

# **ORF522 – Linear and Nonlinear Optimization**

## **14. Gradient descent**

# 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

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

## Theorem

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

$$g_i(x) \leq 0, \quad (\text{for all affine } g_i)$$

$$g_i(x) < 0, \quad (\text{for all non-affine } g_i)$$

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

**Slater's condition**

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

## Remarks

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

# KKT for convex problems

## Always sufficient

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

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

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

( $L$  convex and differentiable  $\Rightarrow \nabla_x L(x^*, y^*, v^*) = 0$  implies  $x^*$  minimizes  $L$ )

Therefore,  $f(x^*) = q(y^*, v^*)$  and  $x^*, y^*, v^*$  are primal-dual optimal

## Necessary when constraint qualifications (Slater's) condition holds

Slater condition  $\Rightarrow$  strong duality  $p^* = d^*$  and dual optimum attained.

Therefore, since  $x^*, y^*, v^*$  are optimal, they must satisfy the KKT conditions.

# 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  $x^* = T(x^*)$  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  $e(x^*) = 0$

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

## Convergence rate

A sequence converges with order  $p$  and factor  $c$  if

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

# Convergence rates types

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

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

**Examples**

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

**Sublinear convergence** (slower than linear)

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

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

**Superlinear convergence** (faster than linear)

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

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

**Quadratic convergence** ( $c$  can be  $> 1$ )

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

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

# Convergence rates

## Number of iterations

**Solve inequality for  $k$**

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

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

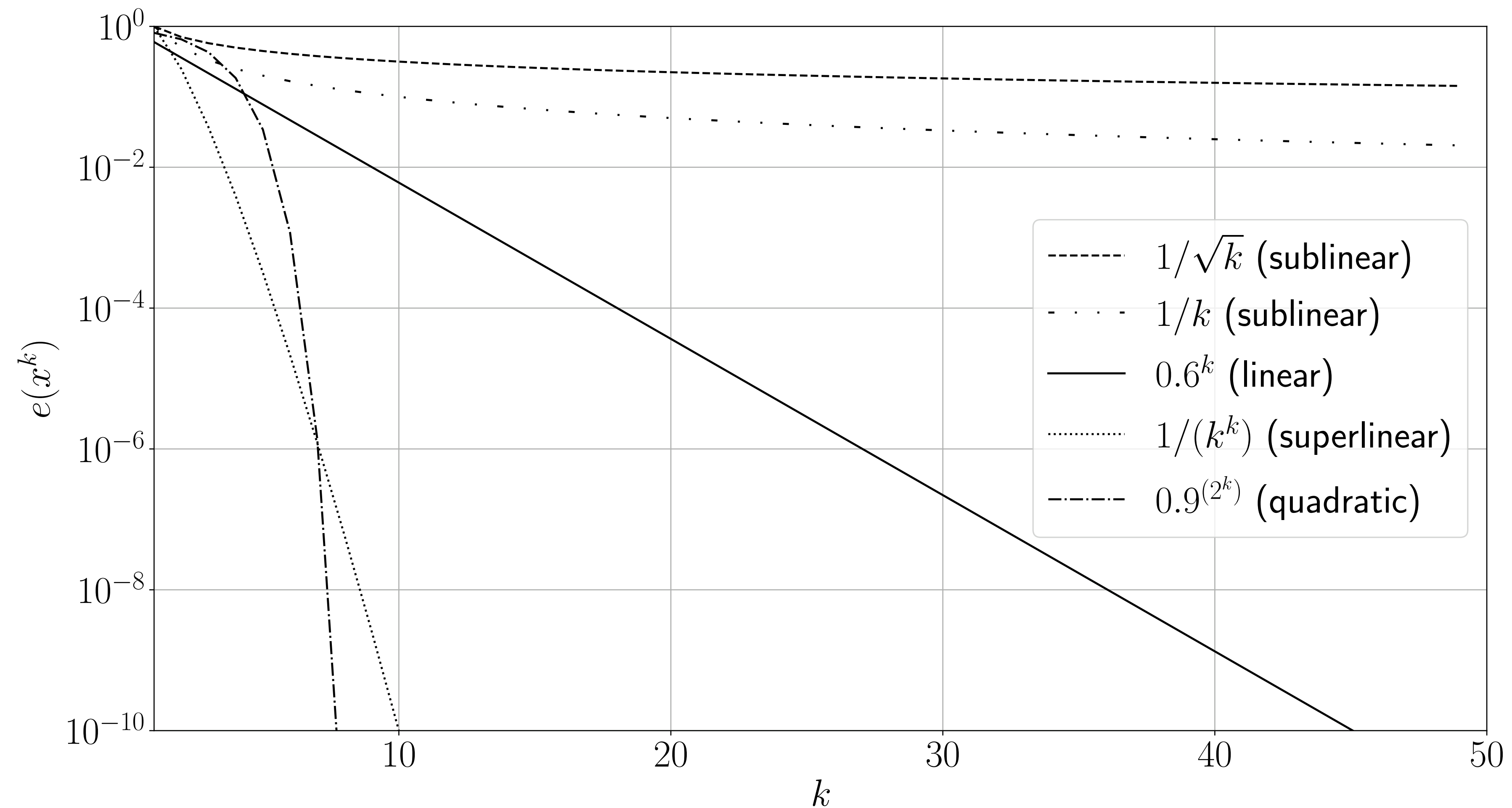
$$e(x^{k+1}) \leq \epsilon \implies c^k e(x^0) \leq \epsilon \implies k \geq O(\log(1/\epsilon))$$

**Example:** sublinear convergence

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

# 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.

**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

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

$f$  is differentiable

# General descent scheme

## Iterations

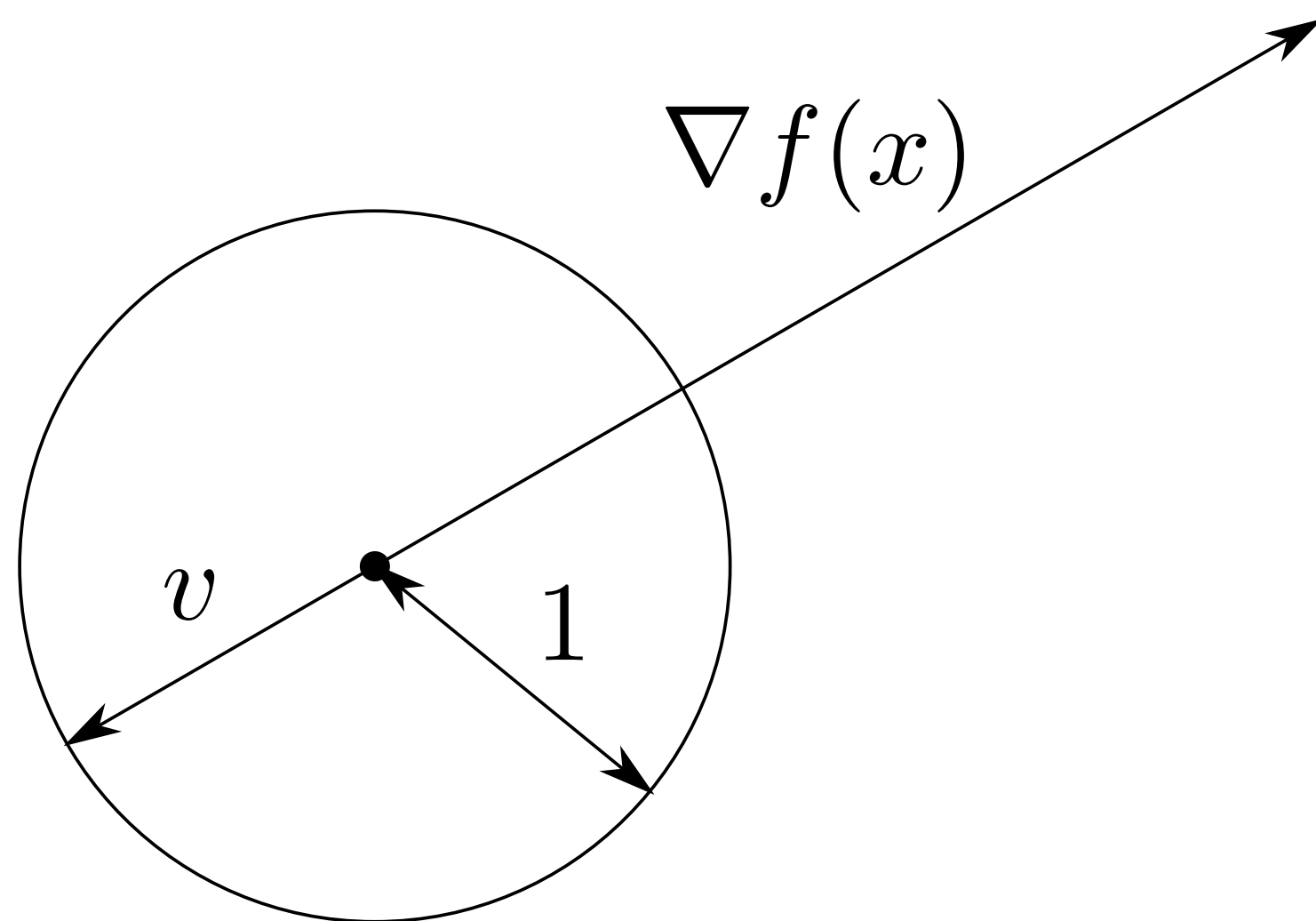
- Pick descent direction  $d^k$ , i.e.,  $\nabla f(x^k)^T d^k < 0$
- Pick step size  $t_k$
- $x^{k+1} = x^k + t^k d^k, \quad k = 0, 1, \dots$



# Gradient descent

[Cauchy 1847]

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



**Interpretation:** steepest descent (Cauchy-Schwarz)

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

## Iterations

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

(very cheap iterations)

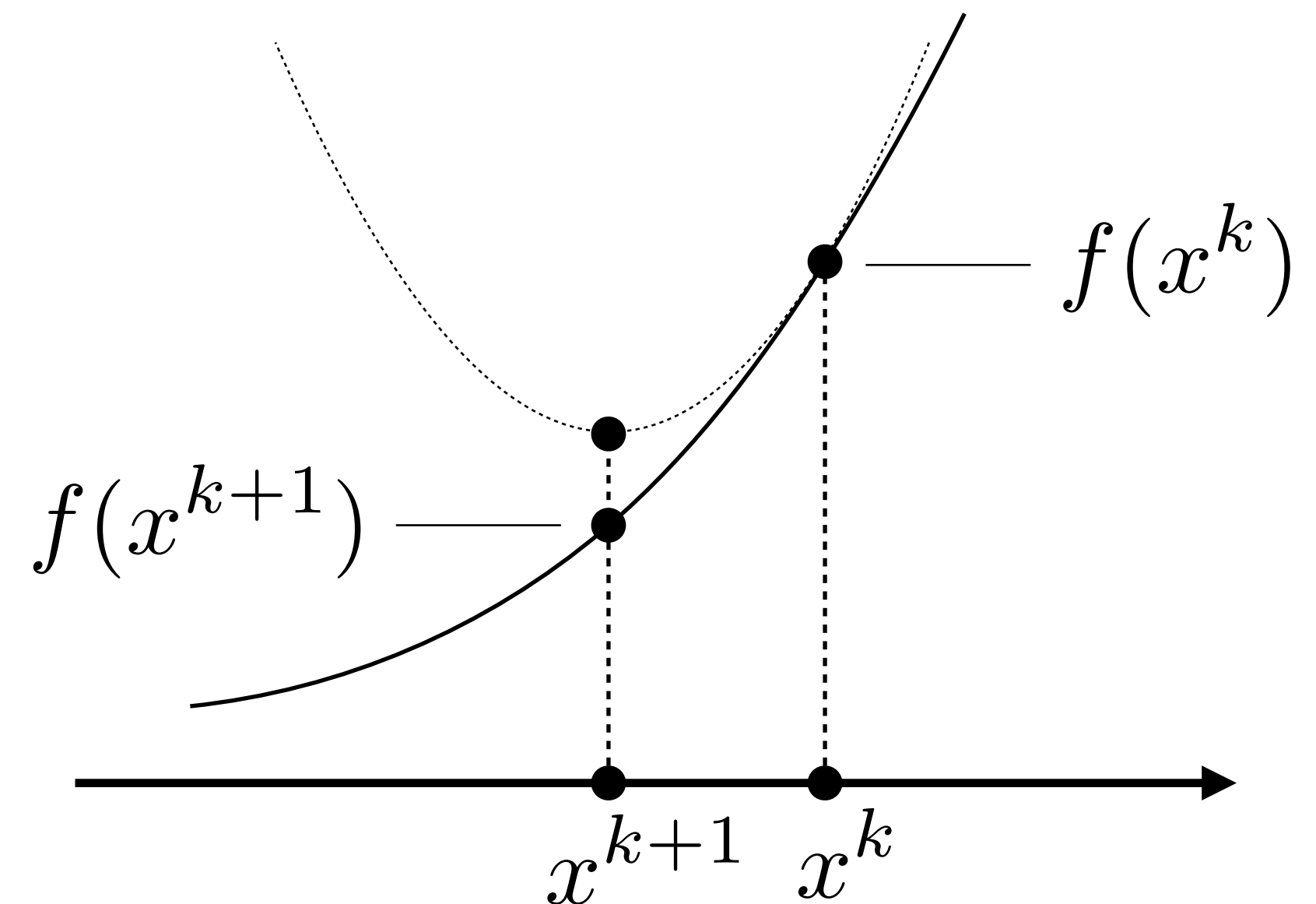
# Quadratic function interpretation

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

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

Set gradient with respect to  $y$  to 0...

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



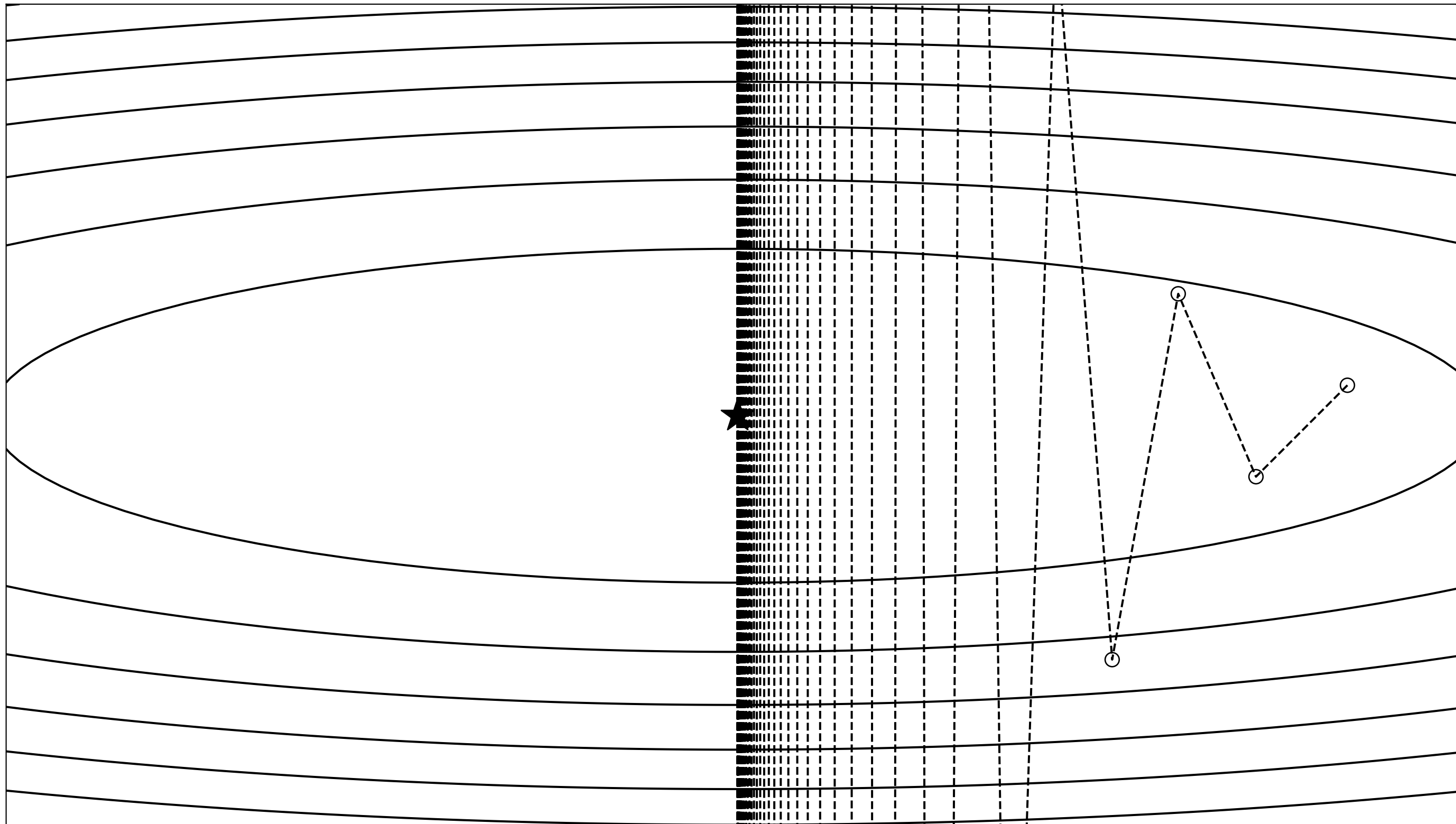
**Fixed step size**

# Fixed step size

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

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

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



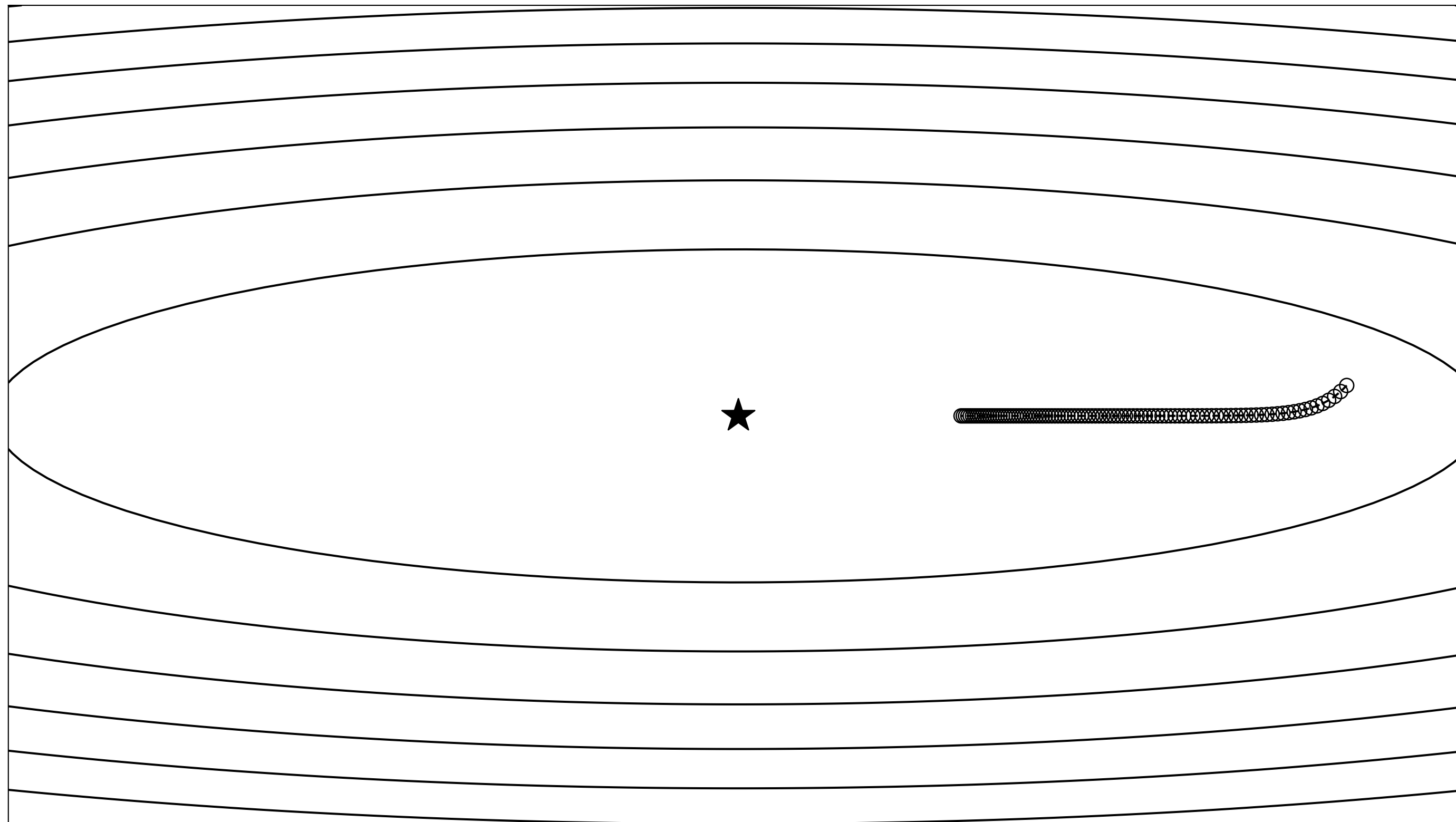
**It diverges**

# Fixed step size

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

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

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



**too slow**

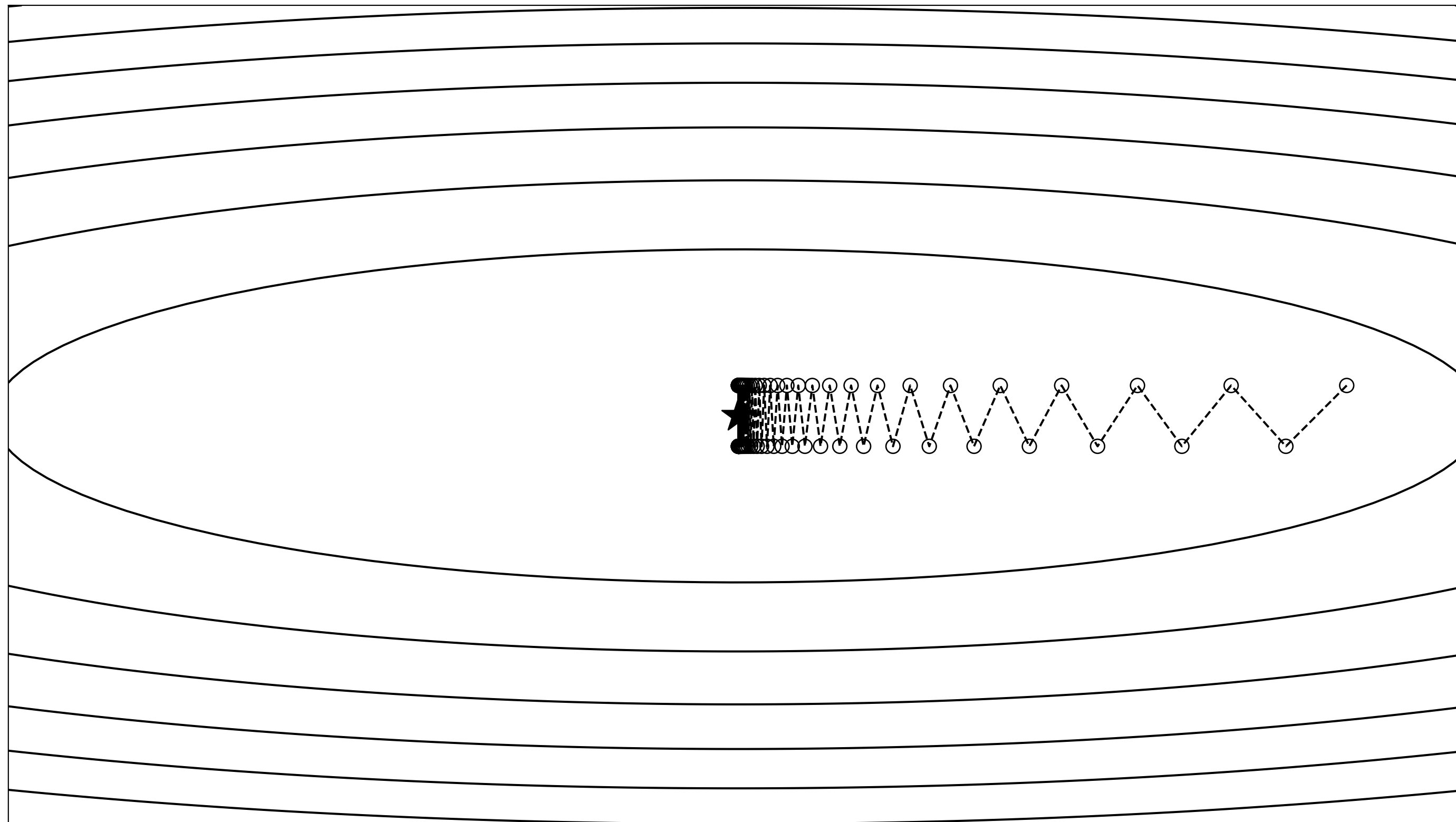
# Fixed step size

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

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

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

$$t = 0.10$$



**it oscillates**

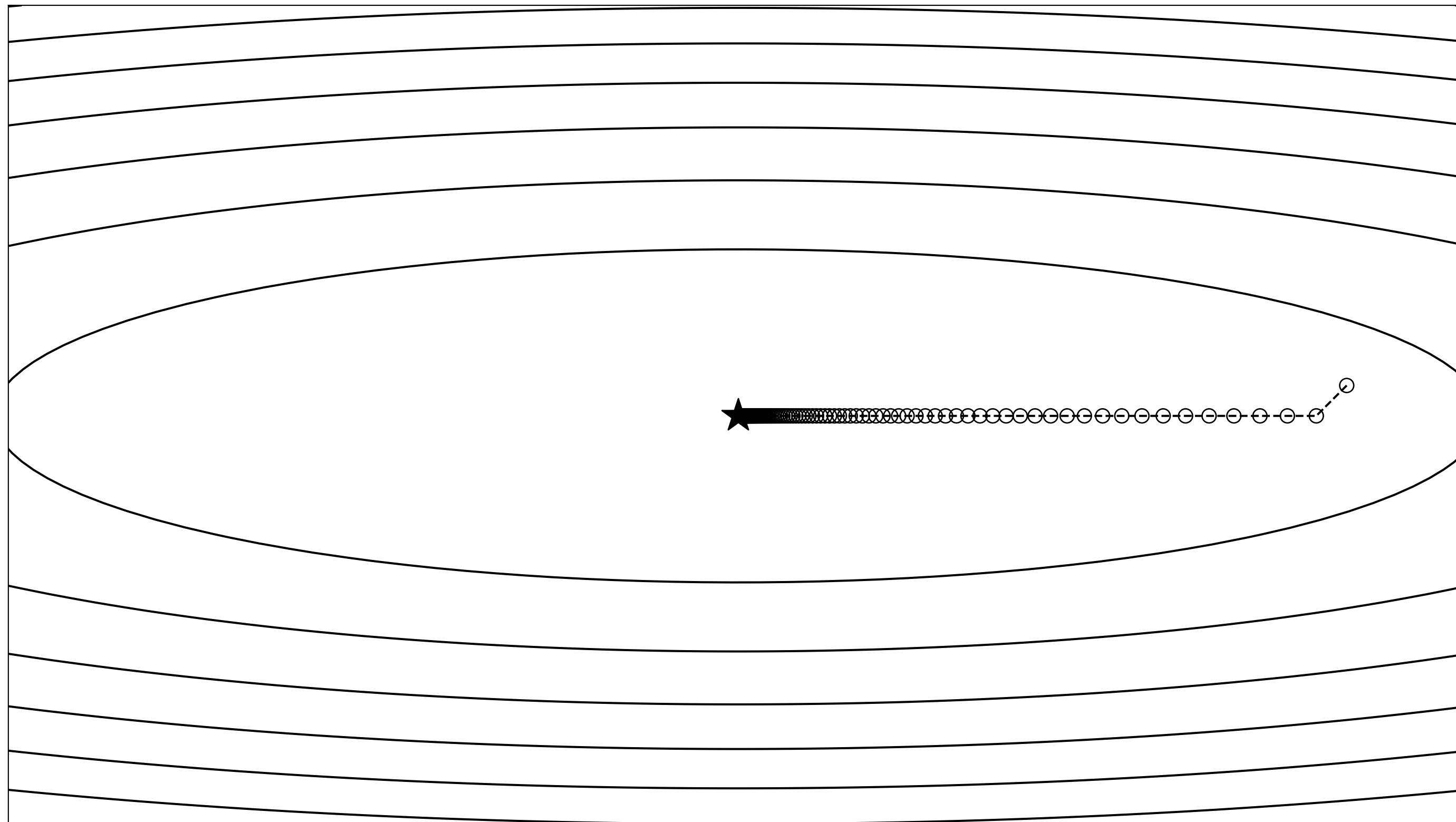
# Fixed step size

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

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

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

$$t = 0.05$$



**just right!**

It converges in 149 iterations

**How do we find the best one?**

# Quadratic optimization



# Quadratic optimization

$$\text{minimize } f(x) = \frac{1}{2}(x - x^*)^T P(x - x^*) \quad \text{where } P \succ 0$$
$$\nabla f(x) = P(x - x^*)$$

**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  $x^*$ ,  $\nabla f(x^*) = 0$  and  $\nabla^2 f(x^*)$  dominates other terms of the Taylor expansion.

# Quadratic optimization convergence

## Theorem

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

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

## Remarks

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

# Quadratic optimization convergence

## Proof

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

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

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

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

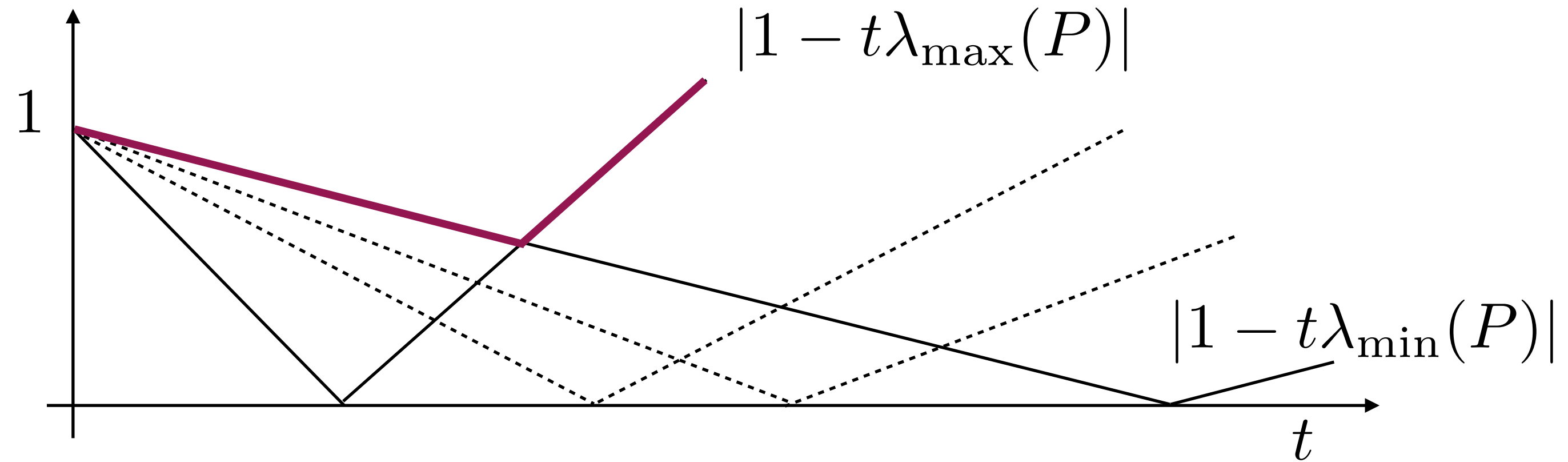
# Quadratic optimization convergence

## Proof (continued)

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

$$= \max\{|1 - t\lambda_{\max}(P)|, |1 - t\lambda_{\min}(P)|\}$$

$$= \max\{1 - t\lambda_{\min}(P), -1 + t\lambda_{\max}(P)\}$$



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

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

# Quadratic optimization convergence

## Proof (continued)

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

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

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

Apply the inequality recursively to get the result



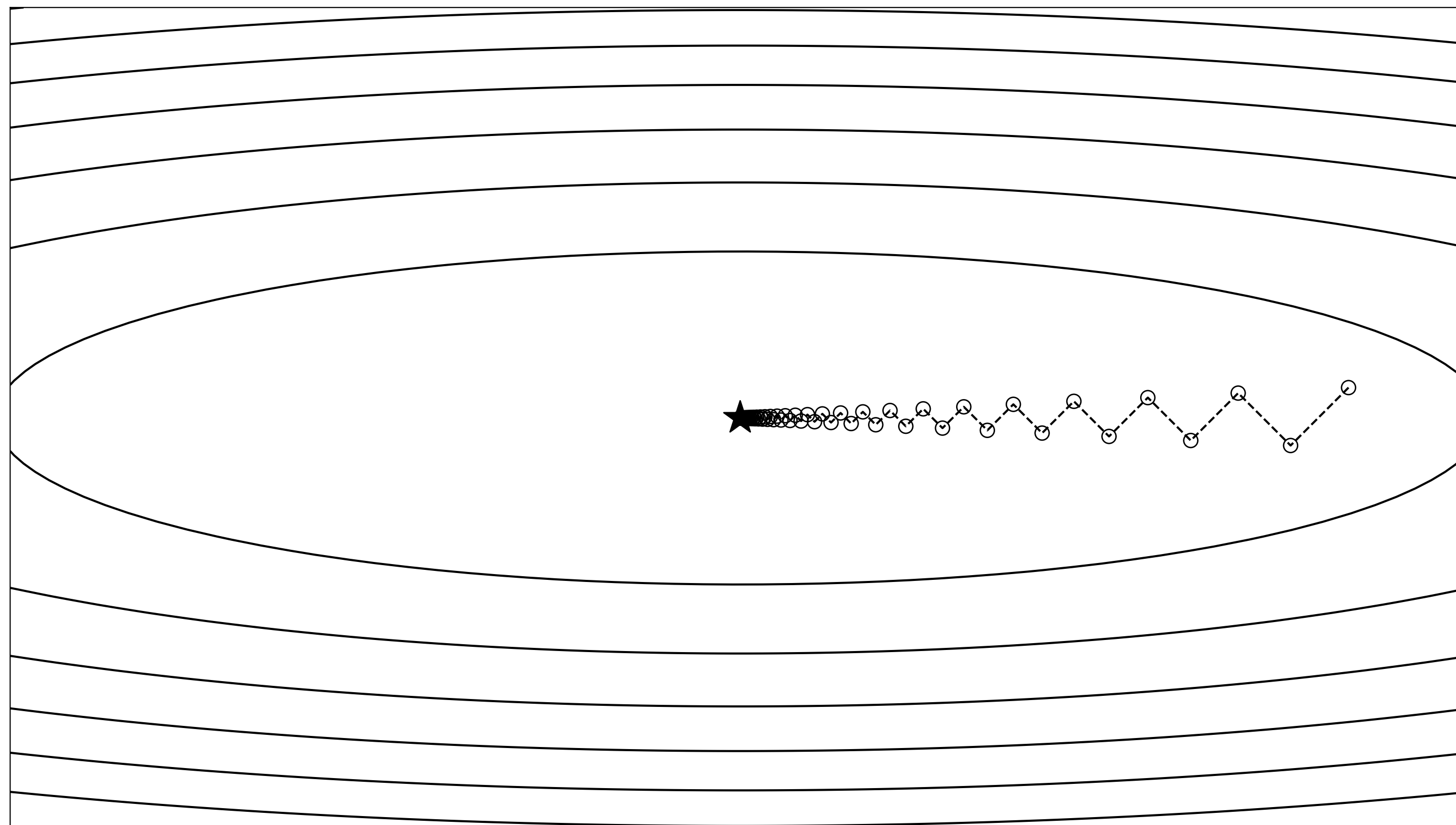
# Optimal fixed step size

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

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

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

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



**Optimal step size**

It converges in 80 iterations

# When does it converge?

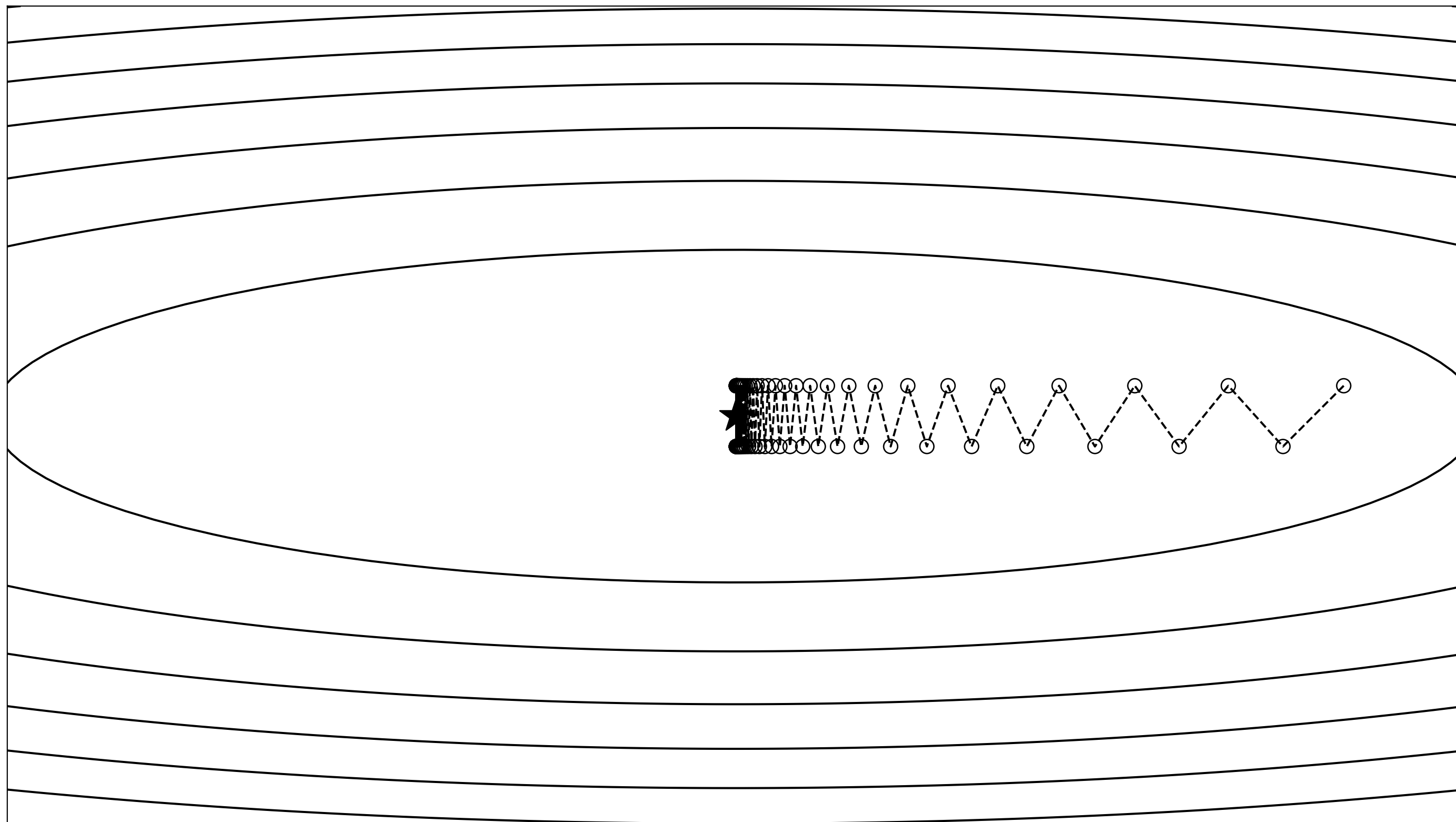
## Iterations

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

## 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

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

## First-order characterization

$$\|\nabla f(x) - \nabla f(y)\|_2 \leq L\|x - y\|_2, \quad \forall x, y \quad (\text{Lipschitz continuous gradient})$$

## 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  $\text{dom } f$  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  $\mu$ -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

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

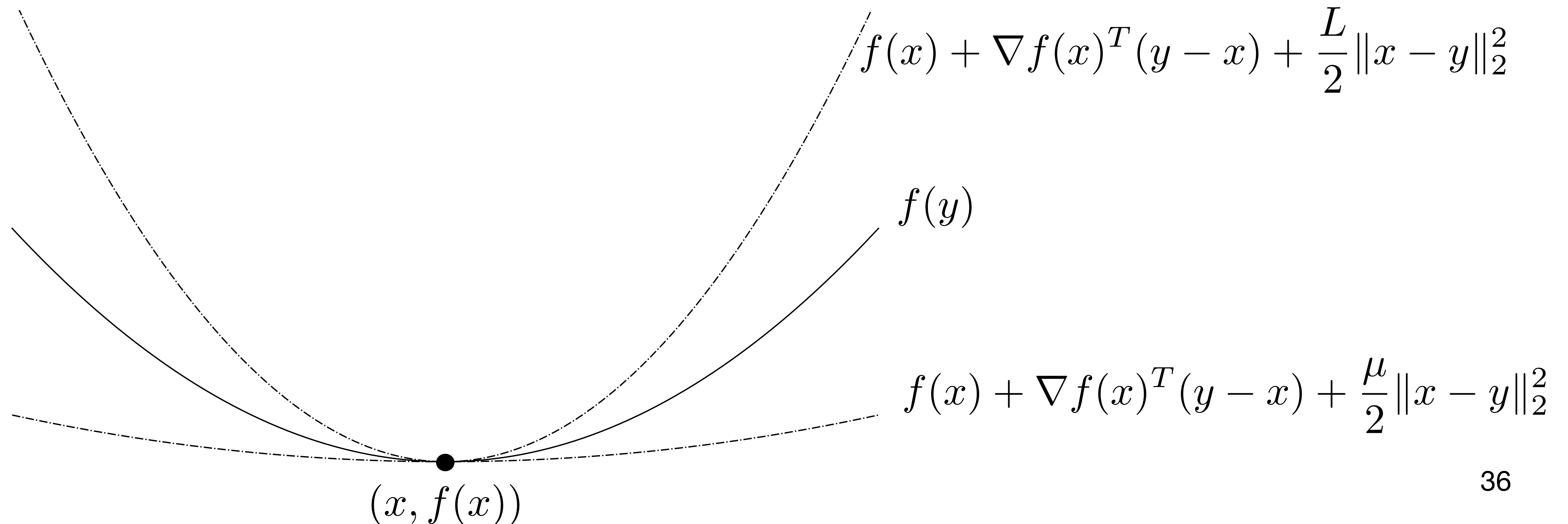
## Second-order characterization

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

# Strongly convex and smooth functions

$f$  is  $\mu$ -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  $\mu$ -strongly convex and  $L$ -smooth. If  $t = \frac{2}{\mu + L}$ , then

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

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

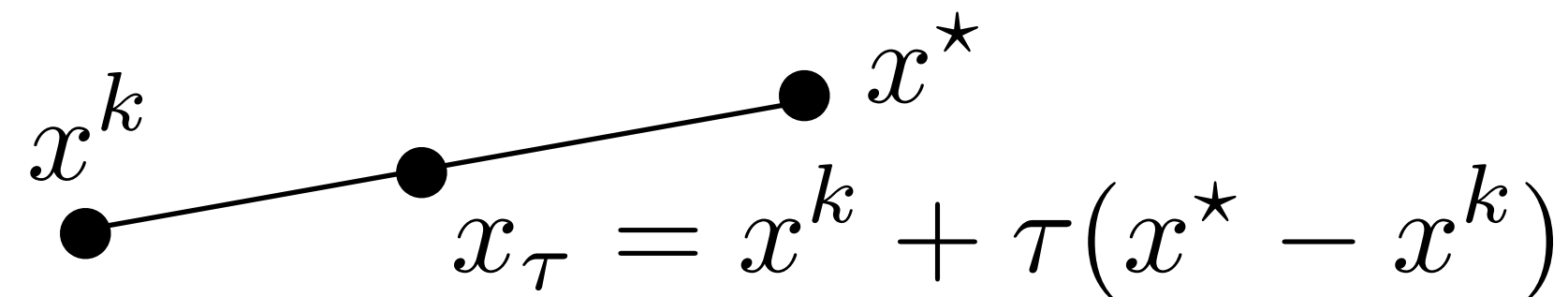
## Remarks

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

# Strongly convex and smooth convergence

**Proof**

Fundamental  
theorem of calculus:



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

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

$$= \int_0^1 \nabla^2 f(x_\tau) d\tau (x^k - x^*)$$

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

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

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

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

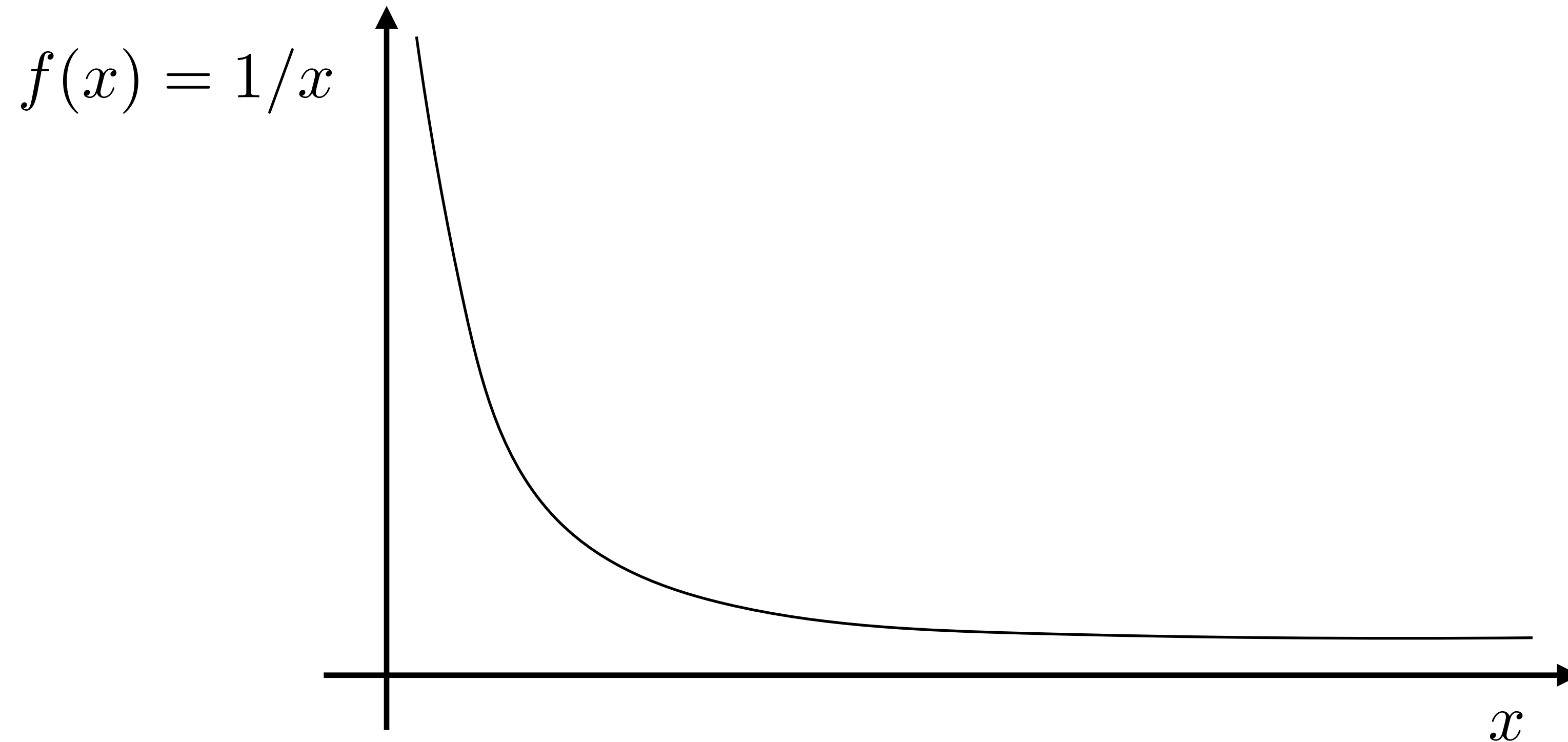
*(similar to quadratic)*

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 ( $x^* = \infty$ ) but the objective value very close

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

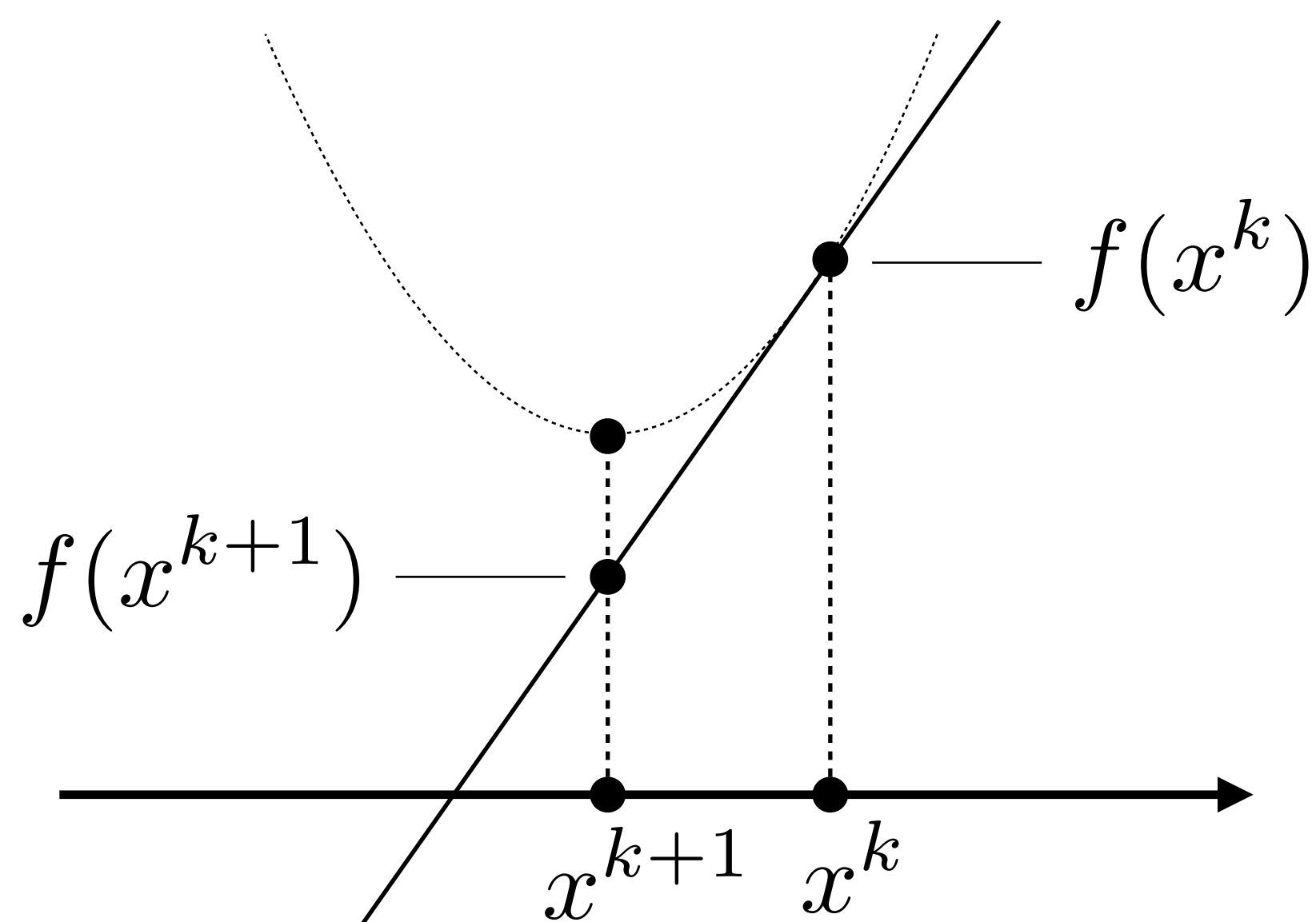


# Null growth directions without strong convexity

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  $\frac{1}{t_k} I$

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



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

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

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

# Convergence for smooth functions

## Proof

Use  $L$ -Lipschitz constant

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

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

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

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

# Convergence for smooth functions

## Proof (continued)

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

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

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

# Convergence for smooth functions

## Proof (continued)

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

$$\begin{aligned}\sum_{i=1}^k (f(x^i) - f(x^*)) &\leq \frac{1}{2t} \sum_{i=1}^k (\|x^{i-1} - x^*\|_2^2 - \|x^i - x^*\|_2^2) \\ &= \frac{1}{2t} (\|x^0 - x^*\|_2^2 - \|x^k - x^*\|_2^2) \\ &\leq \frac{1}{2t} \|x^0 - x^*\|_2^2\end{aligned}$$

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

$$f(x^k) - f(x^*) \leq \frac{1}{k} \sum_{i=1}^k (f(x^i) - f(x^*)) \leq \frac{1}{2kt} \|x^0 - x^*\|_2^2$$



# 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 functions

Very hard to estimate  $\mu$  and  $L$  in general

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

# 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  $0 < \alpha \leq 1/2$

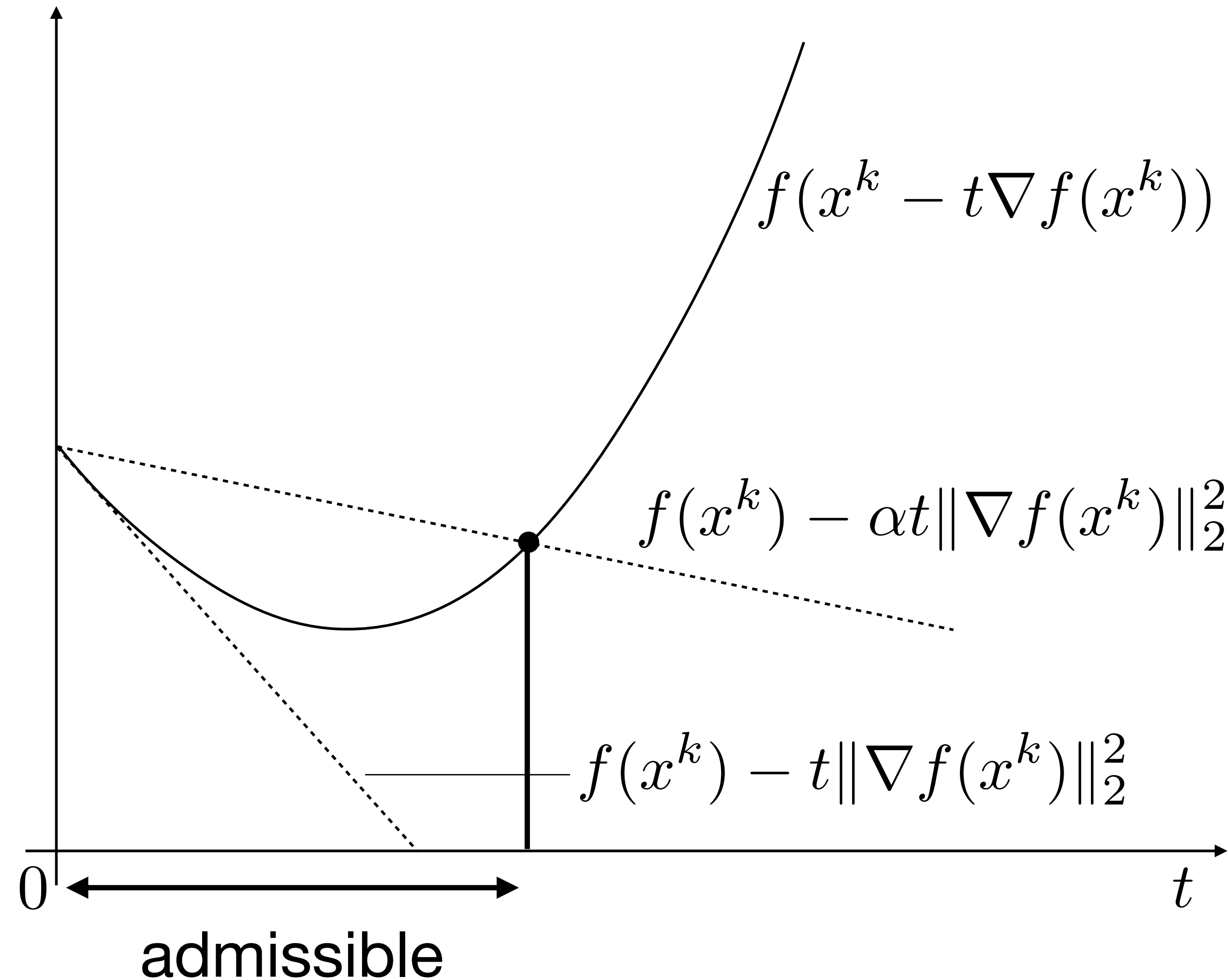
$$f(x^k + td^k) < f(x^k) + \alpha t \nabla f(x^k)^T d^k$$



where  $d^k = -\nabla f(x^k)$

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

Guarantees  
**sufficient decrease**  
in objective value



# Backtracking line search

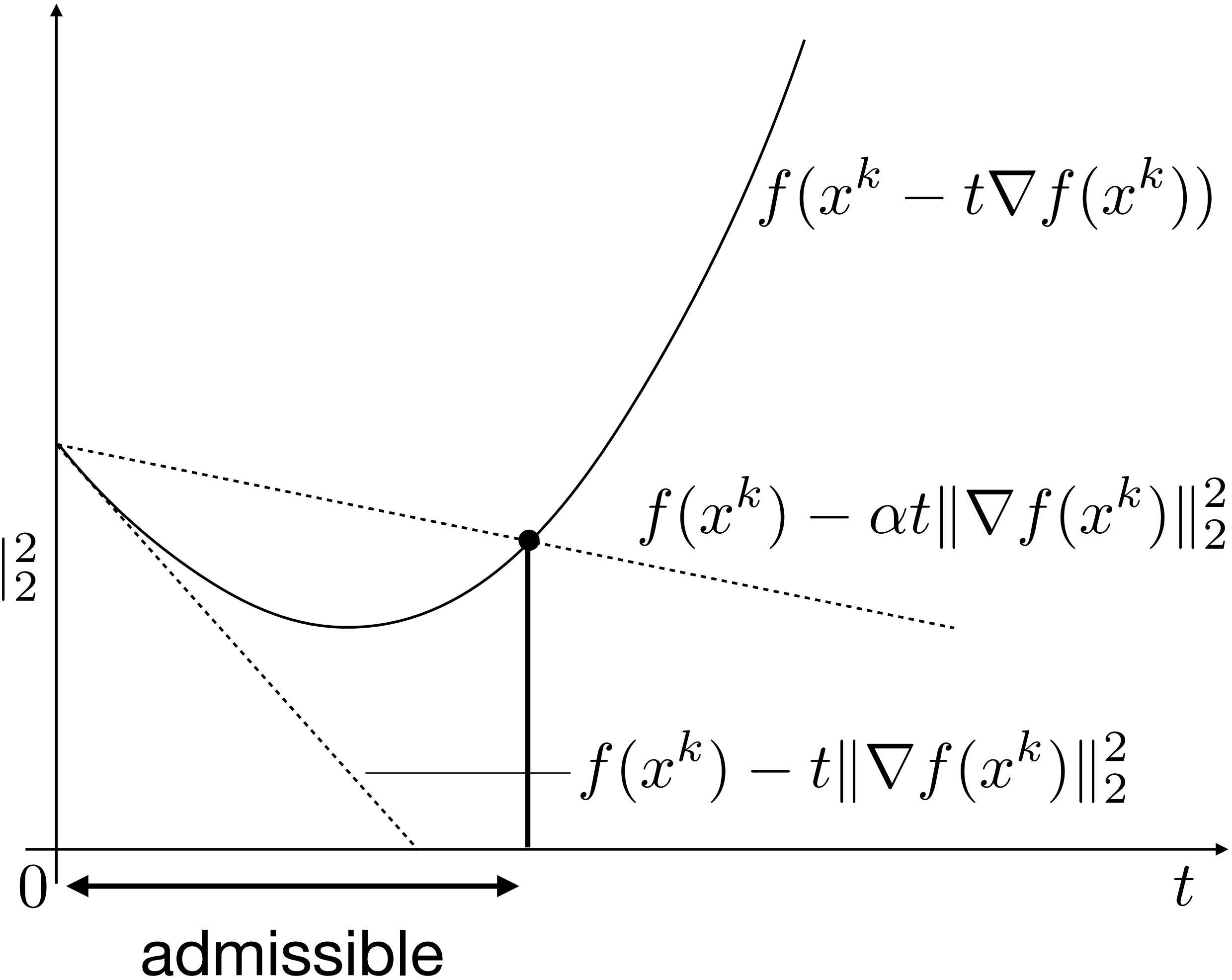
Iterations

**initialization**

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

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

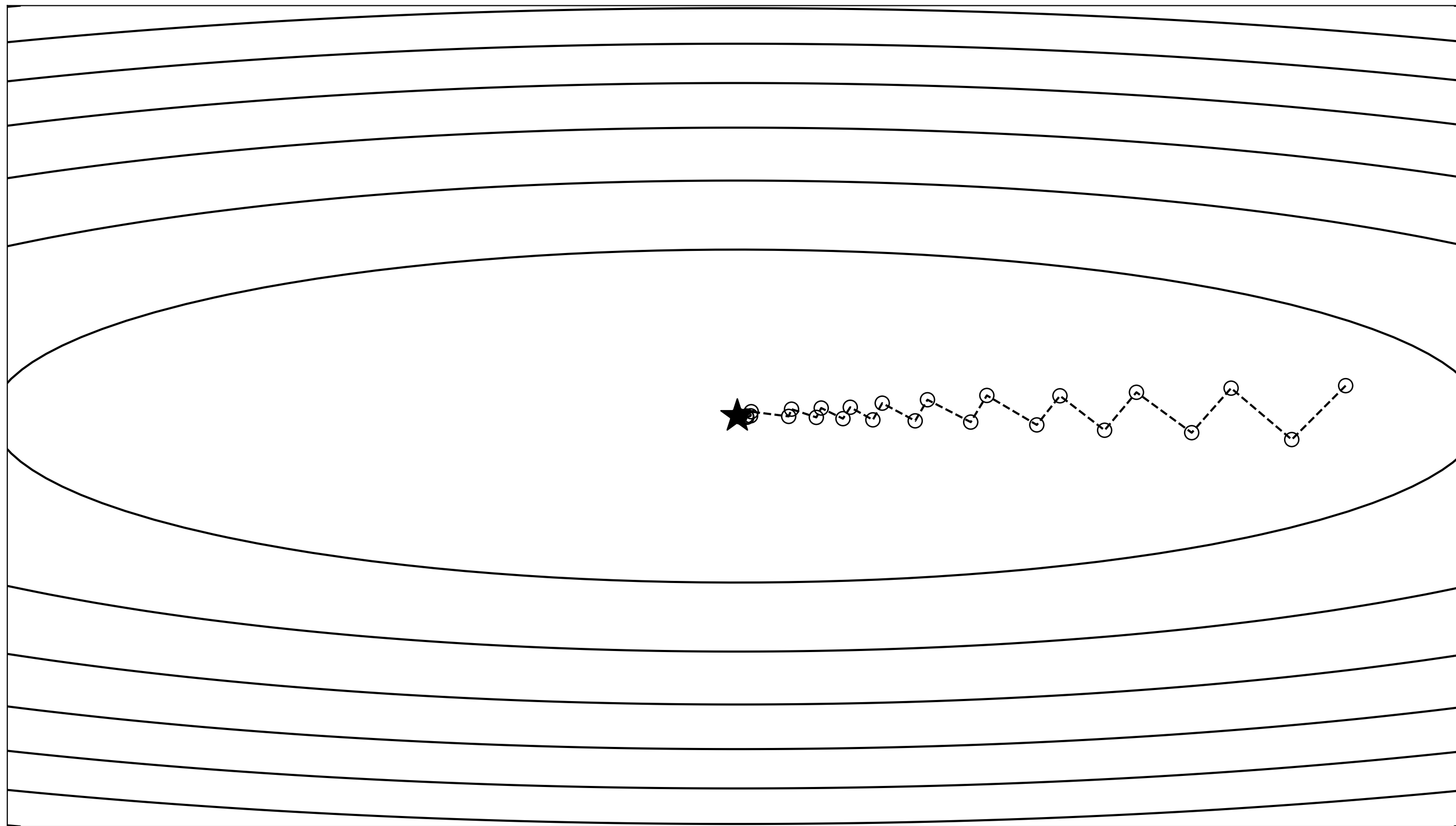
$$t \leftarrow \beta t$$



# Backtracking line search

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

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



**Backtracking line search**

Converges in 31 iterations

# Backtracking line search convergence

## Theorem

Let  $f$  be  $L$ -smooth. Gradient descent with backtracking line search satisfies

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

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

**Proof** almost identical to fixed step case

## Remarks

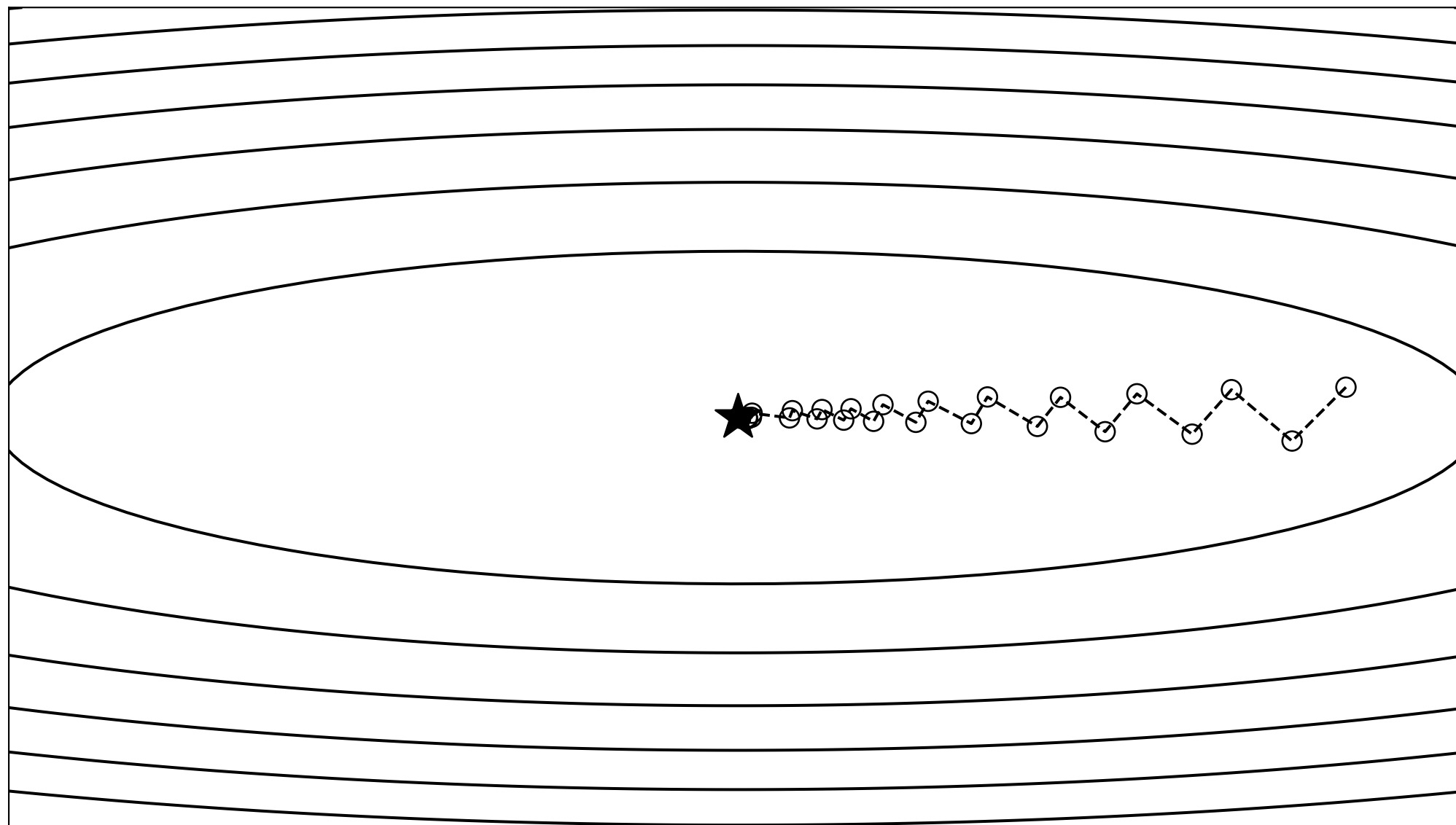
- If  $\beta \approx 1$ , similar to optimal step-size ( $\beta/L$  vs  $1/L$ )
- Still convergence rate  $O(1/\epsilon)$  iterations (can be very slow!)

# Gradient descent issues

# Slow convergence

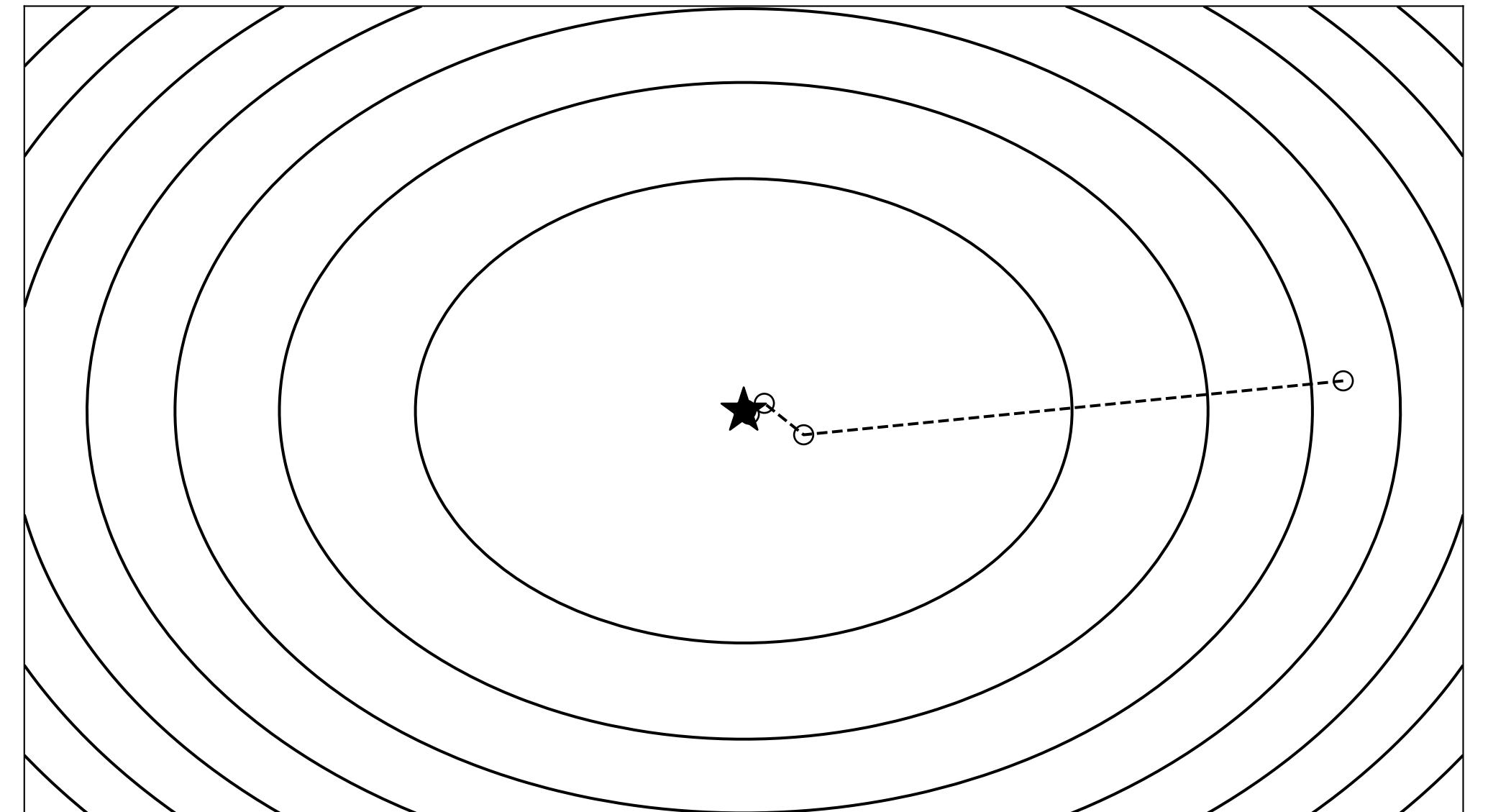
Very dependent on scaling

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



**Slow convergence**

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

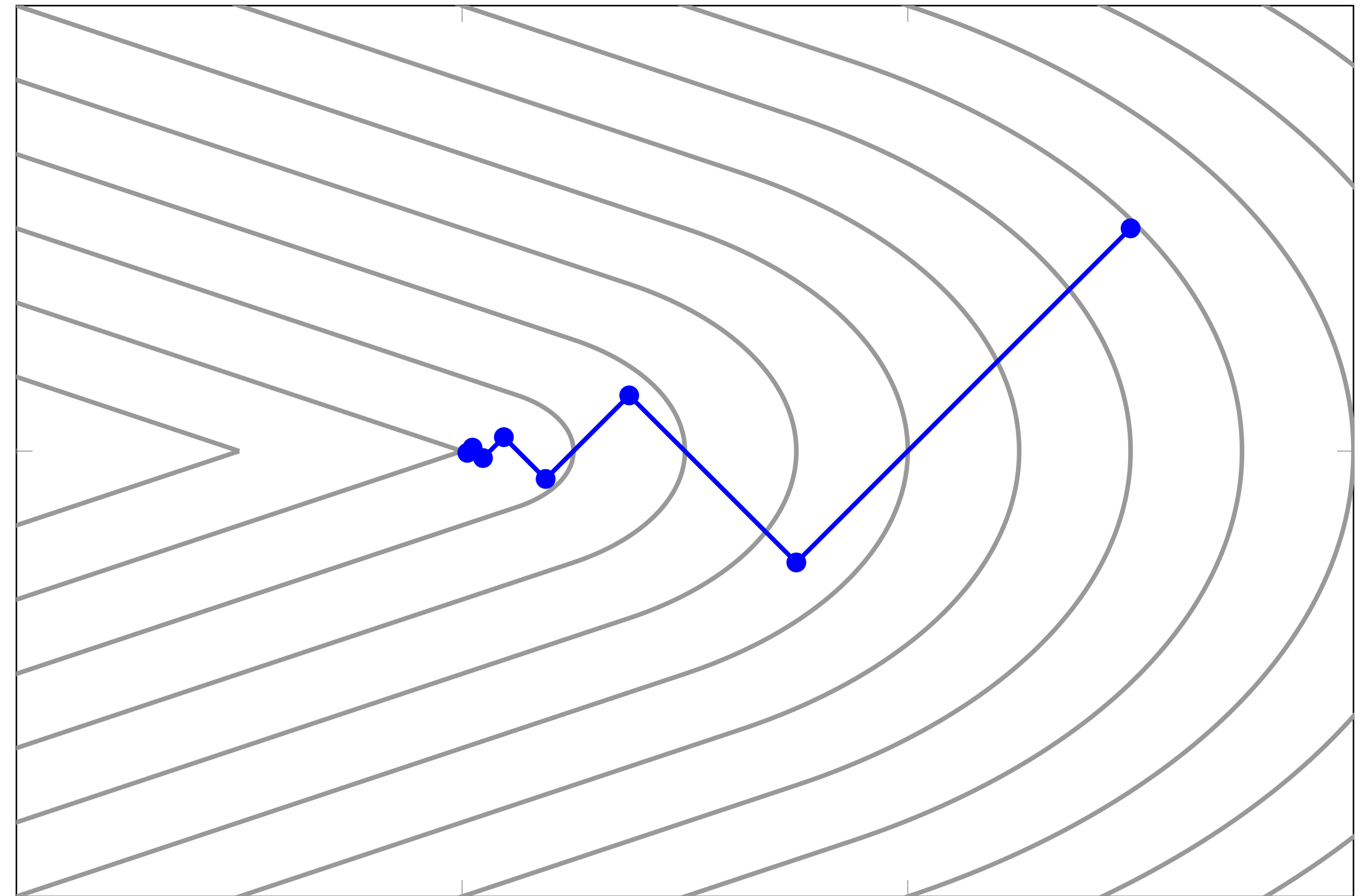


**Faster**

# Non-differentiability

## Wolfe's example

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



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