ORF522 – Linear and Nonlinear Optimization

16. Proximal methods

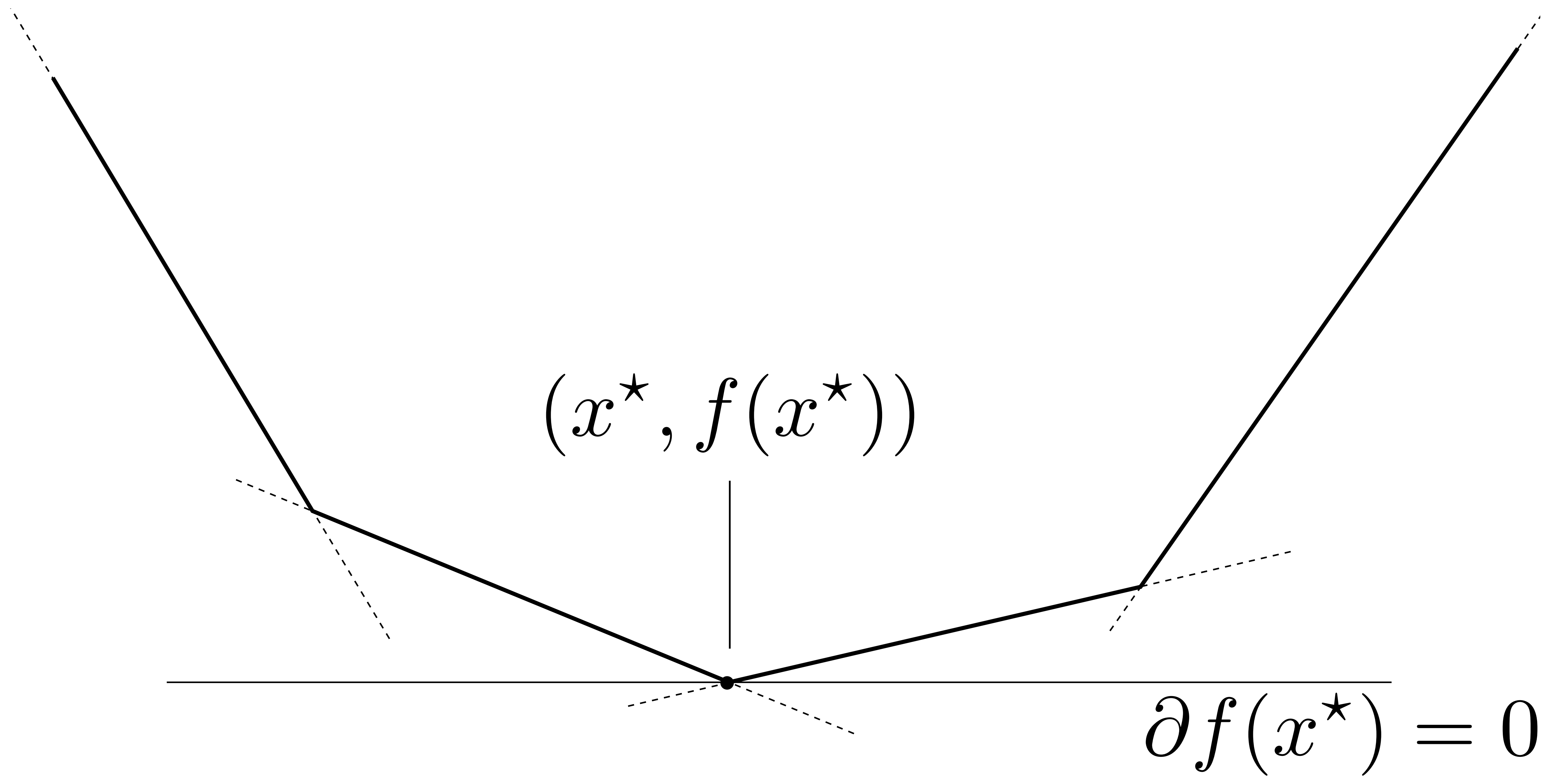
Bartolomeo Stellato — Fall 2022

Recap

Fermat's optimality condition

For any (not necessarily convex) function f where $\partial f(x^\star) \neq \emptyset$,
$x^\star$ is a global minimizer if and only if

$$0 \in \partial f(x^\star)$$



Proof

A subgradient $g = 0$ means that, for all y

$$f(y) \geq f(x^\star) + 0^T(y - x^\star) = f(x^\star) \quad \blacksquare$$

Note differentiable case with $\partial f(x) = \{\nabla f(x)\}$

Today’s lecture

[Chapter 3 and 6, FMO] [PA]

Proximal methods

- Optimality conditions with subdifferentials
- Subgradient method
- Proximal operators
- Proximal gradient method

Optimality conditions with subdifferentials

Constrained optimization

Indicator function
of a convex set

$$\mathcal{I}_C(x) = \begin{cases} 0 & x \in C \\ \infty & x \notin C \end{cases}$$

Constrained form

minimize $\quad f(x)$

subject to $\quad x \in C$

$\longleftrightarrow$

Unconstrained form

minimize $\quad f(x) + \mathcal{I}_C(x)$

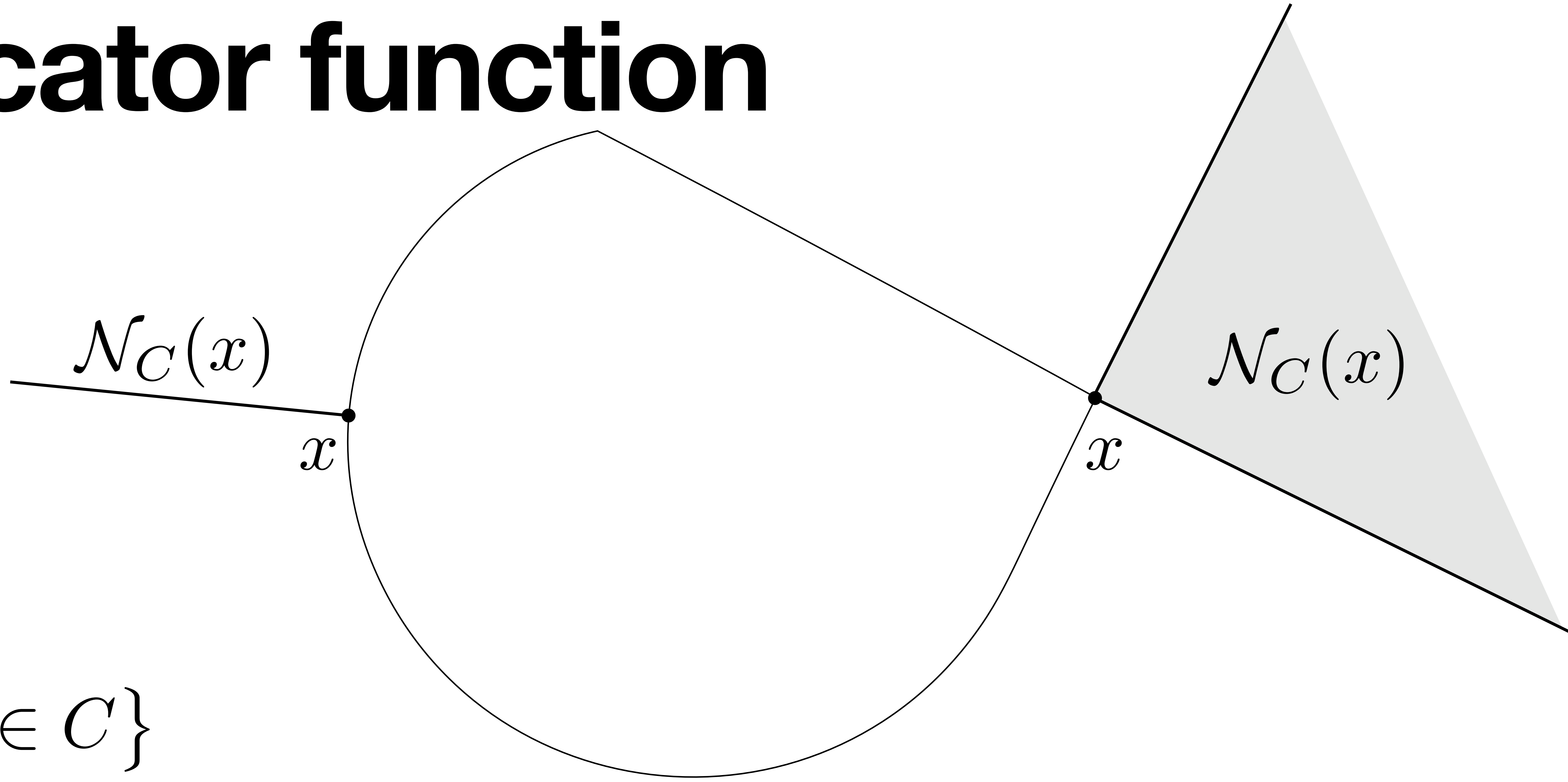
Subgradient of indicator function

The subdifferential of the **indicator functon** is the **normal cone**

$$\partial \mathcal{I}_C(x) = \mathcal{N}_C(x)$$

where,

$$\mathcal{N}_C(x) = \{g \mid g^T(y-x) \le 0, \quad \text{for all } y \in C\}$$



Proof

By definition of subgradient g, $\mathcal{I}_C(y) \ge \mathcal{I}_C(x) + g^T(y-x), \quad \forall y$

$$y \notin C \quad \Longrightarrow \quad \mathcal{I}_C(y) = \infty$$

$$y \in C \quad \Longrightarrow \quad 0 \ge g^T(y-x)$$

$\blacksquare$

First-order optimality conditions from subdifferentials

minimize $\quad f(x) + \mathcal{I}_C(x)$

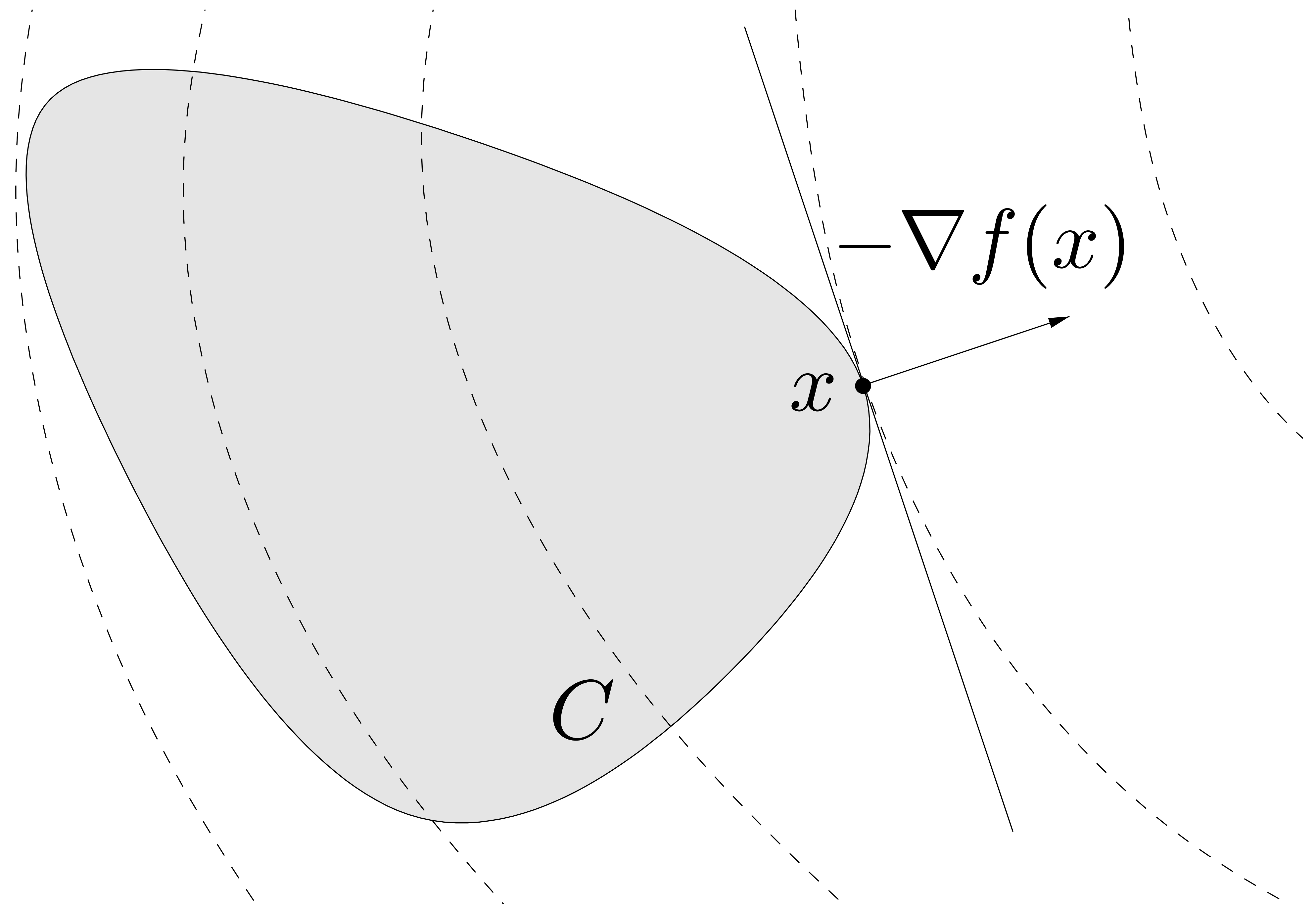
f convex smooth,
C convex

Fermat's optimality condition

$$0 \in \partial(f(x) + \mathcal{I}_C(x))$$

$$\iff 0 \in \{\nabla f(x)\} + \mathcal{N}_C(x)$$

$$\iff -\nabla f(x) \in \mathcal{N}_C(x)$$



Equivalent to

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

Example: KKT of a quadratic program

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & Ax \leq b \end{array} \longrightarrow \text{minimize} \quad (1/2)x^T P x + q^T x + \mathcal{I}_{\{Ax \leq b\}}(x)$$

Gradient

$$\nabla f(x) = Px + q$$

Normal cone to polyhedron

Idea: [Lecture 13].
Proof: [Theorem 6.46, Variational Analysis, Rockafellar & Wets]

$$\mathcal{N}_{\{Ax \leq b\}}(x) = \{A^T y \mid y \geq 0 \quad \text{and} \quad y_i(a_i^T x - b_i) = 0\}$$

First-order optimality condition

$$-\nabla f(x) \in \partial \mathcal{I}_{\{Ax \leq b\}}(x) = \mathcal{N}_{\{Ax \leq b\}}(x) \quad \longleftrightarrow$$

KKT Optimality conditions

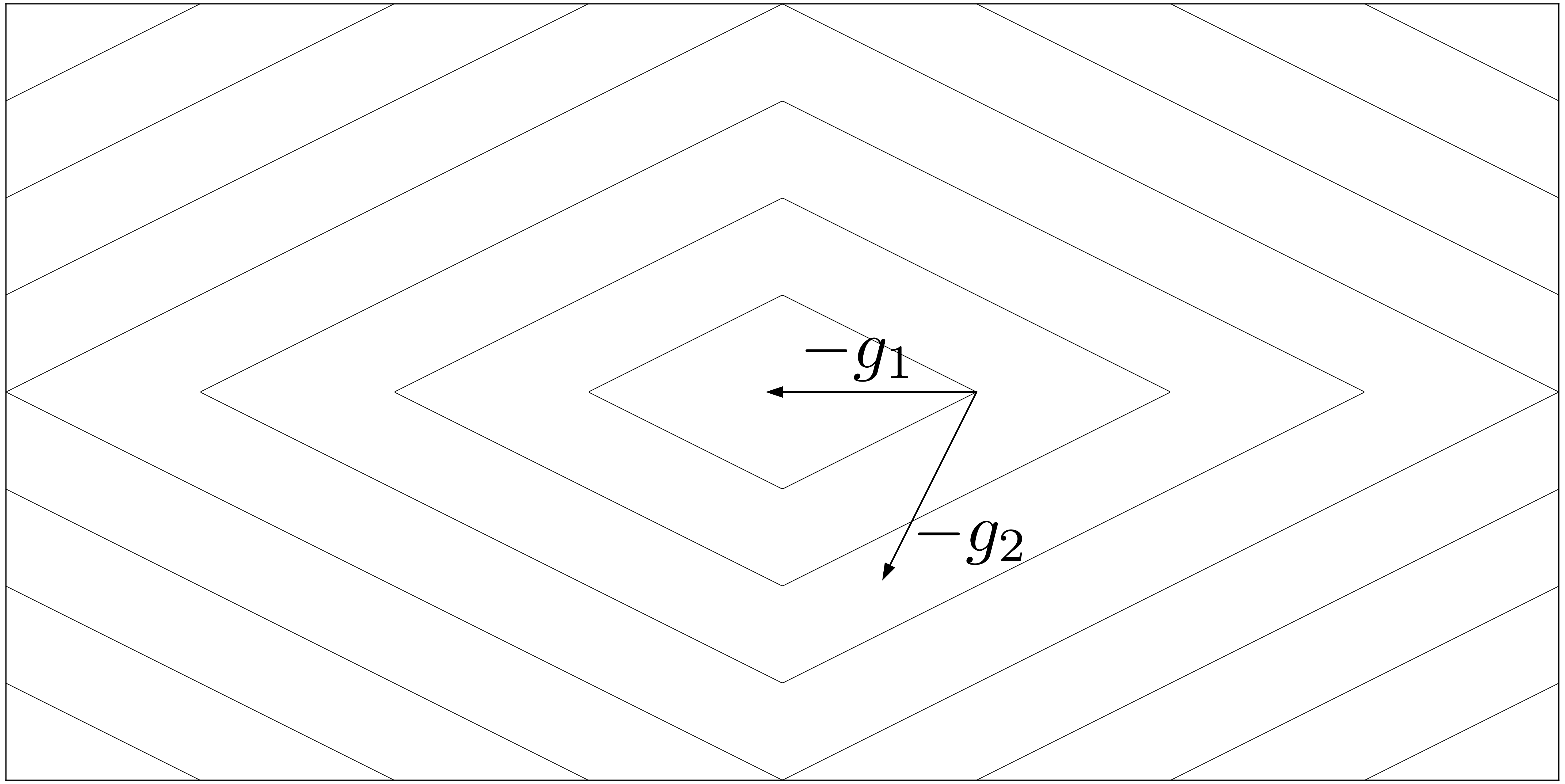
$$\begin{aligned} Px + q + A^T y &= 0 \\ y &\geq 0 \\ Ax - b &\leq 0 \\ y_i(a_i^T x - b_i) &= 0, \quad i = 1, \ldots, m \end{aligned}$$

Subgradient method

Negative subgradients are not necessarily descent directions

$$f(x) = |x_1| + 2|x_2|$$

$$x = (1, 0)$$



$g_1 = (1, 0) \in \partial f(x)$ and
$-g_1$ is a descent direction

$g_2 = (1, 2) \in \partial f(x)$ and
$-g_2$ is not a descent direction

Subgradient method

Convex optimization problem

minimize $\quad f(x) \qquad$ (optimal cost $f^\star$)

Iterations

$$x^{k+1} = x^k - t_k g^k, \qquad g^k \in \partial f(x^k)$$

g^k is **any subgradient** of f at x^k

Not a descent method, keep track of the best point

$$f_{\text{best}}^k = \min_{i=1,\ldots,k} f(x^i)$$

Step sizes

Line search can lead to **suboptimal points**

Step sizes ***pre-specified***, not adaptively computed
(different than gradient descent)

Fixed: $t_k = t$ for $k = 0, \dots$

Diminishing: $\sum_{k=0}^{\infty} t_k^2 < \infty, \quad \sum_{k=0}^{\infty} t_k = \infty$

Square summable but not summable
(goes to 0 but not too fast)

e.g., $t_k = O(1/k)$

Convergence

Assumptions

- f is convex with $\mathbf{dom}\, f = \mathbf{R}^n$
- $f(x^\star) > -\infty$ (finite optimal value)
- f is Lipschitz continuous with constant $G > 0$, i.e.

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

which is equivalent to $\|g\|_2 \leq G, \quad \forall g \in \partial f(x),\ \forall x$

Convergence

Lipschitz continuity equivalence

f is Lipschitz continuous with constant $G > 0$, i.e.

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

which is equivalent to $\|g\|_2 \leq G, \quad \forall g \in \partial f(x),\ \forall x$

Proof

If $\|g\| \leq G$ for all subgradients, pick $x, g_x \in \partial f(x)$ and $y, g_y \in \partial f(y)$. Then,

$$\begin{aligned} & g_x^T(x - y) \geq f(x) - f(y) \geq g_y^T(x - y) \\ \implies \quad & G\|x - y\|_2 \geq f(x) - f(y) \geq -G\|x - y\|_2 \end{aligned}$$

If $\|g\|_2 > G$ for some $g \in \partial f(x)$. Take $y = x + g/\|g\|_2$ such that $\|x - y\|_2 = 1$:

$$f(y) \geq f(x) + g^T(y - x) = f(x) + \|g\|_2 > f(x) + G \qquad \blacksquare$$

Convergence

Theorem

Given a convex, G-Lipschitz continuous f with finite optimal value, the subgradient method obeys

$$f_{\text{best}}^k - f^\star \leq \frac{R^2 + G^2 \sum_{i=0}^k t_i^2}{2 \sum_{i=0}^k t_i}$$

where $\|x^0 - x^\star\|_2 \leq R$

Convergence

Proof

Key quantity: euclidean distance to optimal set
(not function value since it can go up and down)

$$
\begin{aligned}
\|x^{k+1} - x^\star\|_2^2 &= \|x^k - t_k g^k - x^\star\|_2^2 \\
&= \|x^k - x^\star\|_2^2 - 2t_k (g^k)^T (x^k - x^\star) + t_k^2 \|g^k\|_2^2 \\
&\leq \|x^k - x^\star\|_2^2 - 2t_k (f(x^k) - f^\star) + t_k^2 \|g^k\|_2^2
\end{aligned}
$$

using subgradient definition $f^\star = f(x^\star) \geq f(x^k) + (g^k)^T (x^\star - x^k)$

Convergence

Proof (continued)

Combine inequalities for $i = 0, \ldots, k$

$$\|x^{k+1} - x^\star\|_2^2 \leq \|x^0 - x^\star\|_2^2 - 2\sum_{i=0}^{k} t_i(f(x^i) - f^\star) + \sum_{i=0}^{k} t_i^2 \|g^i\|_2^2$$

$$\leq R^2 - 2\sum_{i=0}^{k} t_i(f(x^i) - f^\star) + G^2 \sum_{i=0}^{k} t_i^2$$

Using $\|x^{k+1} - x^\star\|_2^2 \geq 0$ we get

$$2\sum_{i=0}^{k} t_i(f(x^i) - f^\star) \leq R^2 + G^2 \sum_{i=0}^{k} t_i^2$$

Convergence

Proof (continued)

$$2\sum_{i=0}^{k} t_i(f(x^i) - f^\star) \le R^2 + G^2 \sum_{i=0}^{k} t_i^2$$

Combine it with

$$\sum_{i=0}^{k} t_i(f(x^i) - f(x^\star)) \ge \left(\sum_{i=0}^{k} t_i\right) \min_{i=0,\ldots,k}(f(x^i) - f^\star) = \left(\sum_{i=0}^{k} t_i\right)(f_{\text{best}}^k - f^\star)$$

to get

$$f_{\text{best}}^k - f^\star \le \frac{R^2 + G^2 \sum_{i=0}^{k} t_i^2}{2\sum_{i=0}^{k} t_i}$$

$\blacksquare$

Implications for step size rules

$$f_{\text{best}}^{k} - f^{\star} \leq \frac{R^2 + G^2 \sum_{i=0}^{k} t_i^2}{2 \sum_{i=0}^{k} t_i}$$

Fixed: $t_k = t$ for $k = 0, \ldots$

$$f_{\text{best}}^{k} - f^{\star} \leq \frac{R^2 + G^2 (k+1) t^2}{2(k+1)t}$$

May be suboptimal

$$\lim_{k \to \infty} f_{\text{best}}^{k} \leq f^{\star} + \frac{G^2 t}{2}$$

Diminishing: $$\sum_{k=0}^{\infty} t_k^2 < \infty, \quad \sum_{k=0}^{\infty} t_k = \infty$$

e.g., $t_k = \tau/(k+1)$ or $t_k = \tau/\sqrt{k+1}$

Optimal

$$\lim_{k \to \infty} f_{\text{best}}^{k} = f^{\star}$$

Optimal step size and convergence rate

For a tolerance $\epsilon > 0$, let's find the optimal t_k for a fixed k:

$$\frac{R^2 + G^2 \sum_{i=0}^k t_i^2}{2 \sum_{i=0}^k t_i} \leq \epsilon$$

Convex and symmetric in $(t_0, \dots, t_k)$
Hence, minimum when $t_i = t$

$$\longrightarrow \quad \frac{R^2 + G^2(k+1)t^2}{2(k+1)t}$$

Optimal choice $\qquad t = \dfrac{R}{G\sqrt{k+1}}$

Convergence rate

$$f_{\text{best}}^k - f^\star \leq \frac{RG}{\sqrt{k+1}}$$

Iterations required

$$k = O(1/\epsilon^2)$$

(gradient descent $k = O(1/\epsilon)$)

Stopping criterion

Terminating when

$$\frac{R^2 + G^2 \sum_{i=0}^{k} t_i^2}{2 \sum_{i=0}^{k} t_i} \leq \epsilon$$

is really, really slow.

Bad news

There is not really a good stopping criterion for the subgradient method

Optimal step size when $f^\star$ is known

Polyak step size

$$t_k = \frac{f(x^k) - f^\star}{\|g^k\|_2^2}$$

Motivation: minimize righthand side of

$$\|x^{k+1} - x^\star\|_2^2 \le \|x^k - x^\star\|_2^2 - 2t_k(f(x^k) - f^\star) + t_k^2\|g^k\|_2^2$$

Obtaining $\quad (f(x^k) - f^\star)^2 \le \left(\|x^{k+1} - x^\star\|_2^2 - \|x^k - x^\star\|_2^2\right) G^2$

Applying recursively, $\quad f_{\text{best}}^k - f^\star \le \dfrac{GR}{\sqrt{k+1}}$

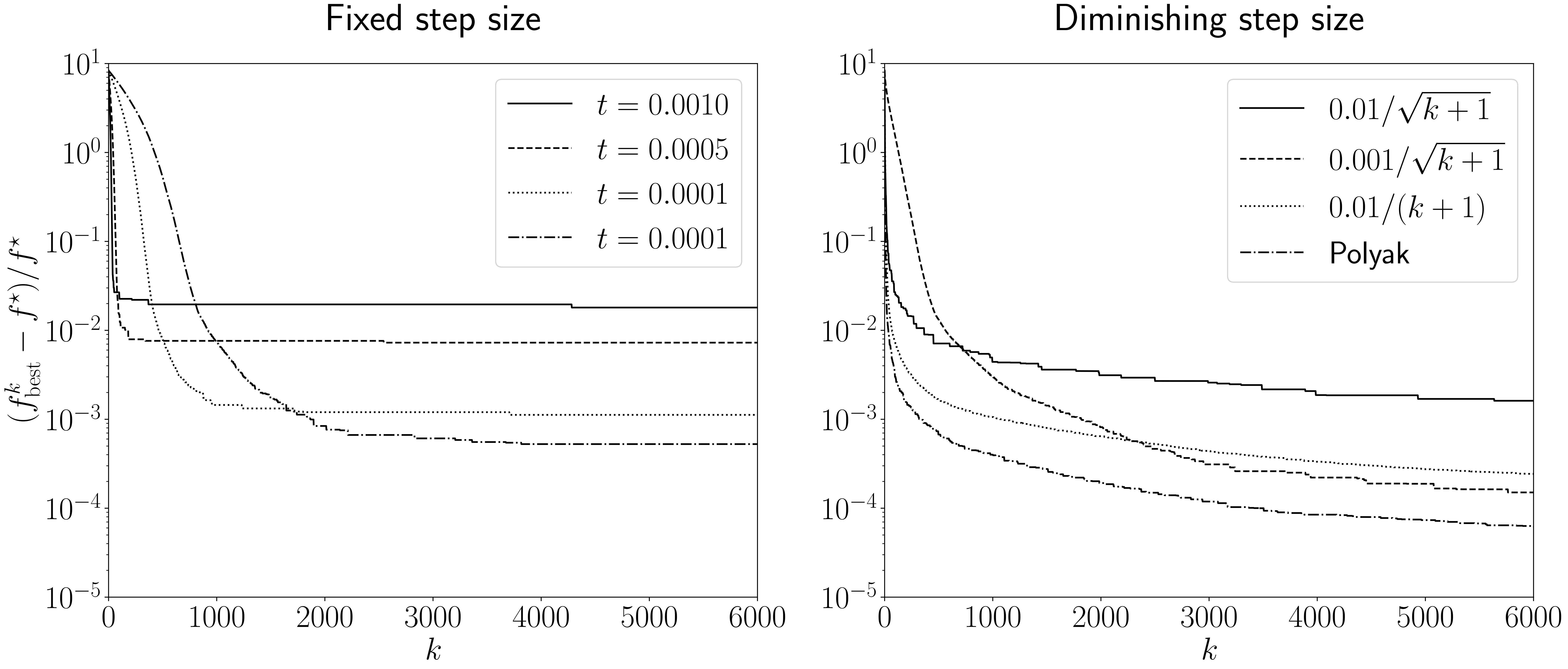
Iterations required

$k = O(1/\epsilon^2)$

still slow

Example: 1-norm minimization

minimize $\quad f(x) = \|Ax - b\|_1$ $\qquad g = A^T \mathbf{sign}(Ax - b) \in \partial f(x)$



Efficient packages to automatically compute (sub)gradients:
Python: JAX, PyTorch
Julia: Zygote.jl, ForwardDiff.jl, ReverseDiff.jl

Summary subgradient method

- Simple
- Handles general nondifferentiable convex functions
- Very slow convergence $O(1/\epsilon^2)$
- No good stopping criterion

Can we do better?

Can we incorporate constraints?

Proximal operators

Composite models

$$\text{minimize} \quad f(x) + g(x)$$

$f(x)$ convex and smooth
$g(x)$ convex (may be not differentiable)

Examples

- Regularized regression: $g(x) = \|x\|_1$
- Constrained optimization: $g(x) = \mathcal{I}_C(x)$

Proximal operator

Definition

The **proximal operator** of the function $g : \mathbf{R}^n \to \mathbf{R}$ is

$$\mathbf{prox}_g(x) = \underset{z}{\operatorname{argmin}} \left(g(z) + \frac{1}{2}\|z - x\|_2^2 \right)$$

Optimality conditions of prox

$$0 \in \partial g(z) + z - x \quad \Longrightarrow \quad x - z \in \partial g(z)$$

Properties

- It involves solving an optimization problem (not always easy!)
- Easy to evaluate for many standard functions, i.e. **proxable functions**
- Generalizes many well-known algorithms

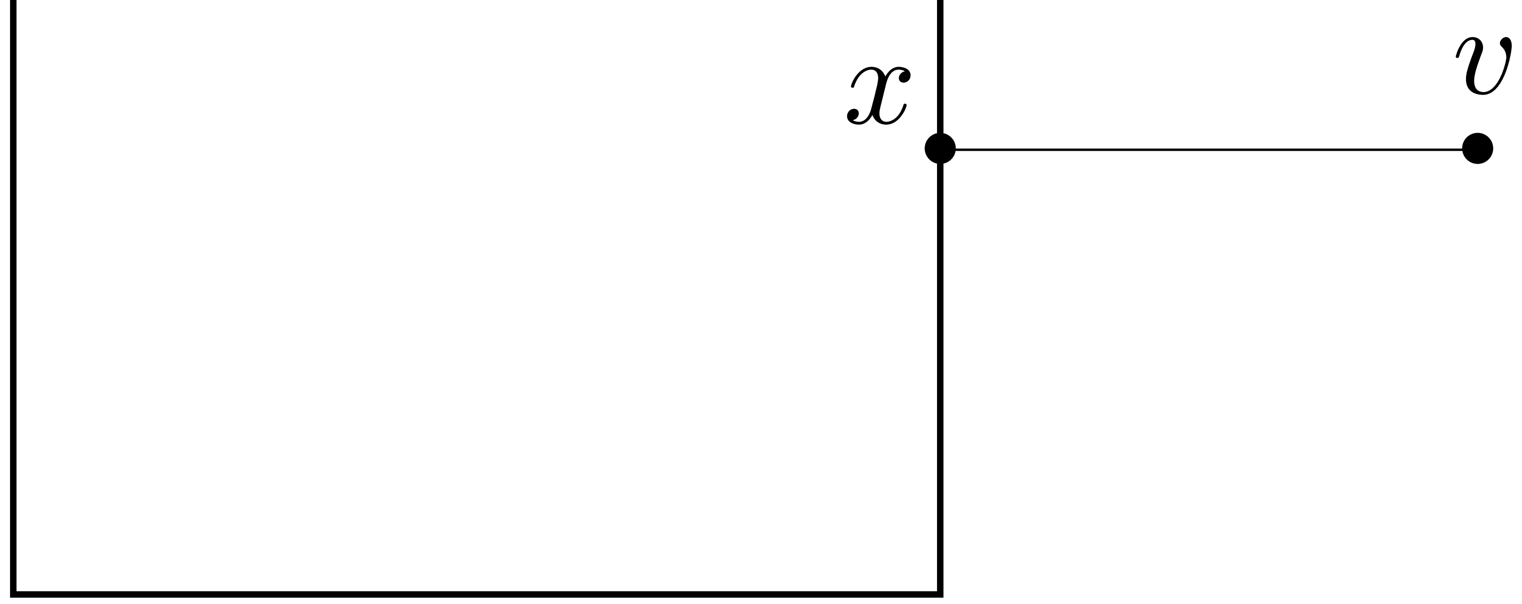
Generalized projection

The $\mathbf{prox}$ operator of the indicator function $\mathcal{I}_C$ is the projection onto C

$$\mathbf{prox}_{\mathcal{I}_C}(v) = \underset{x \in C}{\operatorname{argmin}} \|x - v\|_2 = \Pi_C(v)$$

Example projection onto a box $C = \{x \mid l \leq x \leq u\}$

$$\Pi_C(v)_i = \begin{cases} l_i & v_i \leq l_i \\ v_i & l_i \leq v_i \leq u_i \\ u_i & v_i \geq u_i \end{cases}$$



Remarks

- Easy for many common sets (e.g., closed form)
- Can be "hard" for surprisingly simple lets, e.g., $C = \{Ax \leq b\}$

Projections at [p. 156, FMO]

Quadratic functions

If $g(x) = (1/2)x^T Px + q^T x + r$ with $P \succeq 0$, then

$$\mathbf{prox}_g(v) = (I + P)^{-1}(v - q)$$

Remarks

- Closed-form always solvable (even with P not full rank)
- Symmetric, positive definite and usually sparse linear system
- Can prefactor $I + P$ and solve for different v

Separable sum

If $g(x)$ is block separable, i.e., $$g(x) = \sum_{i=1}^{N} g_i(x_i)$$

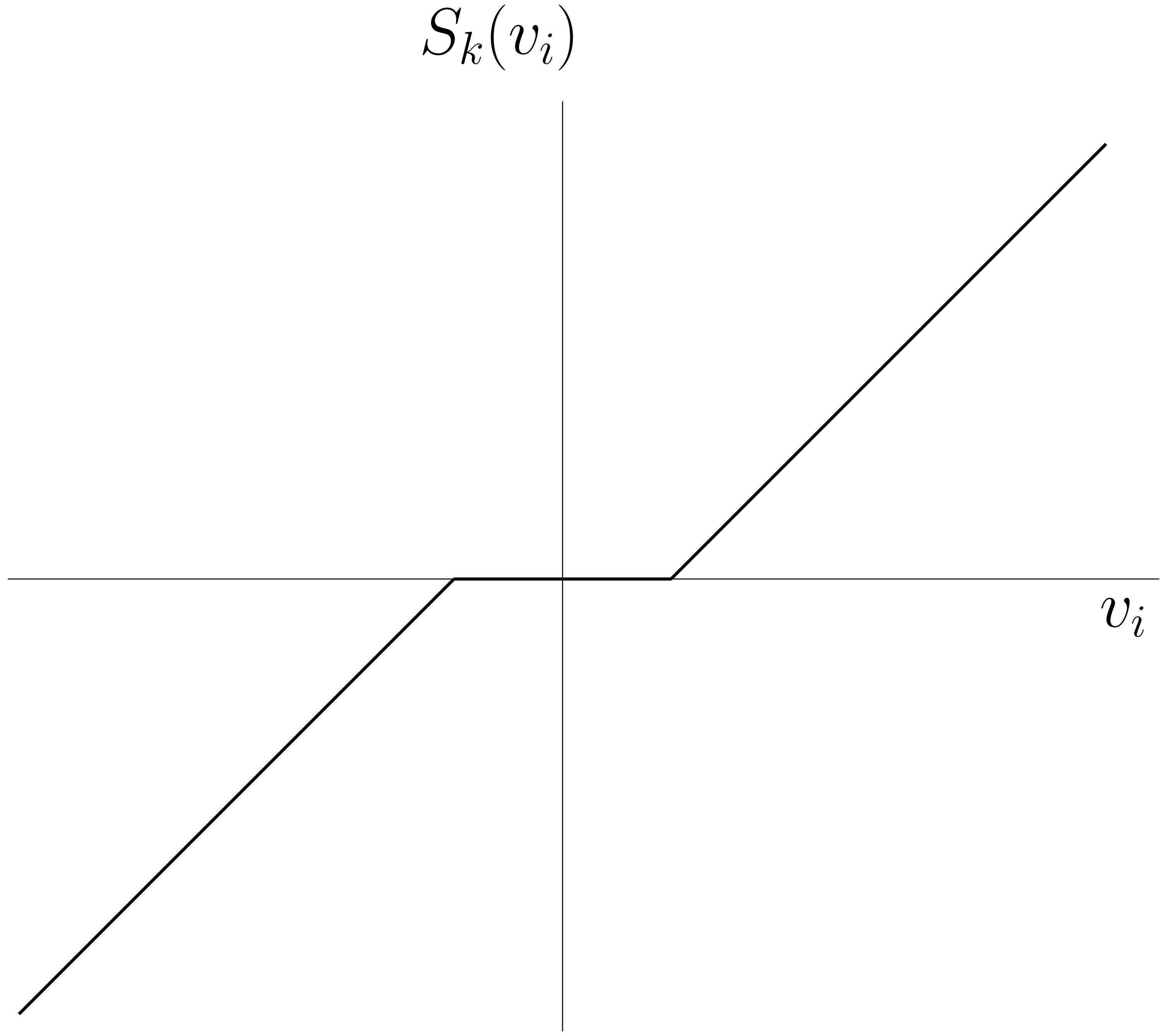
then, $$(\mathbf{prox}_g(v))_i = \mathbf{prox}_{g_i}(v_i), \quad i = 1, \ldots, N$$

(key to parallel/distributed proximal algorithms)

Example: $g(x) = \lambda \|x\|_1 = \sum_{i=1}^{n} \lambda |x_i|$

soft-thresholding

$$(\mathbf{prox}_g(v))_i = \mathbf{prox}_{\lambda|\cdot|}(v_i) = S_\lambda(v_i) = \begin{cases} v_i - \lambda & v_i > \lambda \\ 0 & |v_i| \leq \lambda \\ v_i + \lambda & v_i < -\lambda \end{cases}$$



Basic rules

Examples

- **Scaling and translation:** $g(x) = ah(x) + b$ with $a > 0$, then

$$\mathbf{prox}_g(x) = \mathbf{prox}_{ah}(x)$$

- **Affine addition:** $g(x) = h(x) + a^T x + b$, then

$$\mathbf{prox}_g(x) = \mathbf{prox}_h(x - a)$$

- **Affine transformation:** $g(x) = h(ax + b)$, with $a \neq 0, a \in \mathbf{R}$,

$$\mathbf{prox}_g(x) = \frac{1}{a}\left(\mathbf{prox}_{a^2 h}(ax + b) - b\right)$$

Proofs (exercise):

- Rearrange proximal term: $(1/2)\|z - x\|_2^2$
- Apply $\mathbf{prox}$ optimality conditions

Many more examples at [p. 156, FMO]

Proximal gradient method

Remember: gradient descent interpretation

Problem

$$\text{minimize} \quad f(x)$$

Iterations

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

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

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

Let's exploit the smooth part

minimize $\quad f(x) + g(x)$

$f(x)$ convex and smooth
$g(x)$ convex (may be not differentiable)

Quadratic approximation of f while keeping g

$$x^{k+1} = \underset{z}{\text{argmin}}\ g(z) + f(x^k) + \nabla f(x^k)^T(z - x^k) + \frac{1}{2t}\|z - x^k\|_2^2 \longleftarrow$$

same as
gradient descent

Equivalent to

Proximal operator

$$x^{k+1} = \underset{z}{\text{argmin}}\ tg(z) + \frac{1}{2}\left\|z - (x^k - t\nabla f(x^k))\right\|_2^2 = \mathbf{prox}_{tg}\left(x^k - t\nabla f(x^k)\right)$$

make g
small

stay close to
gradient update

Proximal gradient method

minimize $\quad f(x) + g(x)$

$f(x)$ convex and smooth
$g(x)$ convex (may be not differentiable)

Iterations

$$x^{k+1} = \mathbf{prox}_{tg}\left(x^k - t\nabla f(x^k)\right)$$

Properties

- Alternates between gradient updates of f and proximal updates on g
- Useful if $\mathbf{prox}_{tg}$ is inespensive
- Can handle nonsmooth and constrained problems

Special cases

Generalized gradient descent

Problem

minimize $\quad f(x) + g(x)$

Iterations

$$x^{k+1} = \mathbf{prox}_{tg}\left(x^k - t\nabla f(x^k)\right)$$

Smooth

$$g(x) = 0 \quad \Longrightarrow \quad \mathbf{prox}_{tg}(x) = x$$

Gradient descent

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

Constraints

$$g(x) = \mathcal{I}_C(x) \quad \Longrightarrow \quad \mathbf{prox}_{tg}(x) = \Pi_C(x)$$

Projected gradient descent

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

Non smooth

$$f(x) = 0$$

Proximal minimization

$$\Longrightarrow \quad x^{k+1} = \mathbf{prox}_{tg}(x^k)$$

Note: useful if $\mathbf{prox}_{tg}$ is cheap

What happens if we cannot evaluate the prox?

At every iteration, it can be very expensive to evaluate

$$\mathbf{prox}_g(x) = \underset{z}{\operatorname{argmin}} \left(g(z) + \frac{1}{2}\|z - x\|_2^2 \right)$$

Idea: solve it approximately!

If you precisely control the $\mathbf{prox}_g(x)$ evaluation errors
you can obtain the same convergence guarantees (and rates)
as the exact evaluations.

[Schmidt et al. (2011), "Convergence rates of inexact proximal-gradient methods for convex optimization"]

Example: Lasso

Iterative Soft Thresholding Algorithm (ISTA)

minimize $\quad \underbrace{(1/2)\|Ax - b\|_2^2}_{f(x)} + \underbrace{\lambda\|x\|_1}_{g(x)}$

Proximal gradient descent

$$x^{k+1} = \mathbf{prox}_{tg}\left(x^k - t\nabla f(x^k)\right)$$

$$\nabla f(x) = A^T(Ax - b)$$

$$\mathbf{prox}_{tg}(x) = S_{\lambda t}(x)$$

(component wise soft-thresholding)

Closed-form iterations

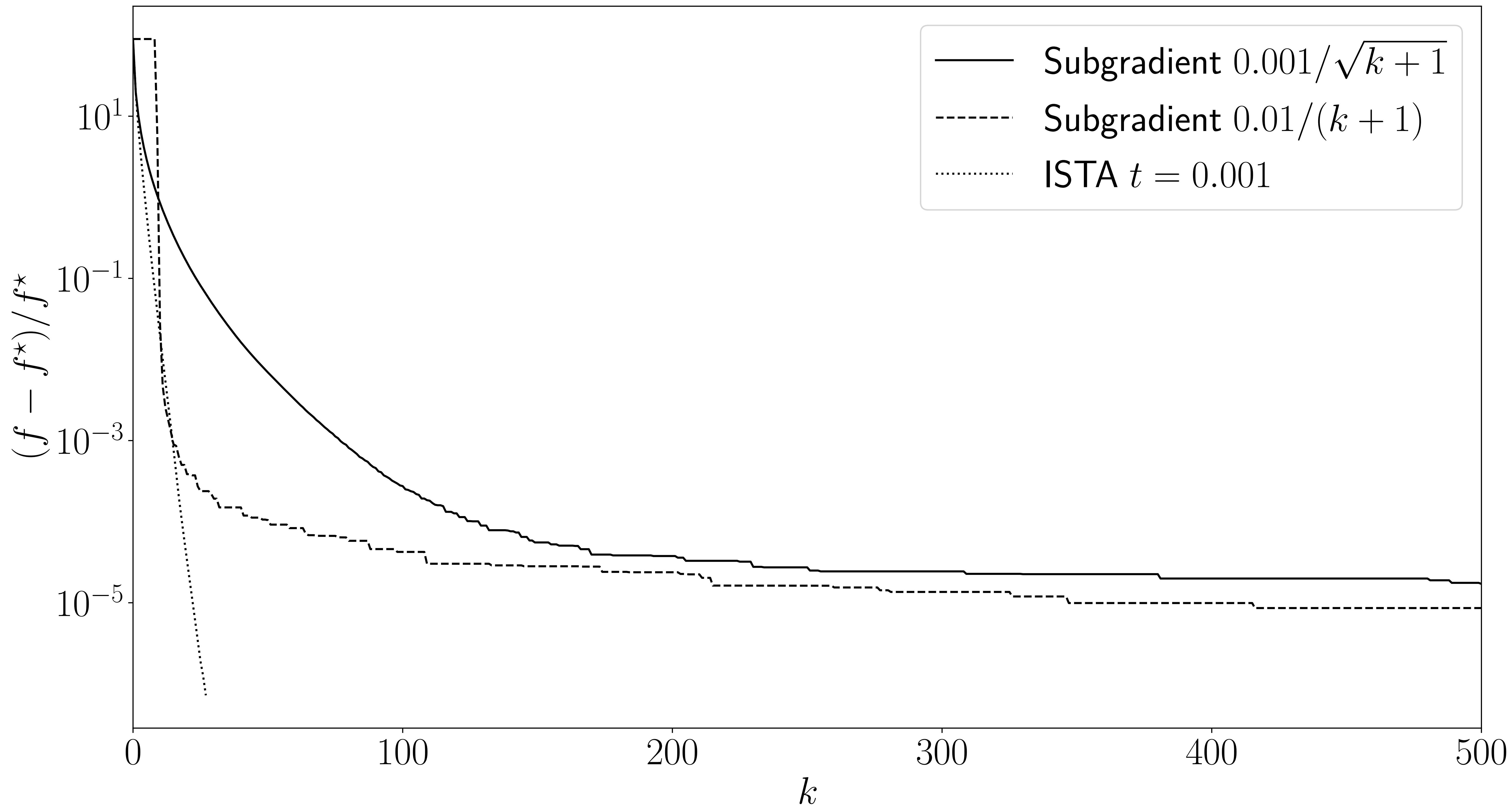
$$x^{k+1} = S_{\lambda t}\left(x^k - tA^T(Ax^k - b)\right)$$

Example: Lasso

Iterative Soft Thresholding Algorithm (ISTA)

$A \in \mathbf{R}^{500 \times 100}$

$$\text{minimize} \quad (1/2)\|Ax - b\|_2^2 + \lambda \|x\|_1$$



Closed-form iterations

$$x^{k+1} = S_{\lambda t}\left(x^k - tA^T(Ax^k - b)\right)$$

Better convergence

Can we prove convergence generally?

Can we combine different operators?

Proximal methods and introduction to operators

Today, we learned to:

- **Define** subgradient method and **analyze** its convergence
- **Derive** optimality conditions for constrained optimization problems using subdifferentials
- **Define** and **evaluate** proximal operators for various common functions
- **Apply** proximal operators to generalize gradient descent (vanilla, projected, proximal)

Next lecture

- Operator theory