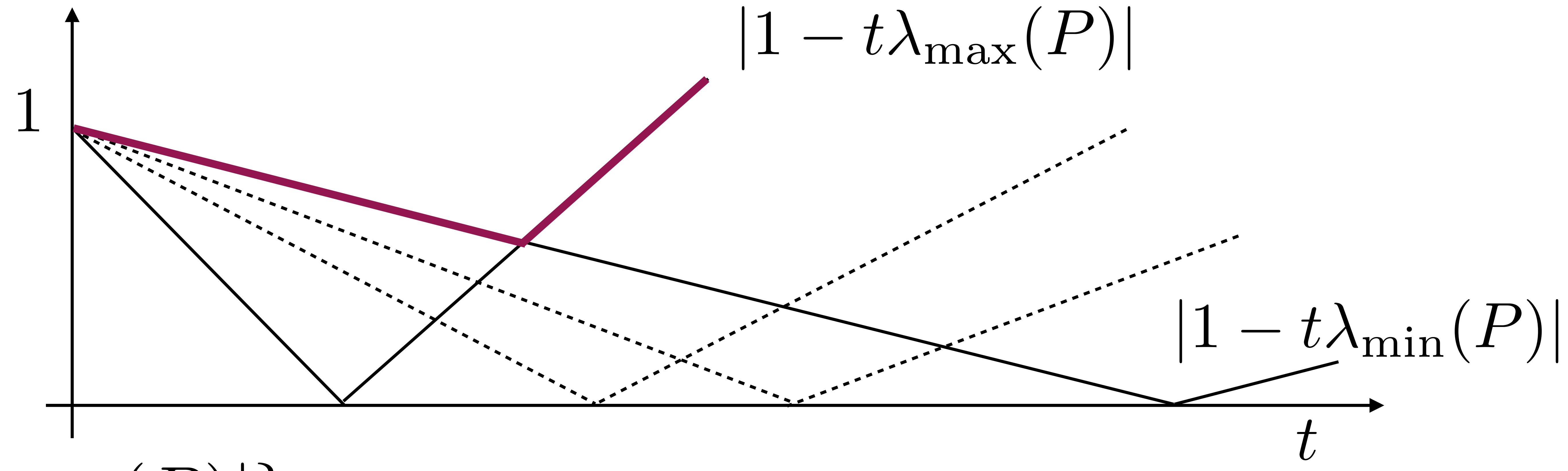
Proof (continued)



$$
\begin{aligned}
\|I - tP\|_2 &= \max_i |1 - t\lambda_i(P)| \\
&= \max\{|1 - t\lambda_{\max}(P)|, |1 - t\lambda_{\min}(P)|\} \\
&= \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}
\end{aligned}
$$

Quadratic optimization convergence

Proof (continued)



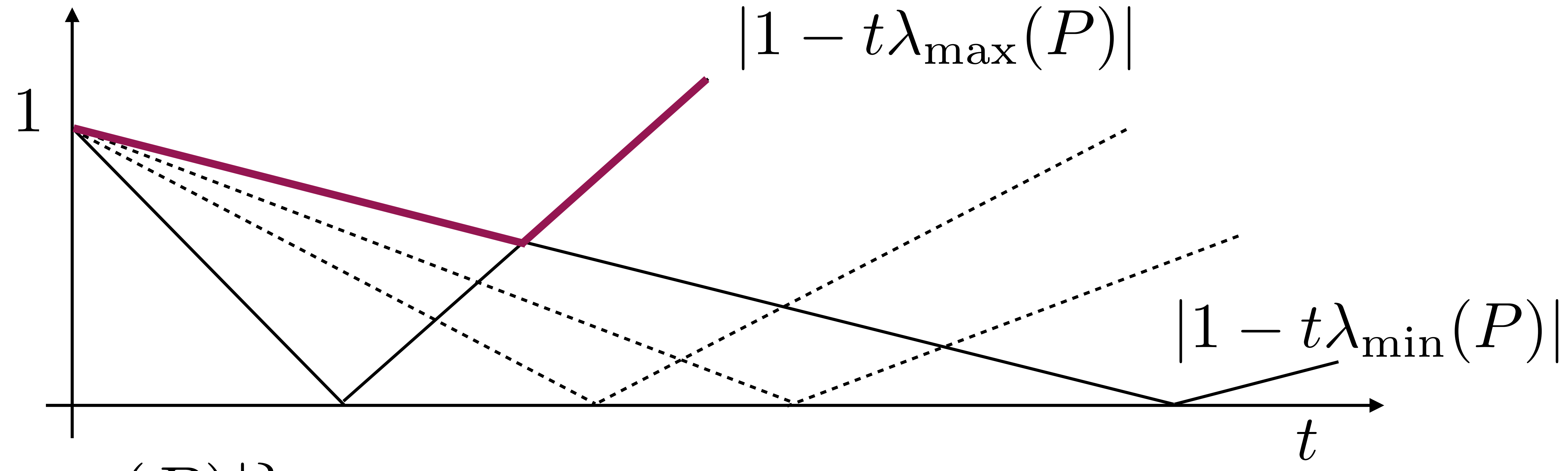
$$
\begin{aligned}
\|I - tP\|_2 &= \max_i |1 - t\lambda_i(P)| \\
&= \max\{|1 - t\lambda_{\max}(P)|, |1 - t\lambda_{\min}(P)|\} \\
&= \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}
\end{aligned}
$$

To have the fastest convergence, we want to minimize

$$\min_t \|I - tP\|_2 = \min_t \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$

Quadratic optimization convergence

Proof (continued)



$$\begin{aligned}\|I - tP\|_2 &= \max_i |1 - t\lambda_i(P)| \\ &= \max\{|1 - t\lambda_{\max}(P)|, |1 - t\lambda_{\min}(P)|\} \\ &= \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}\end{aligned}$$

To have the fastest convergence, we want to minimize

$$\min_t \|I - tP\|_2 = \min_t \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$

Minimum achieved when

$$1 - t\lambda_{\min}(P) = -1 + t\lambda_{\max}(P) \quad \Longrightarrow \quad t = \frac{2}{\lambda_{\max}(P) + \lambda_{\min}(P)}$$

Quadratic optimization convergence

Proof (continued)

$$\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$$

with $t = \dfrac{2}{\lambda_{\max}(P) + \lambda_{\min}(P)}$ we have

$$\|I - tP\|_2 = 1 - t\lambda_{\min}(P) = \frac{\lambda_{\max}(P) - \lambda_{\min}(P)}{\lambda_{\max}(P) + \lambda_{\min}(P)} = \left(\frac{\mathbf{cond}(P) - 1}{\mathbf{cond}(P) + 1}\right)$$

Quadratic optimization convergence

Proof (continued)

$$\|x^{k+1} - x^\star\|_2 \leq \|I - tP\|_2 \|x^k - x^\star\|_2$$

with $t = \dfrac{2}{\lambda_{\max}(P) + \lambda_{\min}(P)}$ we have

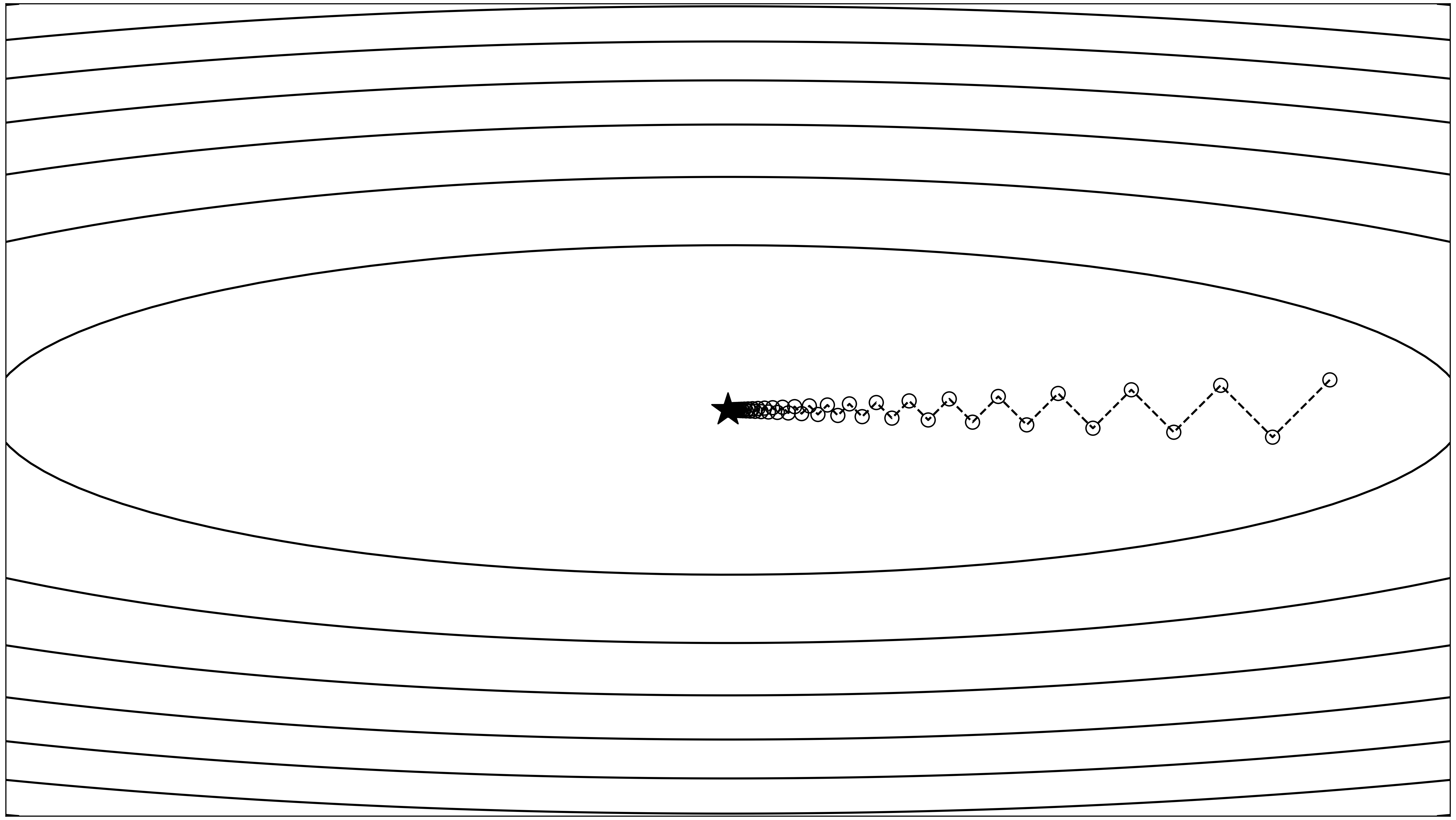
$$\|I - tP\|_2 = 1 - t\lambda_{\min}(P) = \frac{\lambda_{\max}(P) - \lambda_{\min}(P)}{\lambda_{\max}(P) + \lambda_{\min}(P)} = \left(\frac{\mathbf{cond}(P) - 1}{\mathbf{cond}(P) + 1}\right)$$

Apply the inequality recursively to get the result ∎

Optimal fixed step size

$$t_k = t \text{ for all } k = 0, 1, \ldots$$

$$f(x) = (x_1^2 + 20x_2^2)/2$$



$$x^0 = (20, 1)$$

$$t = 2/(1+20) = 0.0952$$

Optimal step size

It converges in 80 iterations

$-1 + t\lambda_{max}(P) < 1$

$t < 2/\lambda_{max}(P)$

When does it converge?

Iterations

$$\|x^k - x^\star\|_2 \leq c^k \|x^0 - x^\star\|_2$$

Contraction factor

$$c = \|I - tP\|_2 = \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$

If $t < 2/\lambda_{\max}(P)$ then $c < 1$

When does it converge?

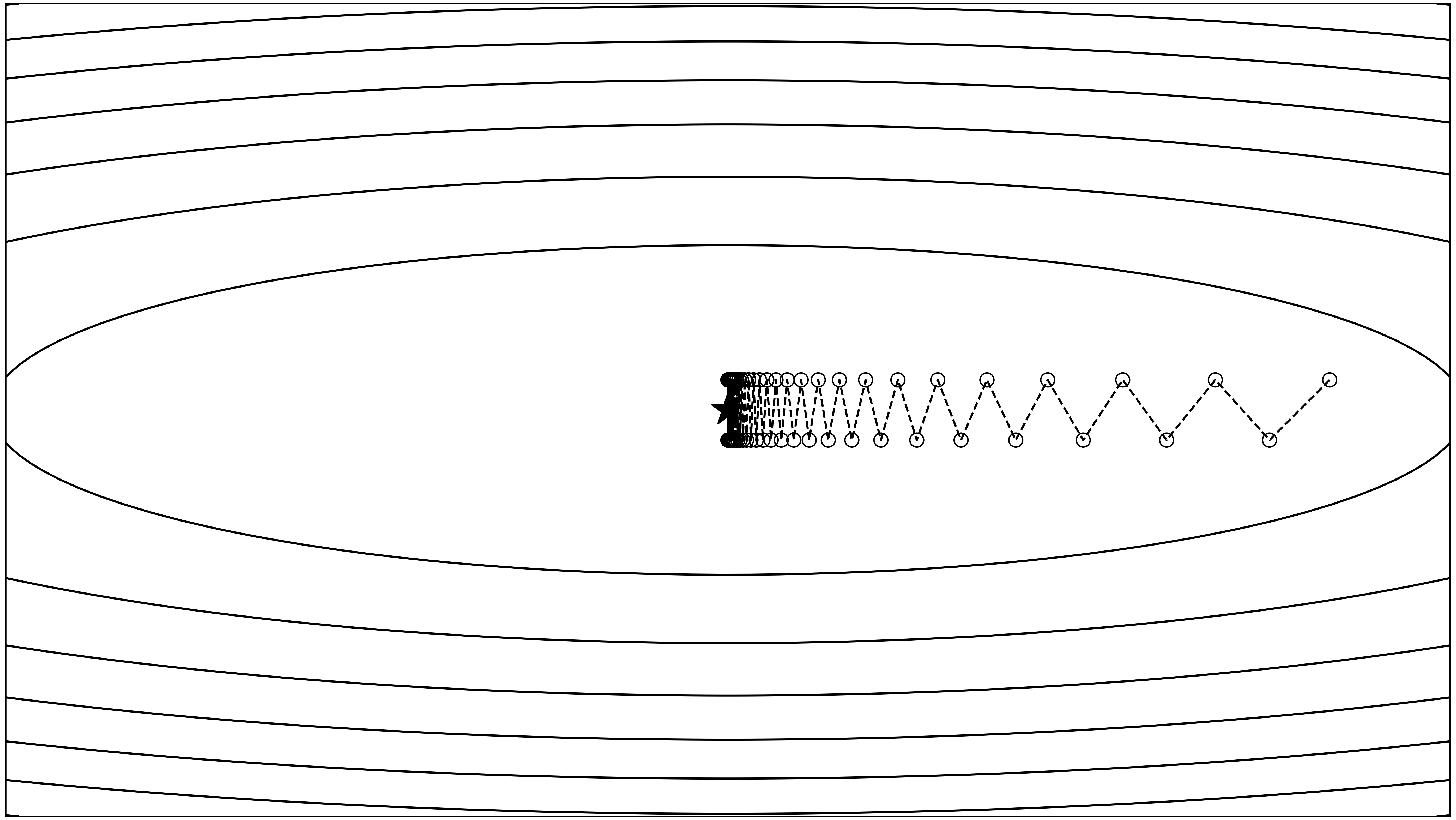
Iterations

$$\|x^k - x^\star\|_2 \le c^k \|x^0 - x^\star\|_2$$

Contraction factor

$$c = \|I - tP\|_2 = \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$

If $t < 2/\lambda_{\max}(P)$ then $c < 1$



Oscillating case

$$f(x) = (x_1^2 + 20x_2^2)/2$$

$$t = 0.1 = 2/20 = 2/\lambda_{\max}(P)$$

When does it converge?

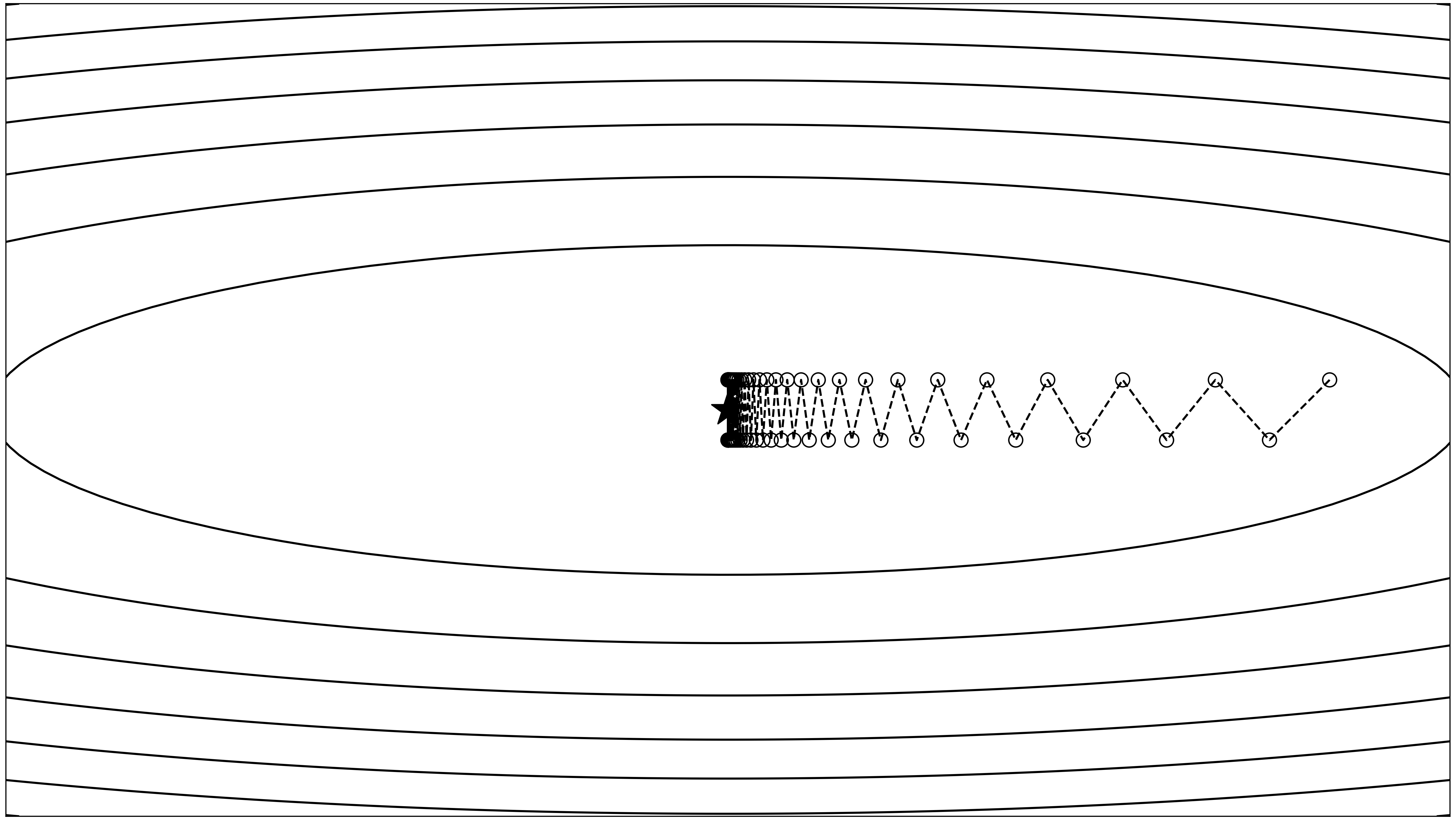
Iterations

$$\|x^k - x^\star\|_2 \leq c^k \|x^0 - x^\star\|_2$$

Contraction factor

$$c = \|I - tP\|_2 = \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$

If $t < 2/\lambda_{\max}(P)$ then $c < 1$



Oscillating case

$$f(x) = (x_1^2 + 20x_2^2)/2$$

$$t = 0.1 = 2/20 = 2/\lambda_{\max}(P)$$

Step size ranges

- If $t < 0.1$, it converges
- If $t = 0.1$, it oscillates
- If $t > 0.1$, it diverges

Strongly convex and smooth problems

Smooth functions

A convex function f is L-smooth if

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} \|x - y\|_2^2, \quad \forall x, y$$

Smooth functions

A convex function f is L-smooth if

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2}\|x - y\|_2^2, \quad \forall x, y$$

First-order characterization

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y$$

(Lipschitz continuous gradient)

Smooth functions

A convex function f is L-smooth if

$$f(y) \leq f(x) + \nabla f(x)^T (y - x) + \frac{L}{2} \|x - y\|_2^2, \quad \forall x, y$$

First-order characterization

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y$$

(Lipschitz continuous gradient)

Second-order characterization

$$\nabla^2 f(x) \preceq LI, \quad \forall x$$

Gradient monotonicity for convex functions

A differentiable function f is convex if and only if $\textbf{dom}\ f$ is convex and

$$(\nabla f(x) - \nabla f(y))^T (x - y) \geq 0, \quad \forall x, y$$

i.e., the gradient is a **monotone mapping.**

Proof (only $\Rightarrow$)

Combine $f(y) \geq f(x) + \nabla f(x)^T (y - x)$ and $f(x) \geq f(y) + \nabla f(y)^T (x - y)$ ■

Strongly convex functions

A function f is μ-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2} \|x - y\|_2^2, \quad \forall x, y$$

Strongly convex functions

A function f is μ-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^T(y - x) + \frac{\mu}{2}\|x - y\|_2^2, \quad \forall x, y$$

First-order characterization

$(\nabla f(x) - \nabla f(y))^T(x - y) \geq \mu\|x - y\|^2, \quad \forall x, y$ (strongly monotone gradient)

Strongly convex functions

A function f is μ-strongly convex if

$$f(y) \geq f(x) + \nabla f(x)^T (y - x) + \frac{\mu}{2}\|x - y\|_2^2, \quad \forall x, y$$

First-order characterization

$(\nabla f(x) - \nabla f(y))^T (x - y) \geq \mu \|x - y\|^2, \quad \forall x, y$ (strongly monotone gradient)

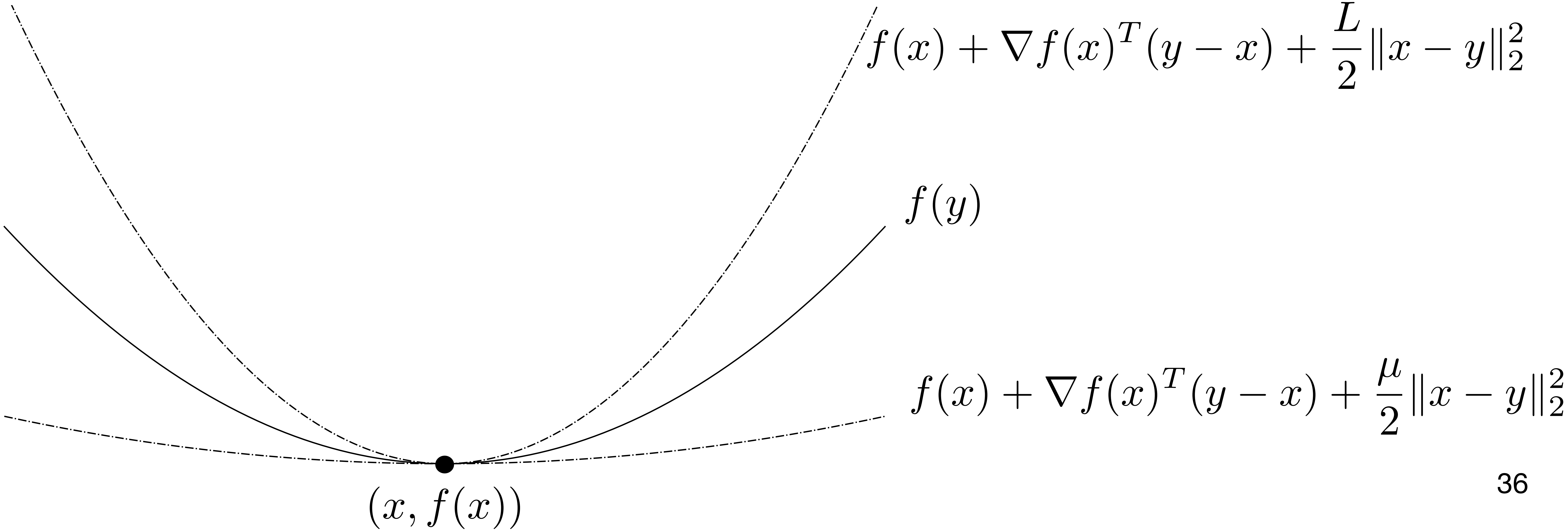
Second-order characterization

$\nabla^2 f(x) \succeq \mu I, \quad \forall x$

Strongly convex and smooth functions

f is μ**-strongly convex** and L**-smooth** if

$$0 \preceq \mu I \preceq \nabla^2 f(x) \preceq LI, \quad \forall x$$



Strongly convex and smooth convergence

Theorem

Let f be μ-strongly convex and L-smooth. If $t = \dfrac{2}{\mu + L}$, then

$$\|x^k - x^\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x^0 - x^\star\|_2$$

where $\kappa = L/\mu$ is the **condition number**

Strongly convex and smooth convergence

Theorem

Let f be μ-strongly convex and L-smooth. If $t = \dfrac{2}{\mu + L}$, then

$$\|x^k - x^\star\|_2 \leq \left(\frac{\kappa - 1}{\kappa + 1}\right)^k \|x^0 - x^\star\|_2$$

where $\kappa = L/\mu$ is the **condition number**

Remarks

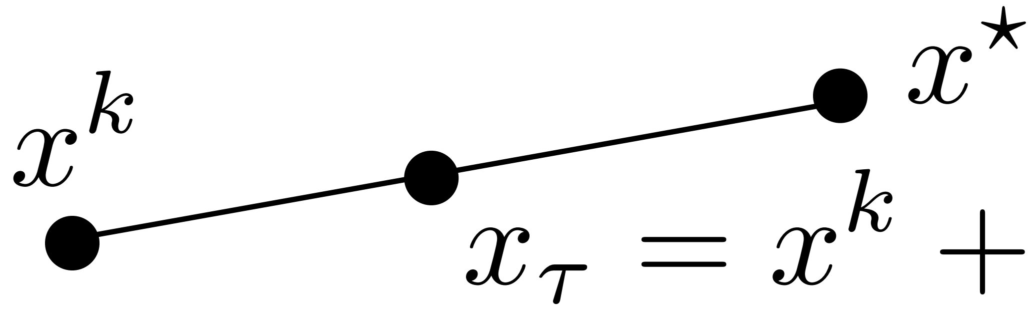
- **Linear (geometric) convergence rate** $O(\log(1/\epsilon))$ iterations
- **Generalizes quadratic problems** where
  $t = 2/(\lambda_{\max}(P) + \lambda_{\min}(P)), \quad \mathbf{cond}(P)$ instead of κ
- **Dimension-free contraction factor**, if κ does not depend on n

Strongly convex and smooth convergence

Proof

Fundamental theorem of calculus:

$$\nabla f(x^k) = \nabla f(x^k) - \underbrace{\nabla f(x^\star)}_{=0} = \int_{x^k}^{x^\star} \nabla^2 f(x_\tau) \mathrm{d}x_\tau$$



$x_\tau = x^k + \tau(x^\star - x^k)$

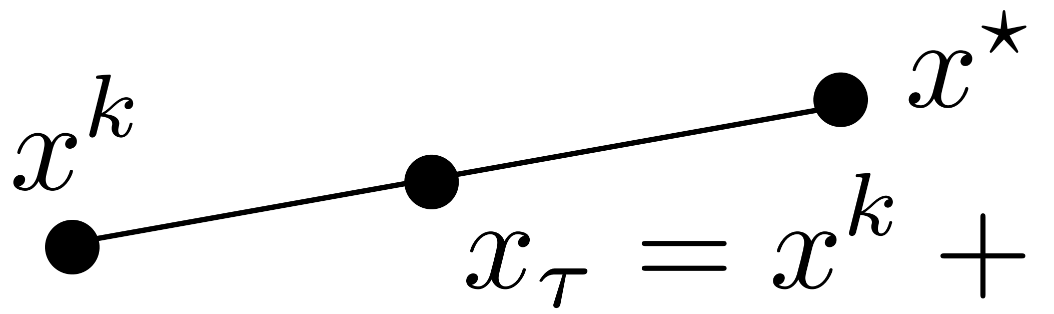
$$= \int_0^1 \nabla^2 f(x_\tau) \mathrm{d}\tau (x^k - x^\star)$$

Strongly convex and smooth convergence

Proof

Fundamental theorem of calculus:

$$\nabla f(x^k) = \nabla f(x^k) - \underbrace{\nabla f(x^\star)}_{=0} = \int_{x^k}^{x^\star} \nabla^2 f(x_\tau) \mathrm{d}x_\tau$$



$$x_\tau = x^k + \tau(x^\star - x^k)$$

$$= \int_0^1 \nabla^2 f(x_\tau) \mathrm{d}\tau (x^k - x^\star)$$

Therefore,

$$\|x^{k+1} - x^\star\|_2 = \|x^k - x^\star - t\nabla f(x^k)\|_2$$

$$= \left\| \left(\int_0^1 \left(I - t\nabla^2 f(x_\tau)\right) \mathrm{d}\tau \right) (x^k - x^\star) \right\|$$

$$\leq \max_{0 \leq \tau \leq 1} \|I - t\nabla^2 f(x_\tau)\|_2 \|x^k - x^\star\|_2$$

$$\leq \frac{L - \mu}{L + \mu} \|x^k - x^\star\|_2$$

(similar to quadratic)

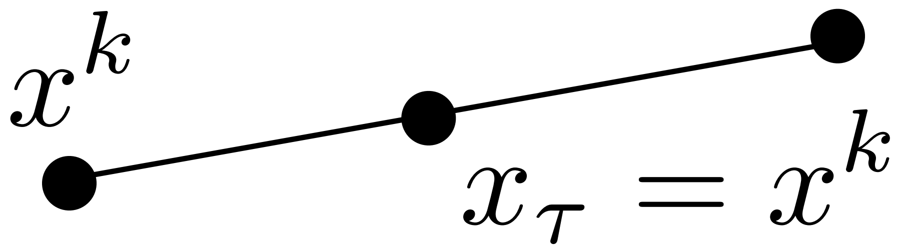
$\mu I \preceq \nabla^2 f(x_\tau) \preceq L I$

$\left(\frac{L/\mu - 1}{L/\mu + 1} \right)$

Strongly convex and smooth convergence

Proof Fundamental theorem of calculus:

$$\nabla f(x^k) = \nabla f(x^k) - \underbrace{\nabla f(x^\star)}_{=0} = \int_{x^k}^{x^\star} \nabla^2 f(x_\tau) \mathrm{d}x_\tau$$



$$x_\tau = x^k + \tau(x^\star - x^k)$$

$$= \int_0^1 \nabla^2 f(x_\tau) \mathrm{d}\tau (x^k - x^\star)$$

Therefore,

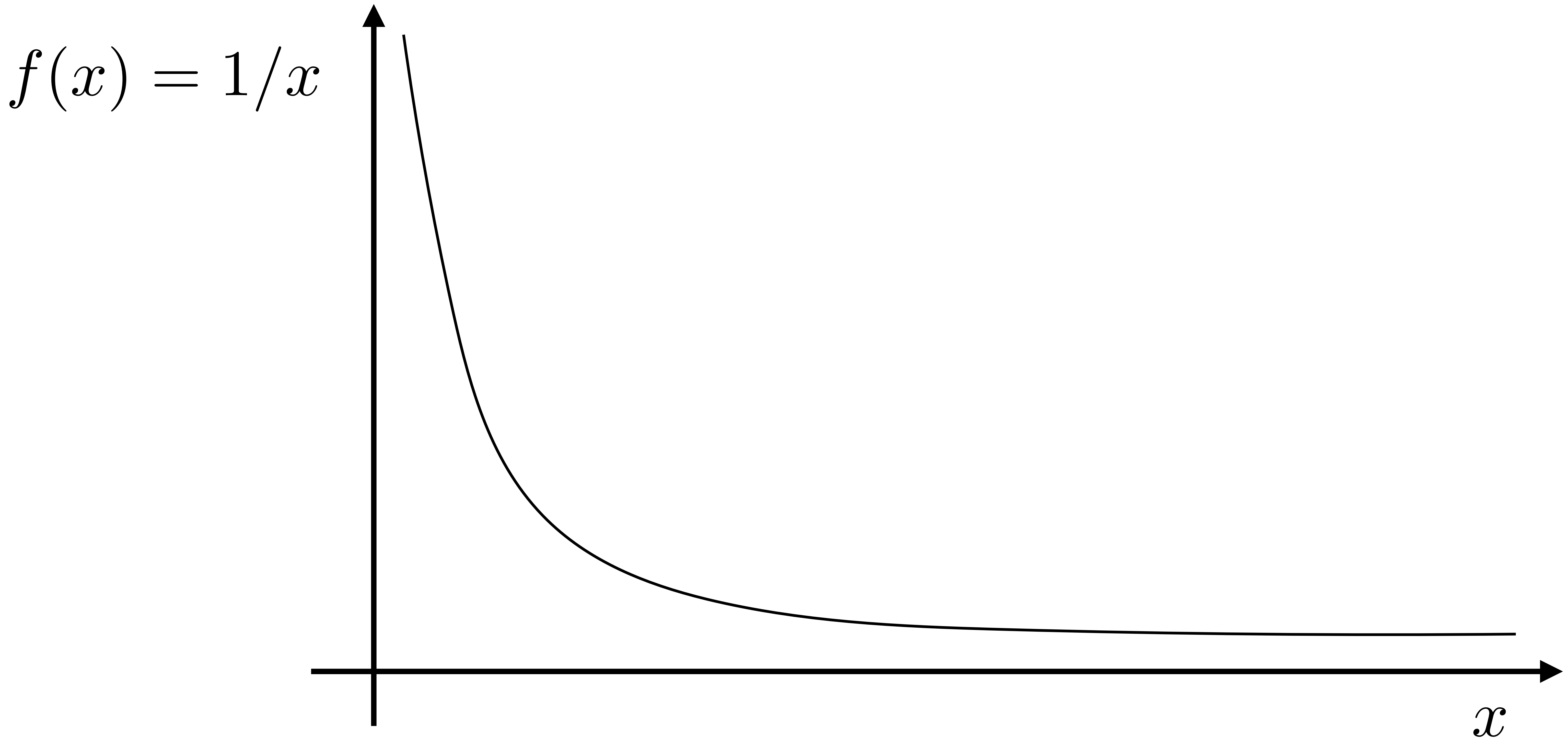
$$\begin{aligned} \|x^{k+1} - x^\star\|_2 &= \|x^k - x^\star - t\nabla f(x^k)\|_2 \\ &= \left\| \left(\int_0^1 \left(I - t\nabla^2 f(x_\tau)\right) \mathrm{d}\tau \right) (x^k - x^\star) \right\| \\ &\le \max_{0 \le \tau \le 1} \|I - t\nabla^2 f(x_\tau)\|_2 \|x^k - x^\star\|_2 \\ &\le \frac{L - \mu}{L + \mu} \|x^k - x^\star\|_2 \end{aligned}$$

(similar to quadratic)

Apply the inequality recursively to get the result ■

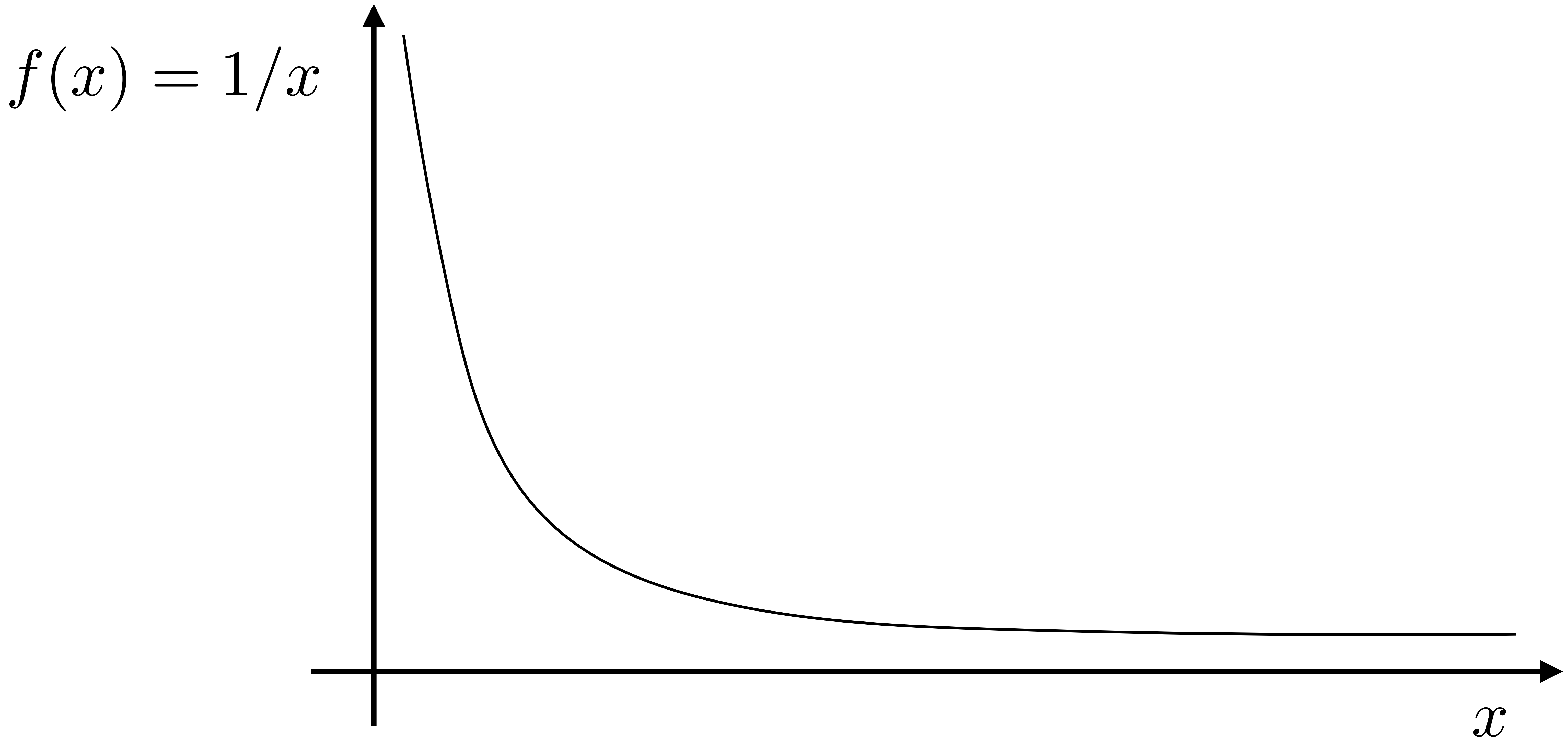
Dropping strong convexity

Many functions are not strongly convex



Without strong convexity, the optimal solution might be very far ($x^\star = \infty$) but the objective value very close

Many functions are not strongly convex



Without strong convexity, the optimal solution might be very far ($x^\star = \infty$) but the objective value very close

Focus on objective error $f(x^k) - f(x^\star)$ instead of variable error $\|x^k - x^\star\|_2$

Null growth directions without strong convexity

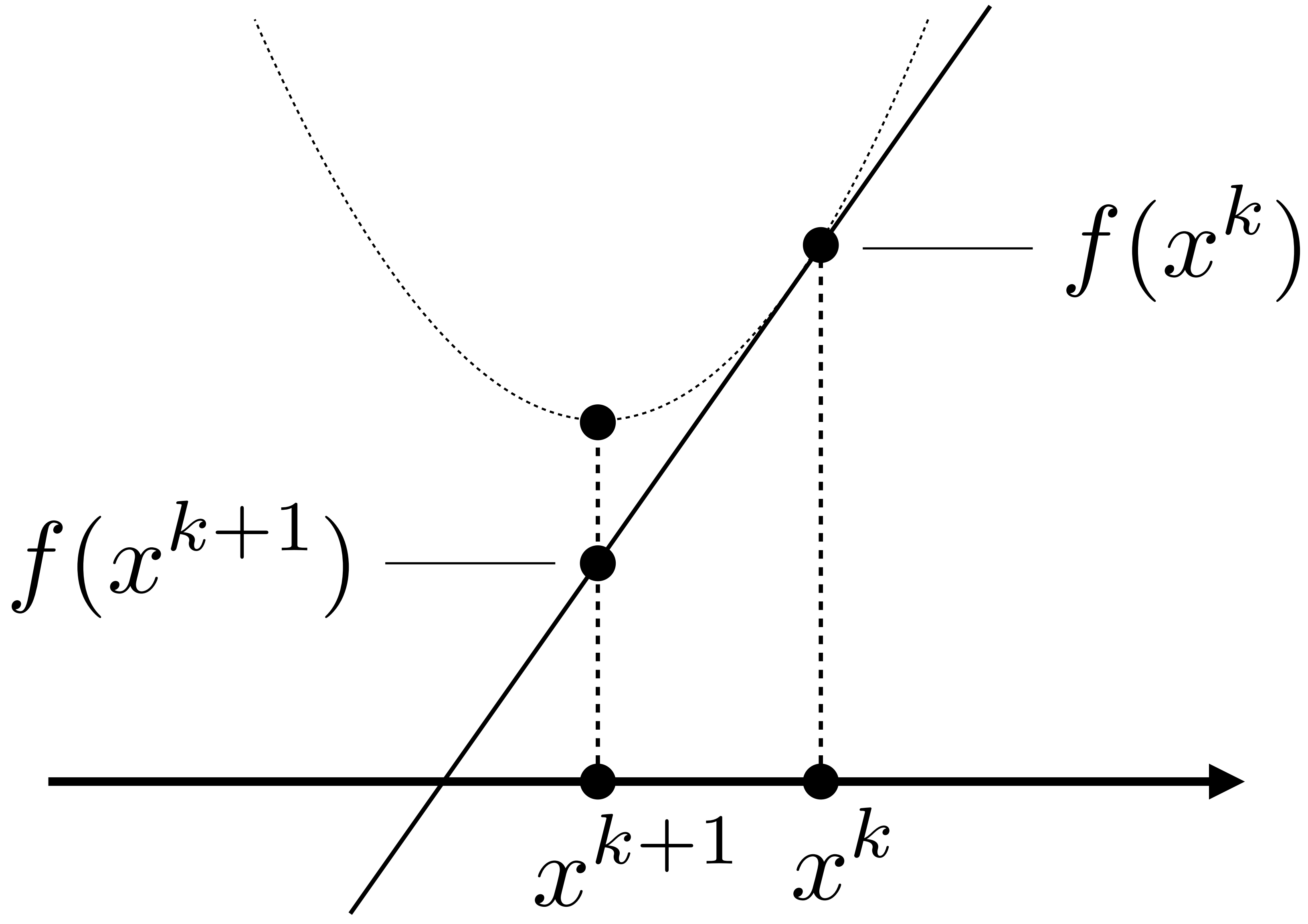
Hessian $\nabla^2 f(x)$ has some null growth directions (it can even be 0)

Null growth directions without strong convexity

Hessian $\nabla^2 f(x)$ has some null growth directions (it can even be 0)

Gradient descent interpretation: replace $\nabla^2 f(x^k)$ with $\frac{1}{t_k} I$

$$x^{k+1} = \underset{y}{\text{argmin}}\ f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2t_k}\|y - x^k\|_2^2$$

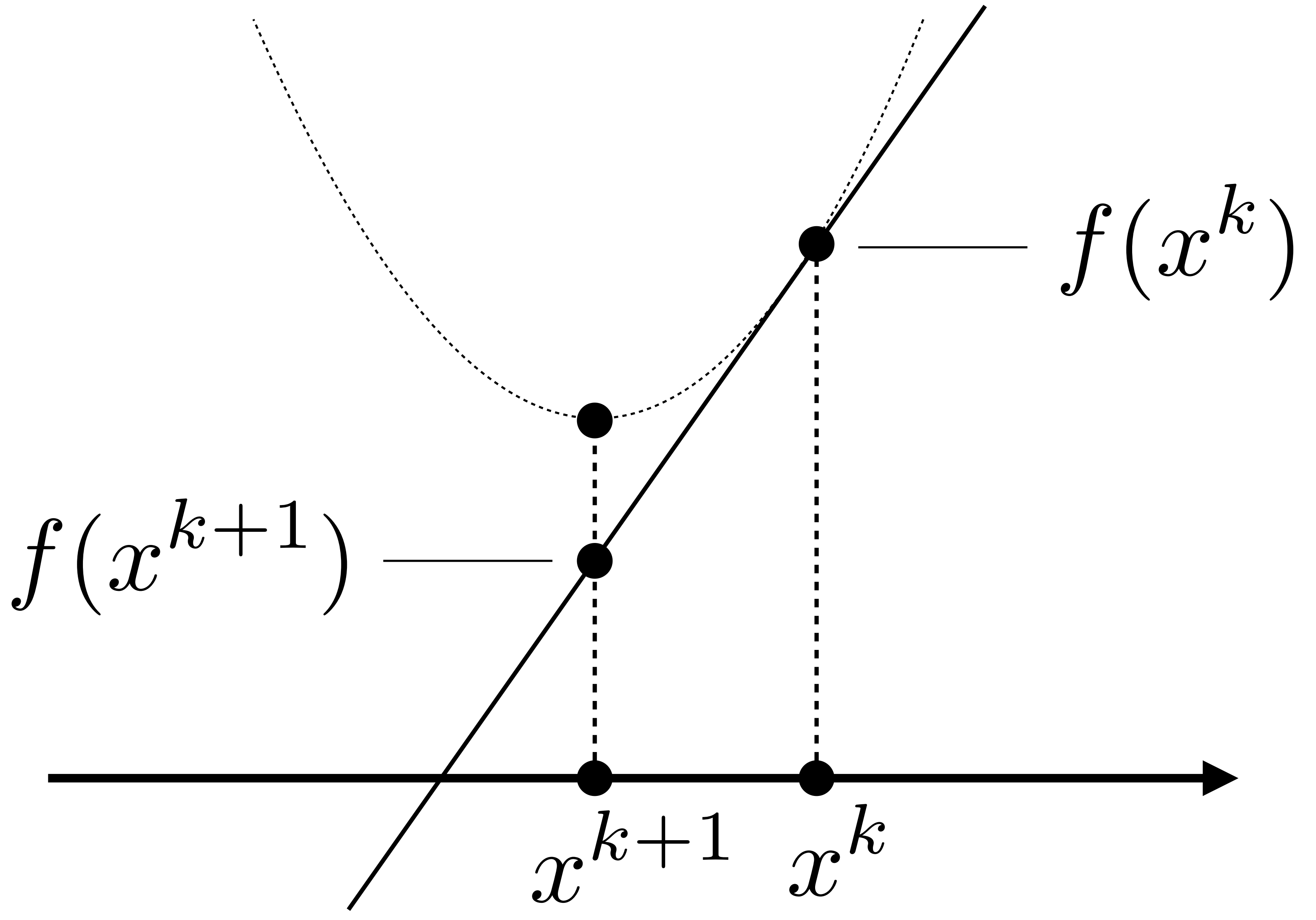


Null growth directions without strong convexity

Hessian $\nabla^2 f(x)$ has some null growth directions (it can even be 0)

Gradient descent interpretation: replace $\nabla^2 f(x^k)$ with $\dfrac{1}{t_k} I$

$$x^{k+1} = \underset{y}{\text{argmin}}\ f(x^k) + \nabla f(x^k)^T (y - x^k) + \frac{1}{2t_k}\|y - x^k\|_2^2$$



How to pick a quadratic approximation?

↓

Use L-Lipschitz smoothness

Convergence for smooth functions

Theorem

Let f be L-smooth. If $t < 1/L$ then gradient descent satisfies

$$f(x^k) - f(x^\star) \le \frac{\|x^0 - x^\star\|_2^2}{2tk}$$

Sublinear convergence rate $O(1/\epsilon)$ iterations (can be very slow!)

Convergence for smooth functions

Proof

Use L-Lipschitz constant

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L}{2}\|x^k - x^{k+1}\|_2^2$$

Convergence for smooth functions

Proof

Use L-Lipschitz constant

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L}{2}\|x^k - x^{k+1}\|_2^2$$

Plug in iterate $x^{k+1} = x^k - t\nabla f(x^k)$ in right-hand side

$$f(x^{k+1}) \leq f(x^k) - \left(1 - \frac{Lt}{2}\right) t\|\nabla f(x^k)\|_2^2$$

Convergence for smooth functions

Proof

Use L-Lipschitz constant

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T(x^{k+1} - x^k) + \frac{L}{2}\|x^k - x^{k+1}\|_2^2$$

Plug in iterate $x^{k+1} = x^k - t\nabla f(x^k)$ in right-hand side

$$f(x^{k+1}) \leq f(x^k) - \left(1 - \frac{Lt}{2}\right)t\|\nabla f(x^k)\|_2^2$$

Take $0 < t \leq 1/L$ we get $2 > 2 - Lt \geq 1$ and

$$f(x^{k+1}) \leq f(x^k) - \frac{t}{2}\|\nabla f(x^k)\|_2^2$$ **(non increasing cost)**

Convergence for smooth functions

Proof

Use L-Lipschitz constant

$$f(x^{k+1}) \leq f(x^k) + \nabla f(x^k)^T (x^{k+1} - x^k) + \frac{L}{2}\|x^k - x^{k+1}\|_2^2$$

Plug in iterate $x^{k+1} = x^k - t\nabla f(x^k)$ in right-hand side

$$f(x^{k+1}) \leq f(x^k) - \left(1 - \frac{Lt}{2}\right) t\|\nabla f(x^k)\|_2^2$$

Take $0 < t \leq 1/L$ we get $2 > 2 - Lt \geq 1$ and

$$f(x^{k+1}) \leq f(x^k) - \frac{t}{2}\|\nabla f(x^k)\|_2^2$$ **(non increasing cost)**

Note: non-increasing for any $t > 0$ such that $\left(1 - \frac{Lt}{2}\right) t > 0 \quad \Rightarrow \; t \in (0, 2/L)$

Convergence for smooth functions

Proof (continued)

Convexity of f implies $f(x^k) \leq f(x^\star) + \nabla f(x^k)^T (x^k - x^\star)$

Convergence for smooth functions

Proof (continued)

Convexity of f implies $f(x^k) \leq f(x^\star) + \nabla f(x^k)^T(x^k - x^\star)$

Therefore, we rewrite $f(x^{k+1}) \leq f(x^k) - \frac{t}{2}\|\nabla f(x^k)\|_2^2$ as

NON INCREASING CONDITION

$$
\begin{aligned}
f(x^{k+1}) - f(x^\star) &\leq \nabla f(x^k)^T(x^k - x^\star) - \frac{t}{2}\|\nabla f(x^k)\|_2^2 \\
&= \frac{1}{2t}\left(\|x^k - x^\star\|_2^2 - \|x^k - x^\star - t\nabla f(x^k)\|_2^2\right) \\
&= \frac{1}{2t}\left(\|x^k - x^\star\|_2^2 - \|x^{k+1} - x^\star\|_2^2\right)
\end{aligned}
$$

Convergence for smooth functions

Proof (continued)

Summing over the iterations with $i = 1, \dots, k$

$$\sum_{i=1}^{k} \left(f(x^i) - f(x^\star)\right) \leq \frac{1}{2t} \sum_{i=1}^{k} \left(\|x^{i-1} - x^\star\|_2^2 - \|x^i - x^\star\|_2^2\right)$$

$$= \frac{1}{2t} \left(\|x^0 - x^\star\|_2^2 - \|x^k - x^\star\|_2^2\right)$$

$$\leq \frac{1}{2t} \|x^0 - x^\star\|_2^2$$

Convergence for smooth functions

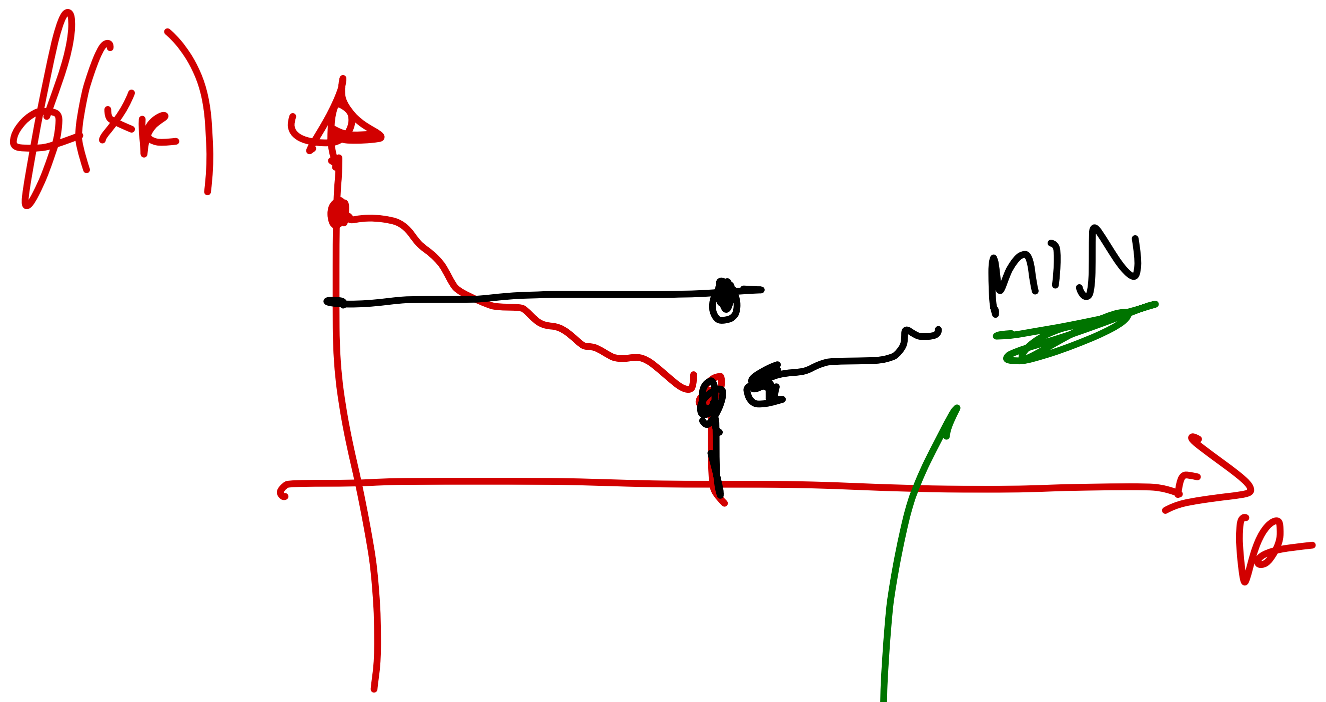
Proof (continued)

Summing over the iterations with $i = 1, \dots, k$

$$\sum_{i=1}^{k} \left(f(x^i) - f(x^\star)\right) \le \frac{1}{2t} \sum_{i=1}^{k} \left(\|x^{i-1} - x^\star\|_2^2 - \|x^i - x^\star\|_2^2\right)$$

$$= \frac{1}{2t} \left(\|x^0 - x^\star\|_2^2 - \|x^k - x^\star\|_2^2\right)$$

$$\le \frac{1}{2t} \|x^0 - x^\star\|_2^2$$



Since $f(x^k)$ is non-increasing, we have

$$f(x^k) - f(x^\star) \le \frac{1}{k} \sum_{i=1}^{k} (f(x^i) - f(x^\star)) \le \frac{1}{2kt} \|x^0 - x^\star\|_2^2 \qquad \blacksquare$$

Issues with computing the optimal step size

Quadratic programs

The rule $t = 2/(\lambda_{\max}(P) + \lambda_{\min}(P))$ can be **very expensive to compute**

It relies on eigendecomposition of P (iterative factorizations...)

[Lecture 24, 25, Numerical Linear Algebra, Trefethen and Bau]

Issues with computing the optimal step size

Quadratic programs

The rule $t = 2/(\lambda_{\max}(P) + \lambda_{\min}(P))$ can be **very expensive to compute**

It relies on eigendecomposition of P (iterative factorizations...)

Smooth and strongly convex functons

Very hard to estimate μ and L in general

[Lecture 24, 25, Numerical Linear Algebra, Trefethen and Bau]

Issues with computing the optimal step size

Quadratic programs

The rule $t = 2/(\lambda_{\max}(P) + \lambda_{\min}(P))$ can be **very expensive to compute**

It relies on eigendecomposition of P (iterative factorizations...)

Smooth and strongly convex functons

Very hard to estimate μ and L in general

Can we select a good step-size as we go?

[Lecture 24, 25, Numerical Linear Algebra, Trefethen and Bau]

Line search

Exact line search

Choose the best step along the descent direction

$$t_k = \operatorname*{argmin}_{t \geq 0} f(x^k - t\nabla f(x^k))$$

Used when

- computational cost very low or
- there exist closed-form solutions

In general, impractical to perform exactly

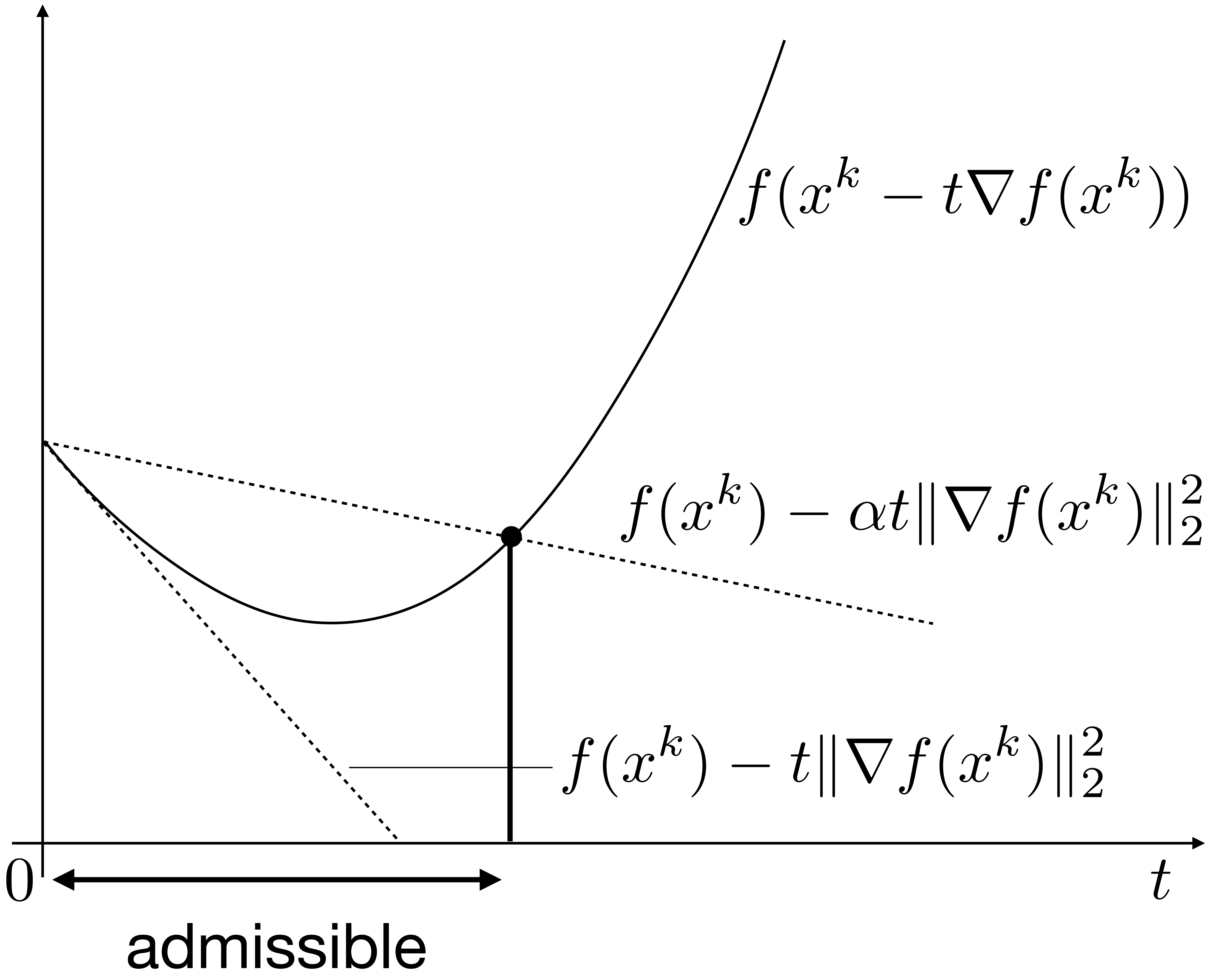
Backtracking line search

Condition

Armijo condition: for some $0 \leq \alpha \leq 1$

$$f(x^k - t\nabla f(x^k)) < f(x^k) - \alpha t \|\nabla f(x^k)\|_2^2$$

Guarantees
sufficient decrease
in objective value



Backtracking line search

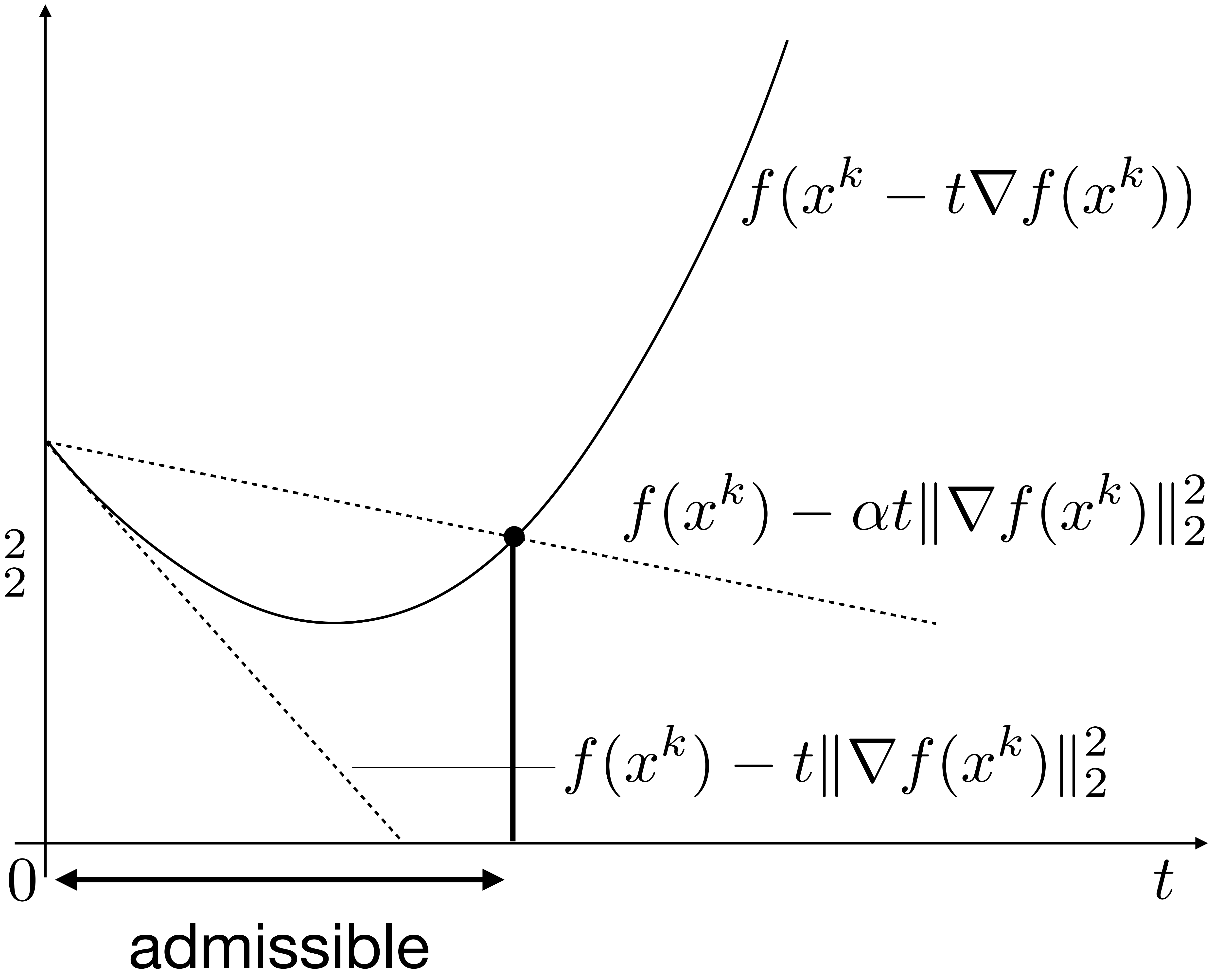
Iterations

initialization

$t = 1, \quad 0 < \alpha \leq 1/2, \quad 0 < \beta < 1$

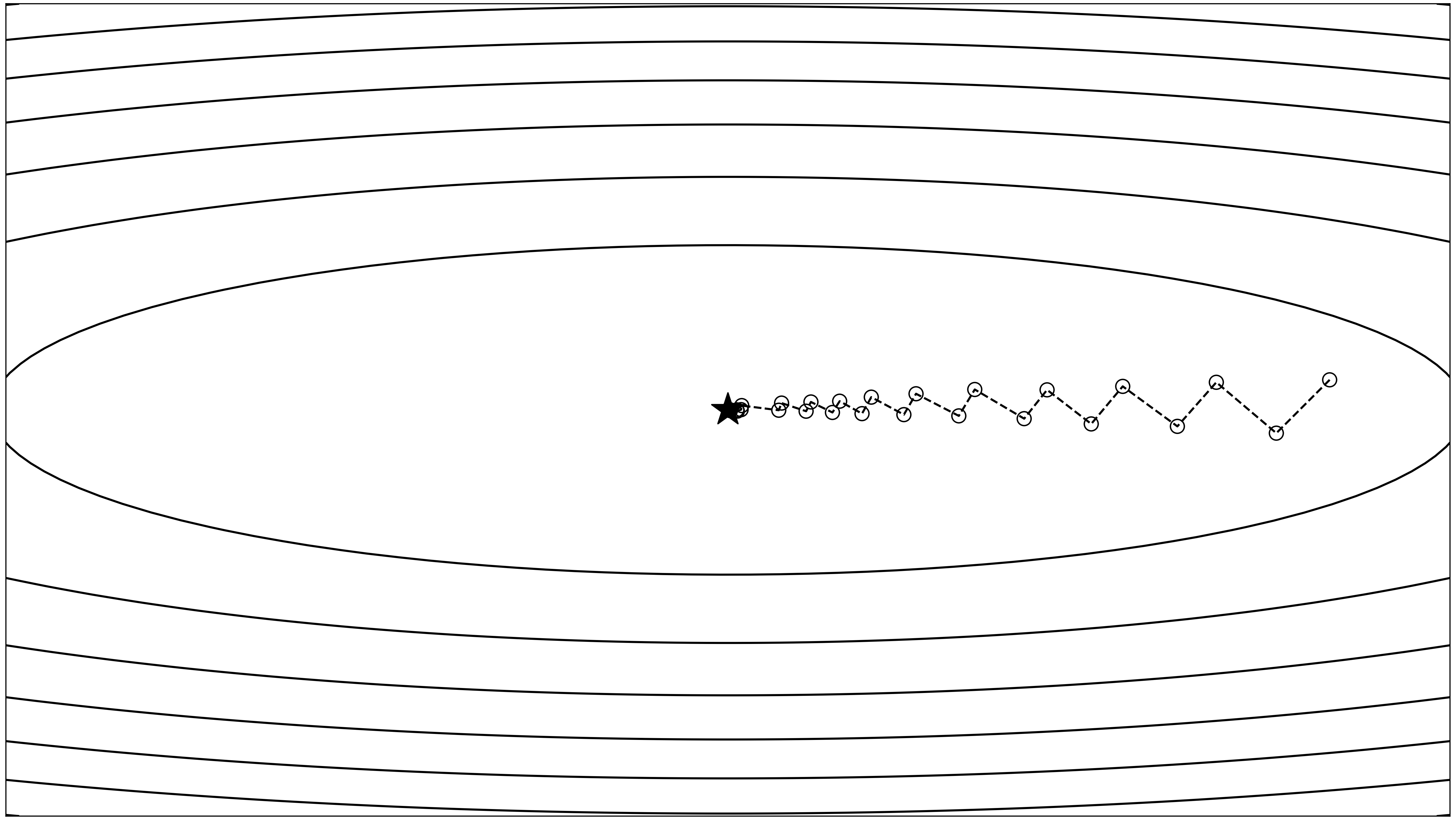
while $f(x^k - t\nabla f(x^k)) > f(x^k) - \alpha t \|\nabla f(x^k)\|_2^2$

$\quad t \leftarrow \beta t$



Backtracking line search

$$f(x) = (x_1^2 + 20x_2^2)/2 \qquad x^0 = (20, 1)$$



Backtracking line search

Converges in 31 iterations

Backtracking line search convergence

Theorem
Let f be L-smooth. If $t < 1/L$ then gradient descent with backtracking line search satisfies

$$f(x^k) - f(x^\star) \leq \frac{\|x^0 - x^\star\|_2^2}{2t_{\min}k}$$

where $t_{\min} = \min\{1, \beta/L\}$

Proof almost identical to fixed step case

Backtracking line search convergence

Theorem

Let f be L-smooth. If $t < 1/L$ then gradient descent with backtracking line search satisfies

$$f(x^k) - f(x^\star) \leq \frac{\|x^0 - x^\star\|_2^2}{2t_{\min}k}$$

where $t_{\min} = \min\{1, \beta/L\}$

Proof almost identical to fixed step case

Remarks

- If $\beta \approx 1$, similar to optimal step-size (β/L vs $1/L$)
- Still convergence rate $O(1/\epsilon)$ iterations (can be very slow!)

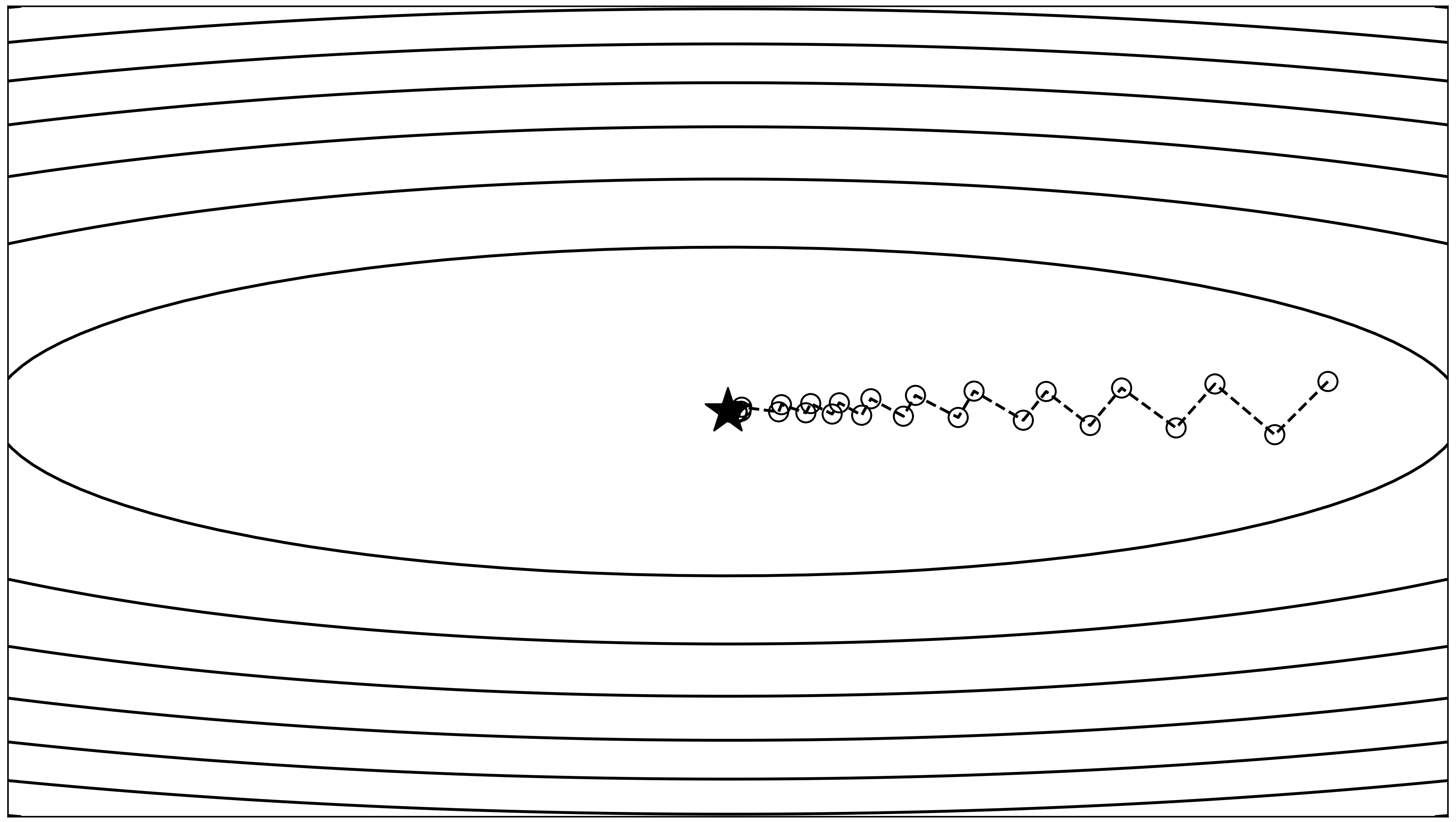
Gradient descent issues

Slow convergence

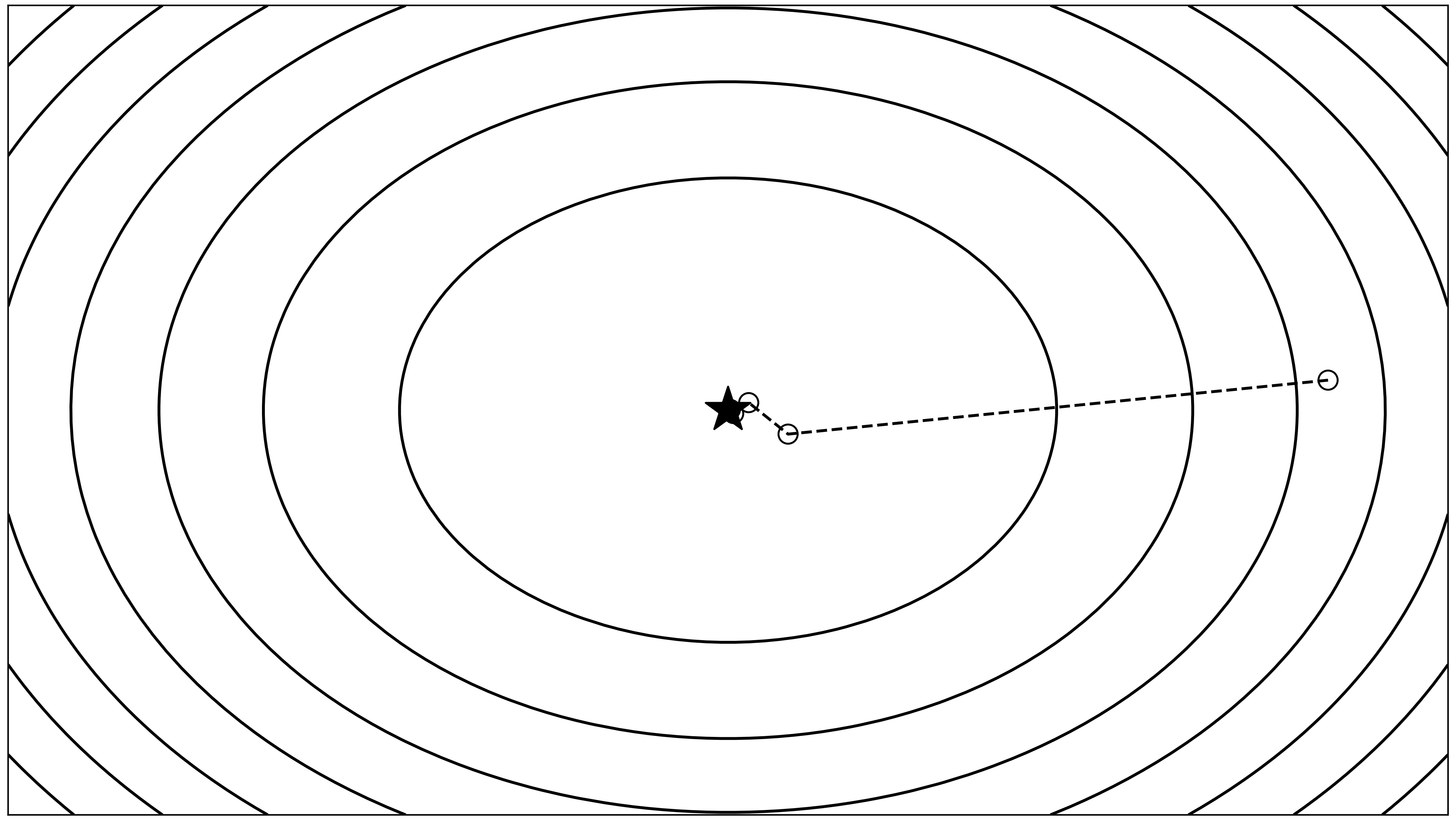
Very dependent on scaling

$$f(x) = (x_1^2 + 20x_2^2)/2$$

$$f(x) = (x_1^2 + 2x_2^2)/2$$



Slow convergence

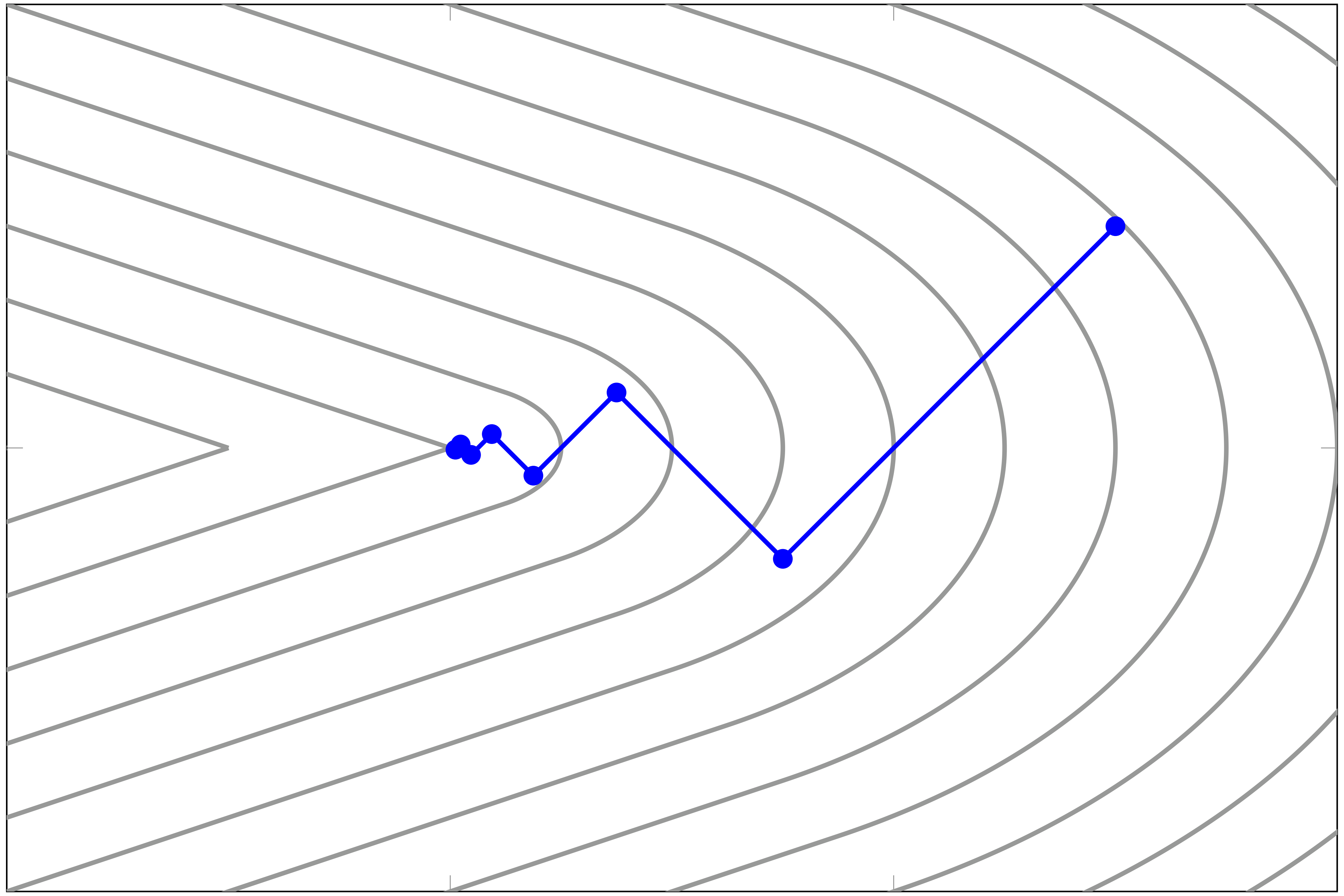


Faster

Non-differentiability

Wolfe's example

$$f(x) = \begin{cases} \sqrt{x_1^2 + \gamma x_2^2} & |x_2| \leq x_1 \\ \dfrac{x_1 + \gamma |x_2|}{\sqrt{1+\gamma}} & |x_2| > x_1 \end{cases}$$

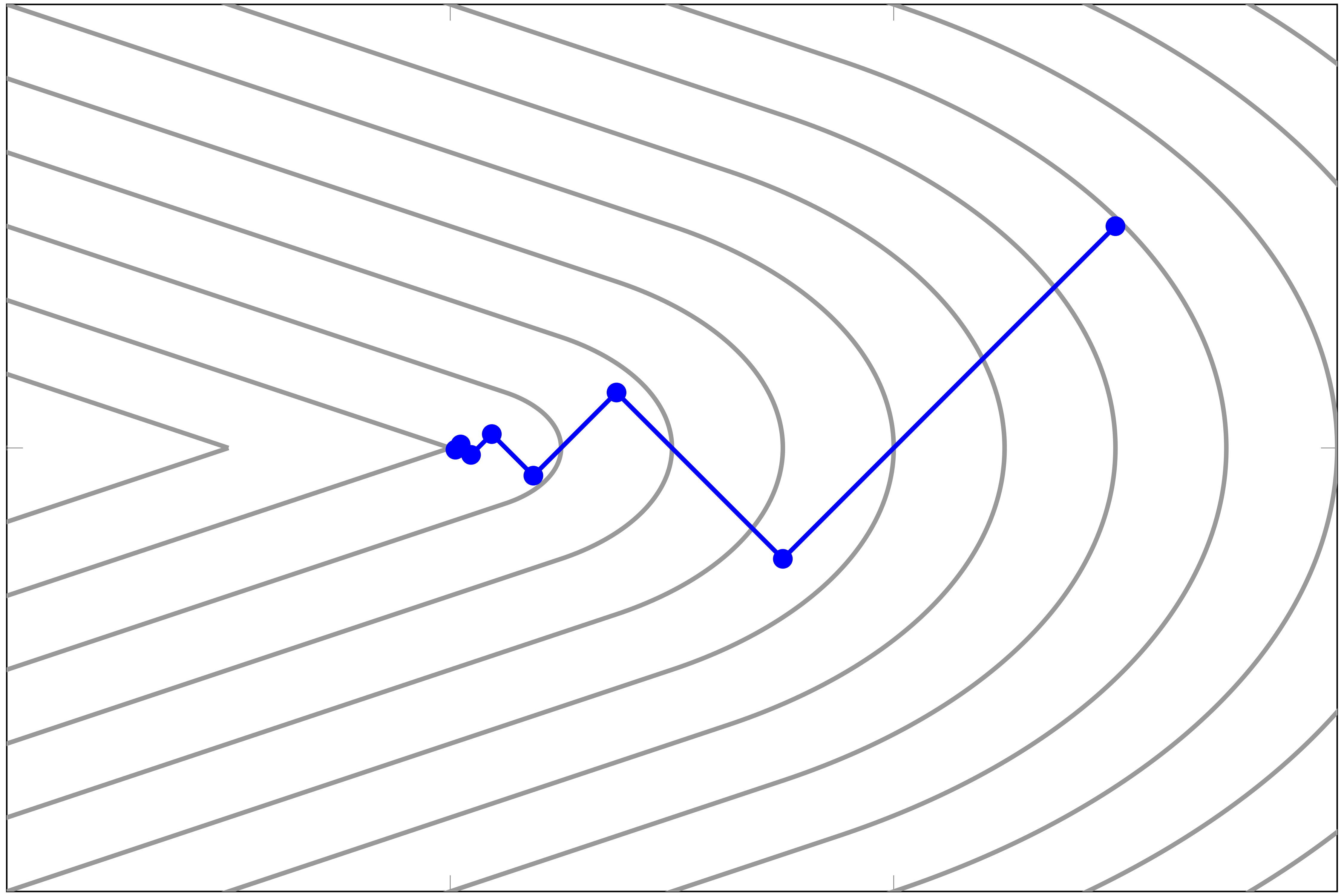


Gradient descent with *exact line search* gets stuck at $x = (0, 0)$

Non-differentiability

Wolfe's example

$$f(x) = \begin{cases} \sqrt{x_1^2 + \gamma x_2^2} & |x_2| \leq x_1 \\ \dfrac{x_1 + \gamma |x_2|}{\sqrt{1+\gamma}} & |x_2| > x_1 \end{cases}$$



Gradient descent with *exact line search* gets stuck at $x = (0, 0)$

In general: gradient descent cannot handle non-differentiable functions and constraints

Gradient descent

Today, we learned to:

- **Classify** optimization algorithms (zero, first, second-order)
- **Derive and recognize** convergence rates
- **Analyze** gradient descent complexity under smoothness and strong convexity (linear convergence, fast!)
- **Analyze** gradient descent complexity under only smoothness (sublinear convergence, slow!)
- **Apply** line search to get better step size
- **Understand** issues of Gradient descent

Next lecture

- Subgradient methods