KAUST

Proximal Splitting Methods for Convex Optimization: An Introduction

Laurent Condat

Visual Computing Center

King Abdullah University of Science and Technology (KAUST)

Dec. 2, 2019

Optimization

?

Optimization

optimize = do better

Optimization

optimize = do better

"Nothing takes place in the world whose meaning is not that of some maximum or minimum."
— **Leonhard Euler** ~1750

Optimization

optimize = do better

1844, «to act as an optimist»

Goal

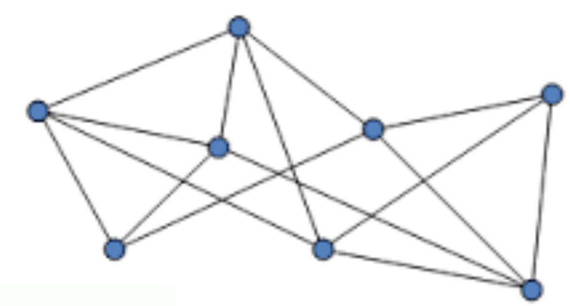
$$\text{Find } \tilde{x} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \Psi(x)$$

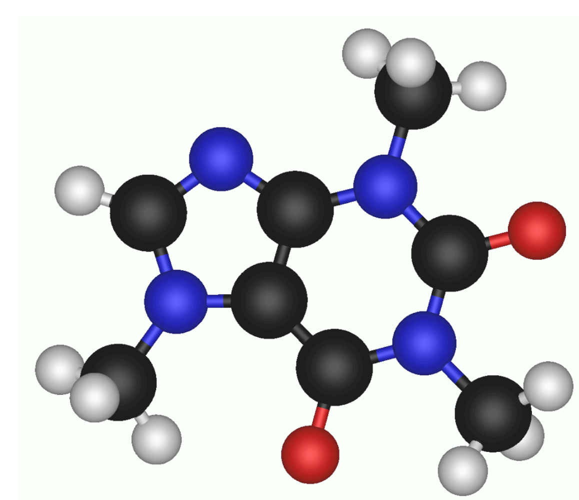
Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min}\ \Psi(x)$$

$\mathcal{X}$ is a real Hilbert space:

$$\begin{bmatrix} \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \\ \times & \times & \times & \times & \times \end{bmatrix}$$





Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min}\ \Psi(x)$$

Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min}\ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

where the L_m are linear operators

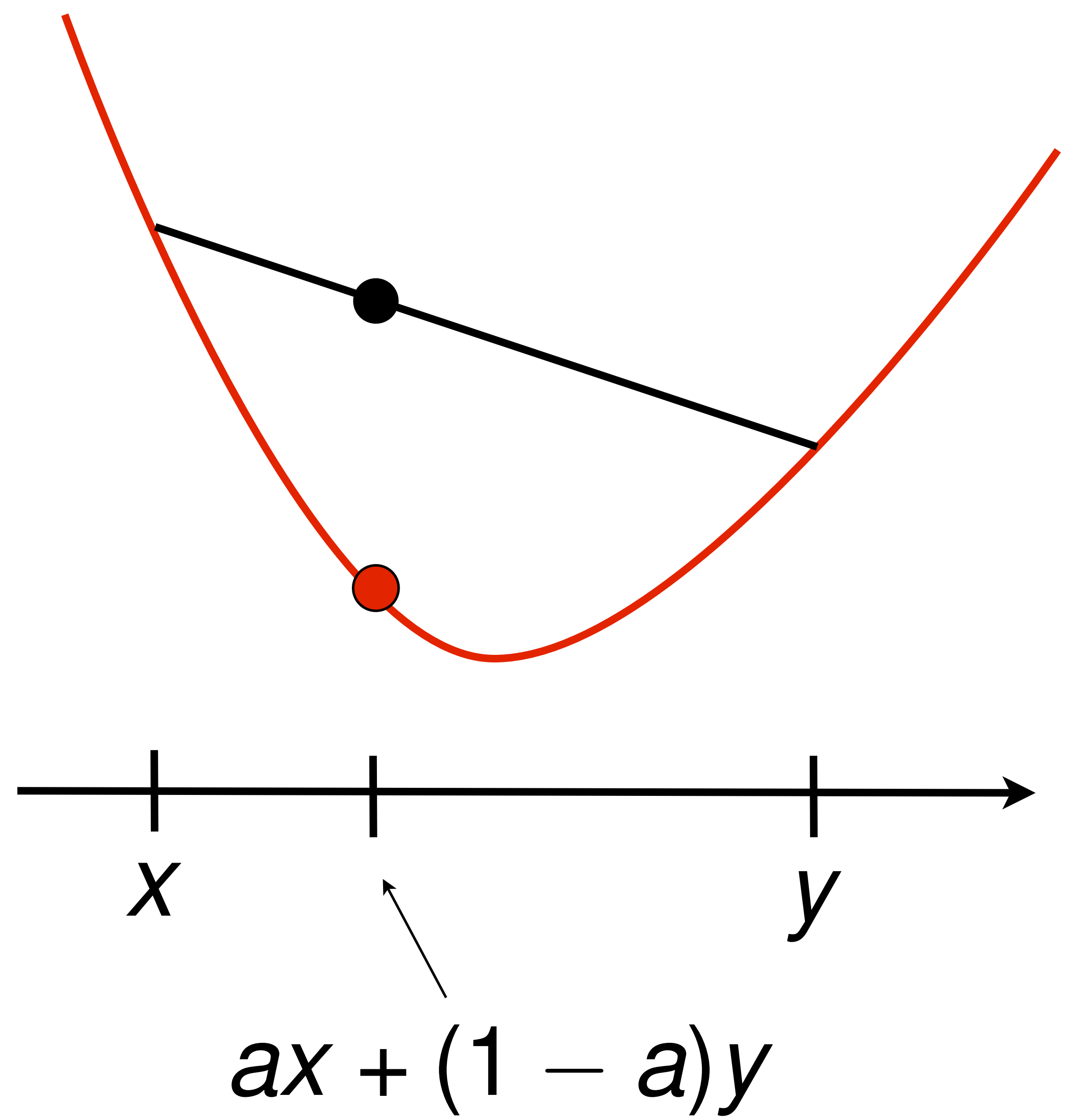
Goal

$$\text{Find } \tilde{x} \in \arg\min_{x \in \mathcal{X}} \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

The functions g_m are **convex**

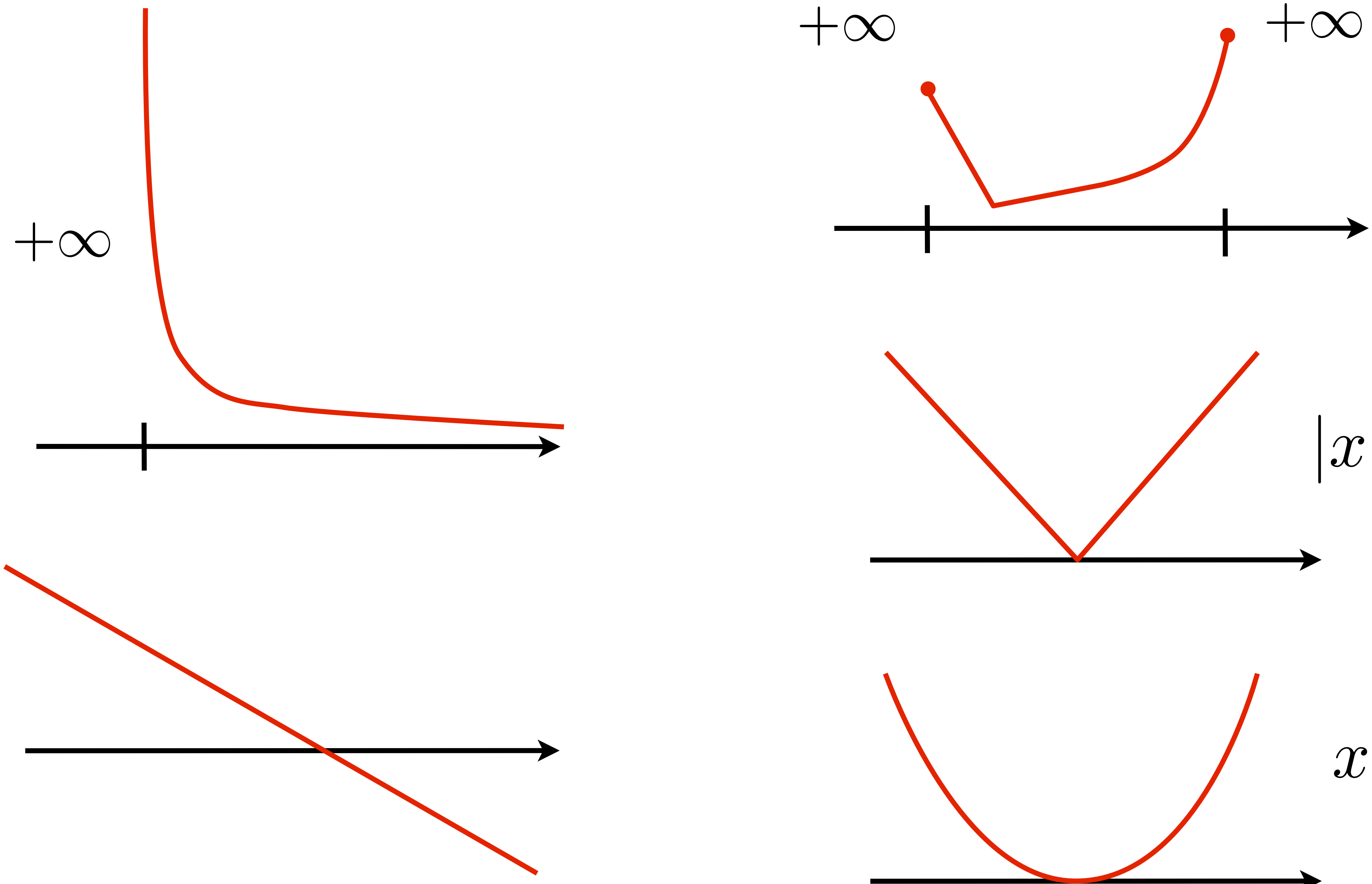
Convex functions

f is **convex** if $\forall x, y \in \mathcal{X}$ and $a \in [0,1]$,
$f(ax + (1-a)y) \le af(x) + (1-a)f(y)$



Convex functions

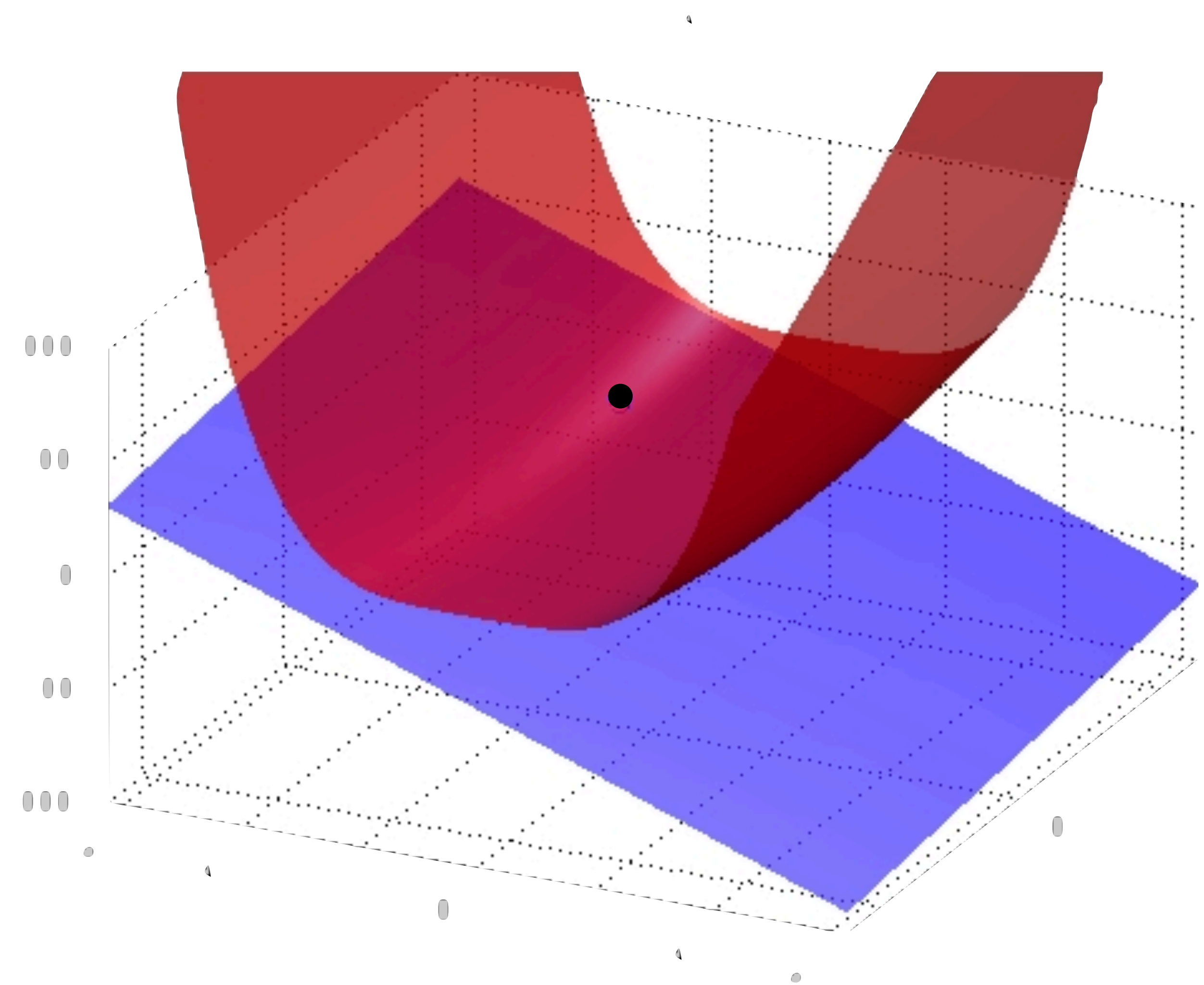
Some convex functions: $\mathbb{R} \to \mathbb{R} \cup \{+\infty\}$



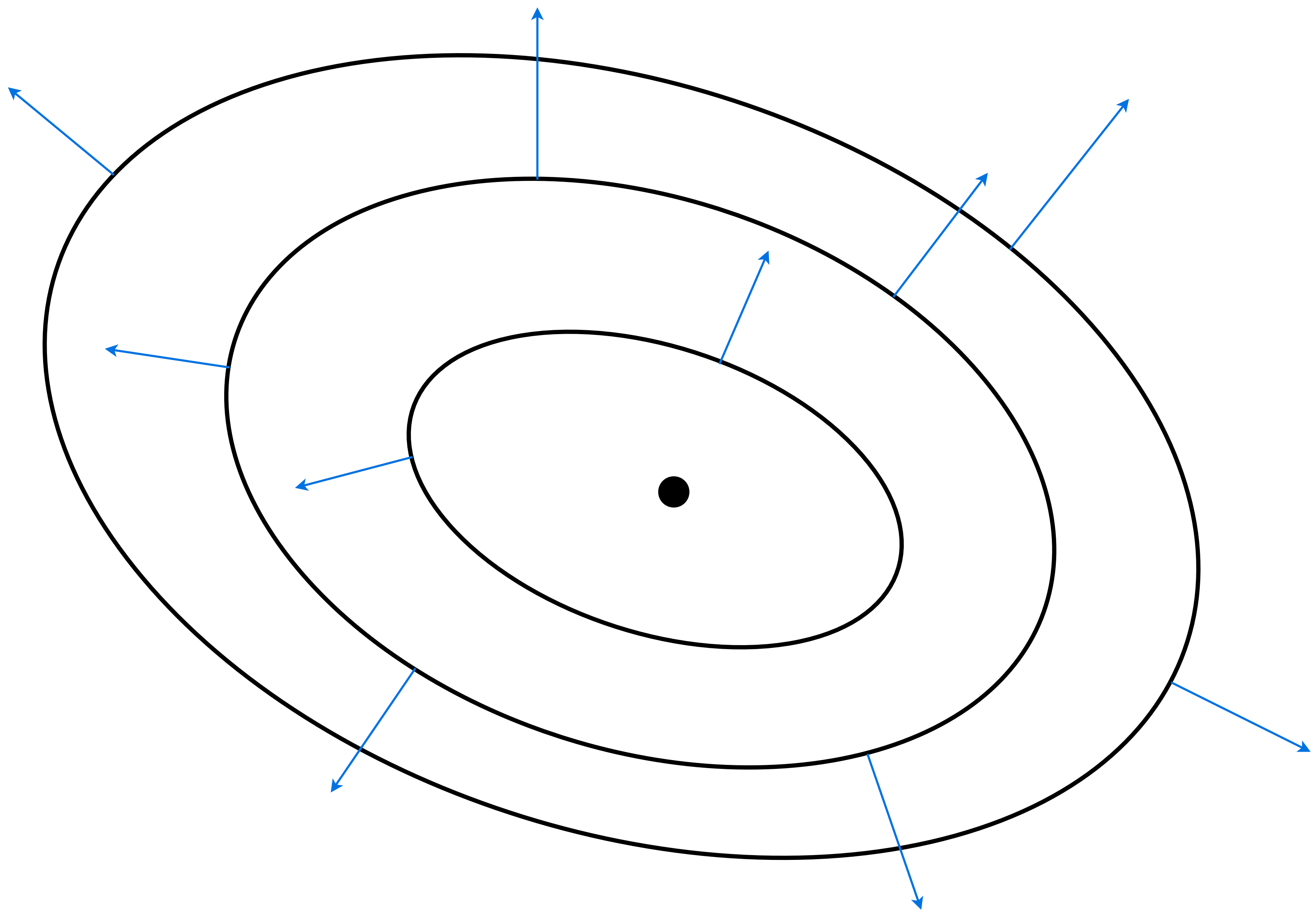
The gradient

$f : \mathcal{X} \to \mathbb{R}$ is differentiable (=smooth) at x
if there exists a unique element $\nabla f(x) \in \mathcal{X}$ such that

$$\forall e \in \mathcal{X}, f(x+e) = f(x) + \langle e, \nabla f(x) \rangle + o(\|e\|)$$

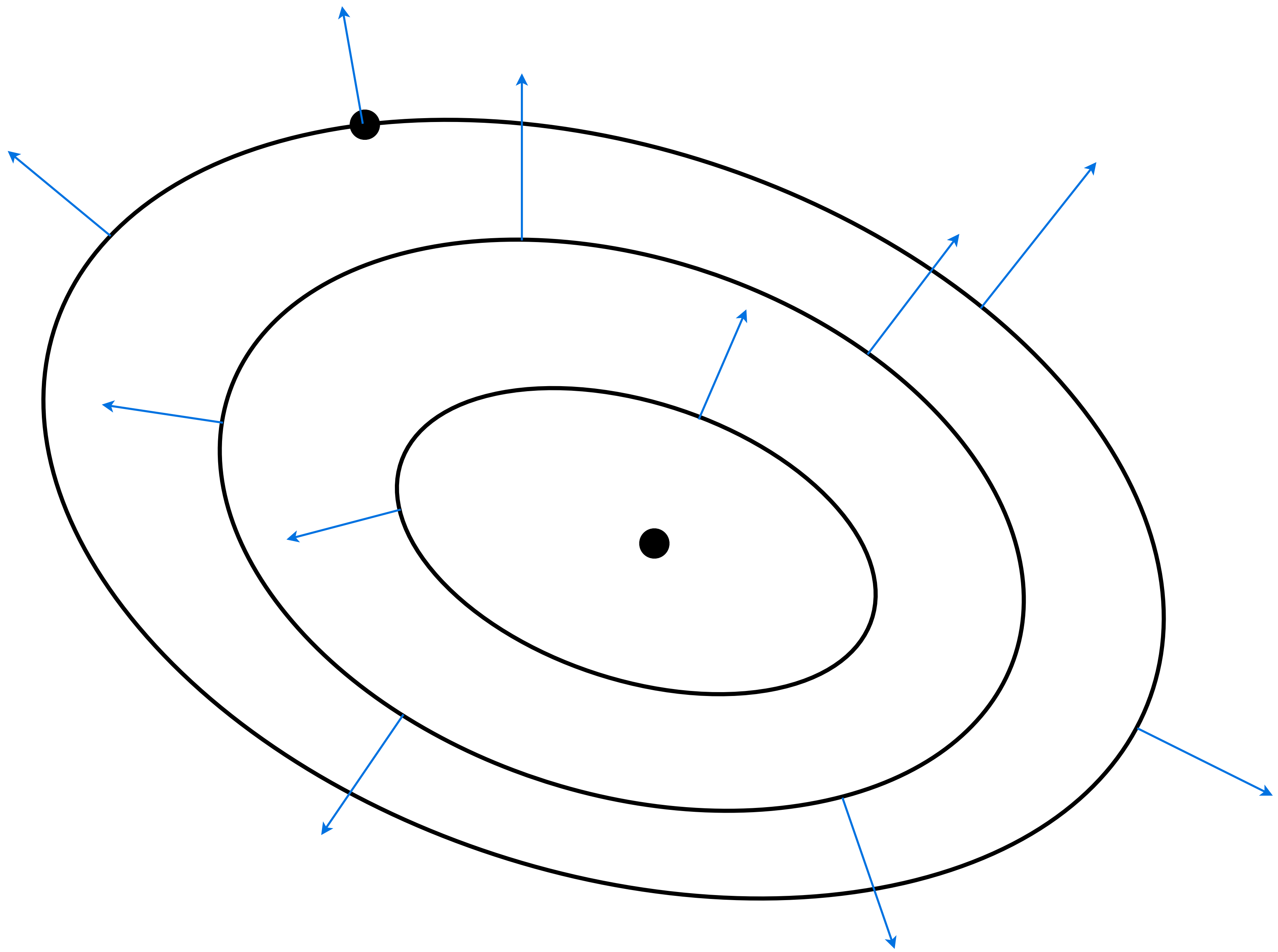


The gradient field



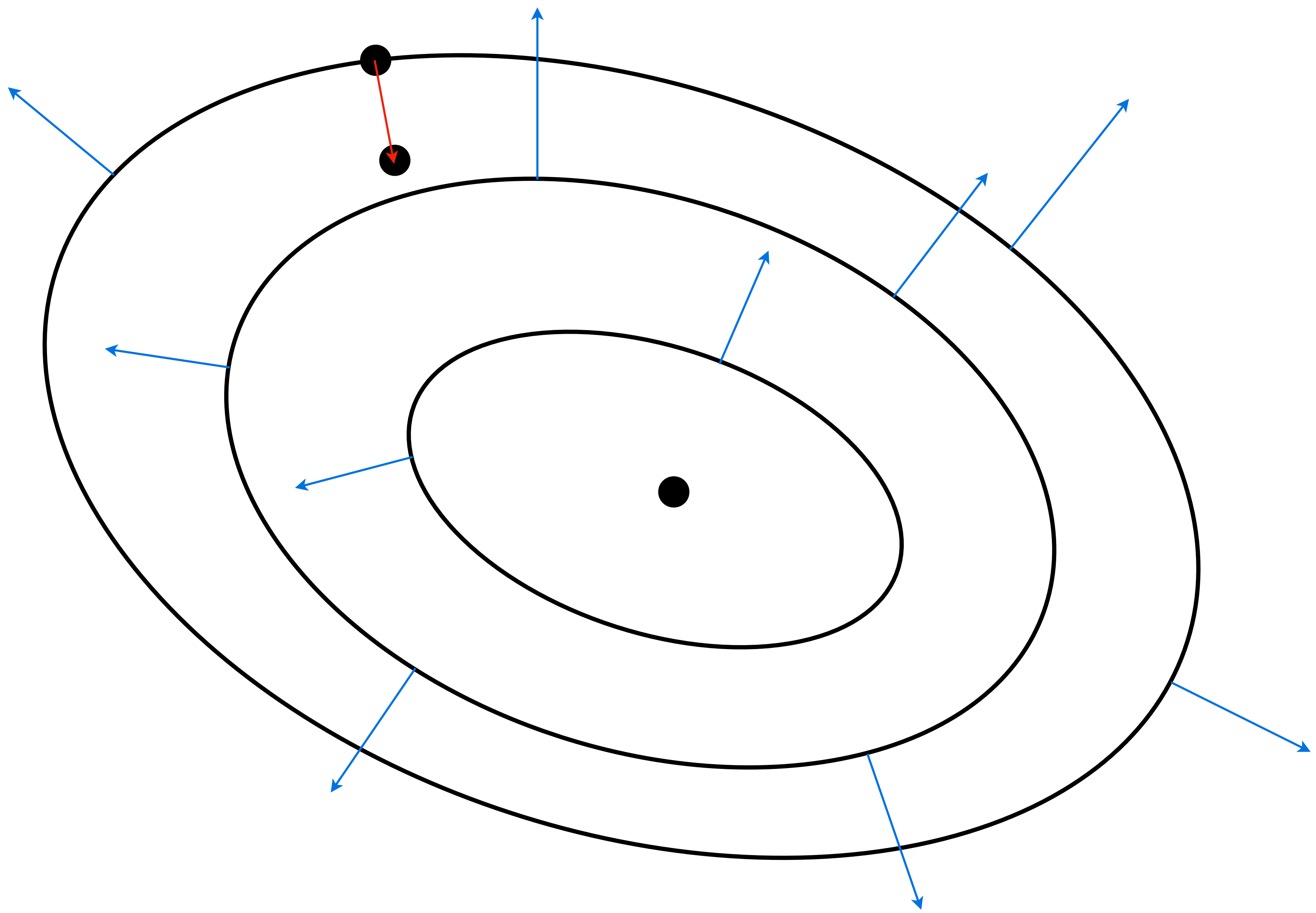
Gradient descent

$$x^{(i+1)} = x^{(i)} - \gamma \nabla(x^{(i)})$$



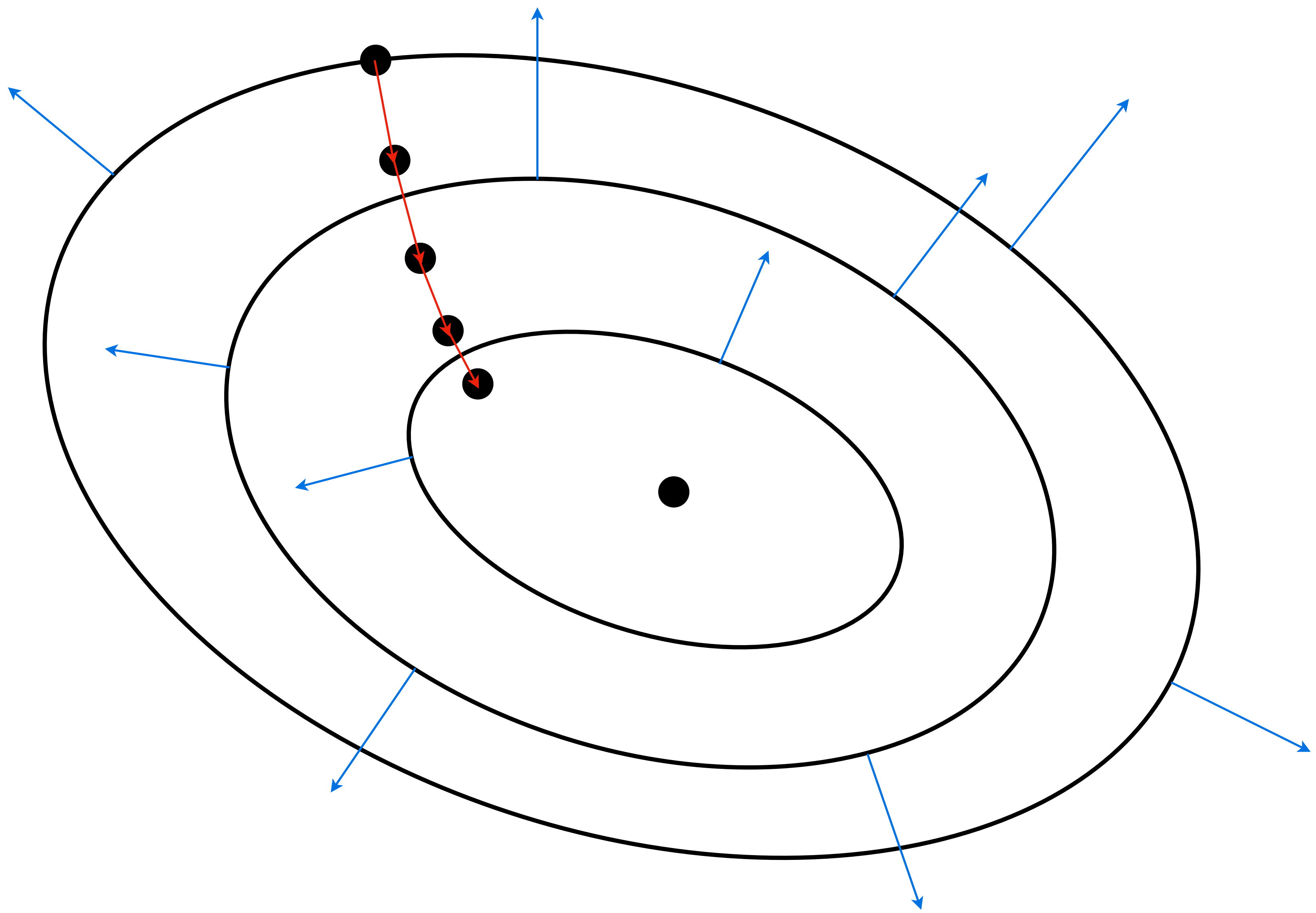
Gradient descent

$$x^{(i+1)} = x^{(i)} - \gamma\nabla(x^{(i)})$$

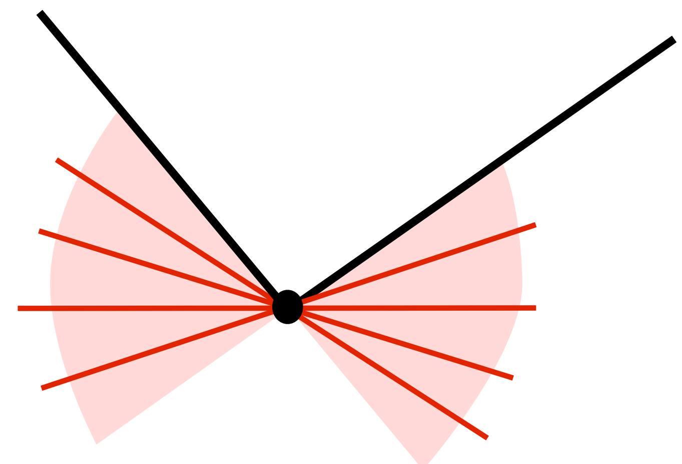


Gradient descent

$$x^{(i+1)} = x^{(i)} - \gamma \nabla(x^{(i)})$$



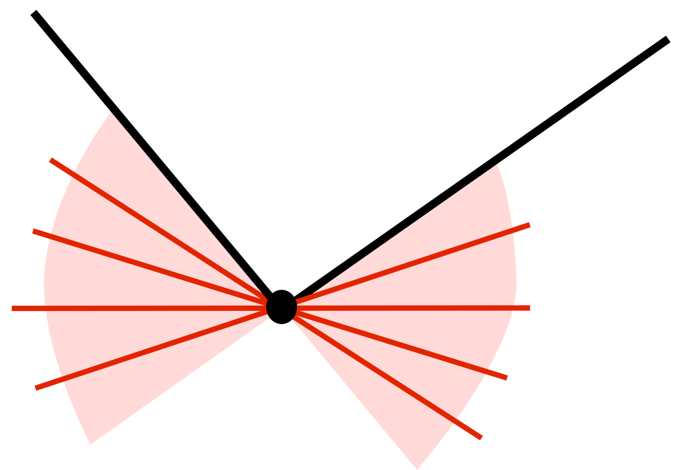
The subdifferential



$$\partial f : \mathcal{X} \to 2^{\mathcal{X}}$$
$$x \mapsto \{u \in \mathcal{X} \;:\; \forall y \in \mathcal{X},\; f(x) + \langle y - x, u\rangle \leq f(y)\}$$

☞ $\partial f(x)$ is the set of gradients of the affine minorants of f at x

The subdifferential

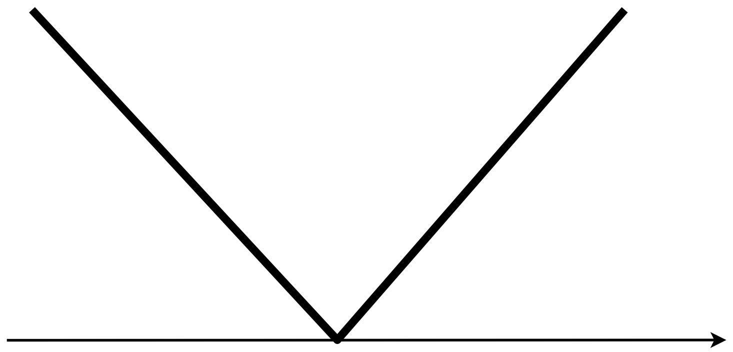


$$\partial f : \mathcal{X} \to 2^{\mathcal{X}}$$

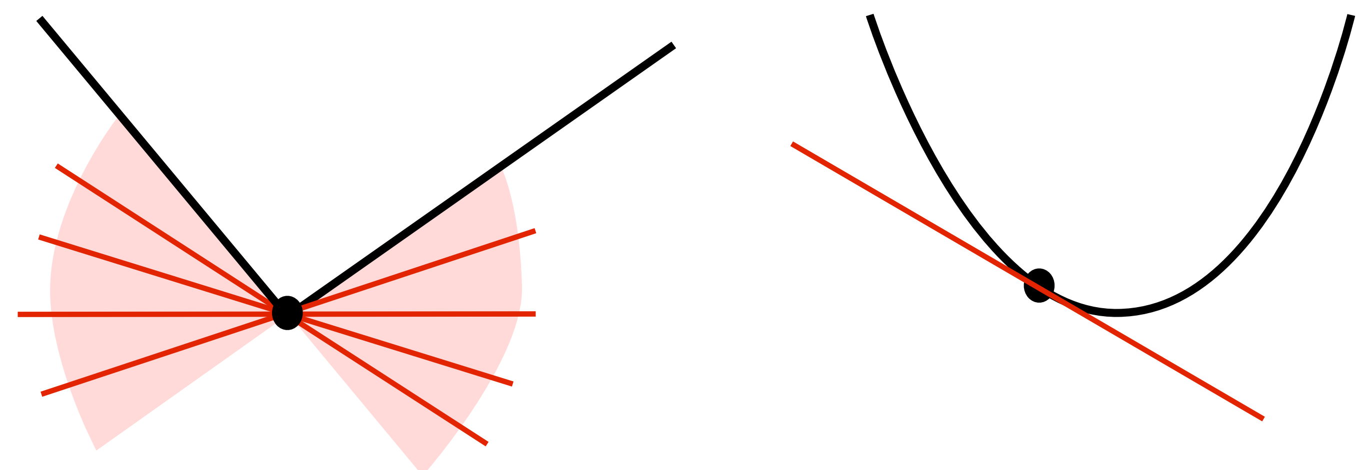
$$x \mapsto \{u \in \mathcal{X} \ : \ \forall y \in \mathcal{X},\ f(x) + \langle y - x, u \rangle \leq f(y)\}$$

Example: $f = |\cdot|$

☞ $\partial f(0) = [-1, 1]$



The subdifferential



$$\partial f : \mathcal{X} \to 2^{\mathcal{X}}$$

$$x \mapsto \{u \in \mathcal{X} \ : \ \forall y \in \mathcal{X},\ f(x) + \langle y - x, u \rangle \leq f(y)\}$$

f is convex and smooth at $x \to \partial f(x) = \{\nabla f(x)\}$.

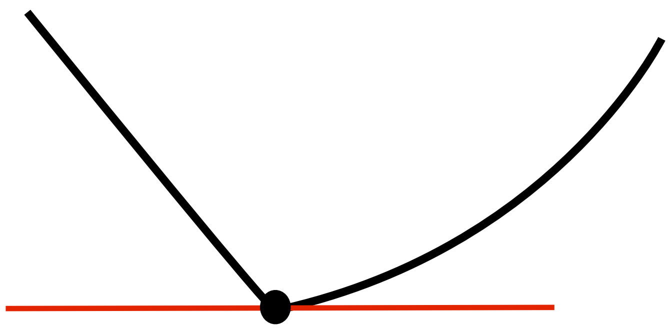
Fermat's rule

$$\underset{x \in \mathcal{X}}{\text{minimize}} \; f(x)$$

$$\equiv$$

find $\tilde{x} \in \mathcal{X}$ such that

$$0 \in \partial f(\tilde{x})$$



Pierre de Fermat, 1601-1665

Goal

$$\text{Find } \tilde{x} \in \underset{x \in \mathcal{X}}{\arg\min}\ \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

$$\sim\equiv$$

$$0 \in \sum_{m=1}^{M} L_m^* \partial g_m(L_m \tilde{x})$$

Goal

$$S = \arg\min_{x \in \mathcal{X}} \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

☞ Design an iterative algorithm which computes $x^{(i+1)} = T(x^{(i)})$
with $\|x^{(i)} - \tilde{x}\| \to 0$, for some $\tilde{x} \in S = \mathrm{Fix}\, T$

Goal

$$S = \arg\min_{x \in \mathcal{X}} \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

☞ Design an iterative algorithm which computes $x^{(i+1)} = T(x^{(i)})$
with $\|x^{(i)} - \tilde{x}\| \to 0$, for some $\tilde{x} \in S = \mathrm{Fix}\, T$

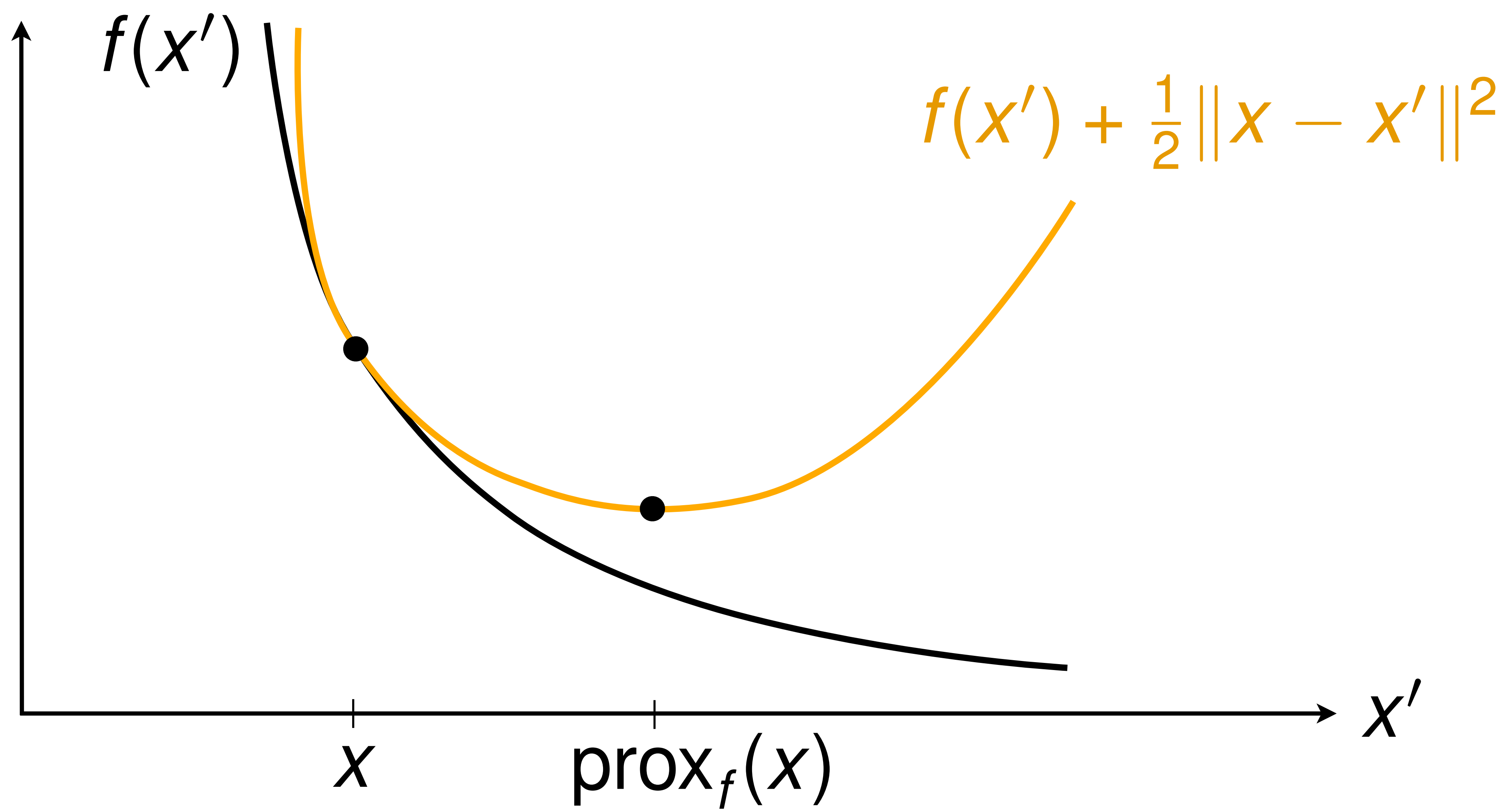
Example: $x^{(i+1)} = x^{(i)} - \gamma \nabla f(x^{(i)})$

Gradient descent

= ski in the night

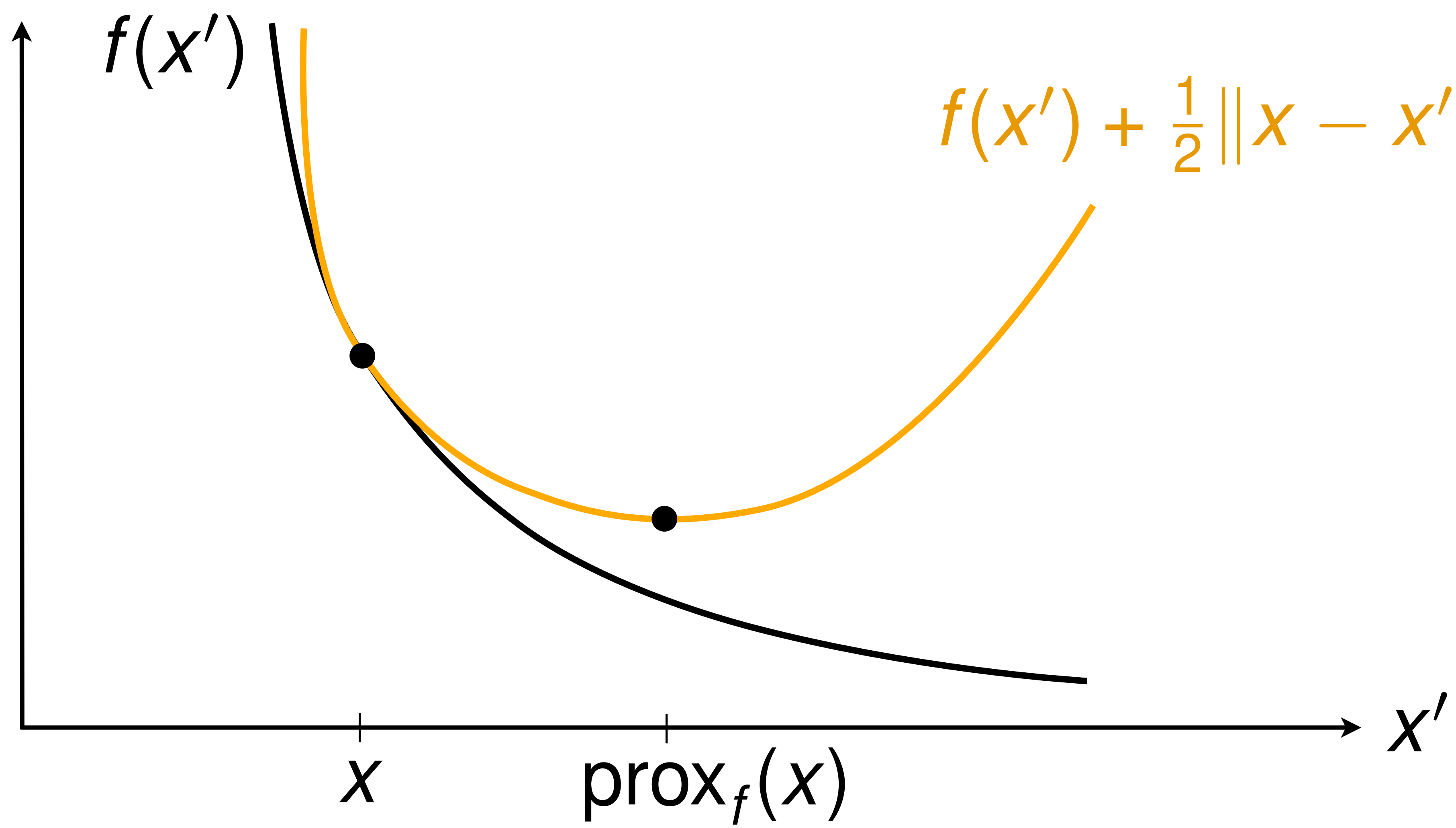
ski in the fog

The proximity operator



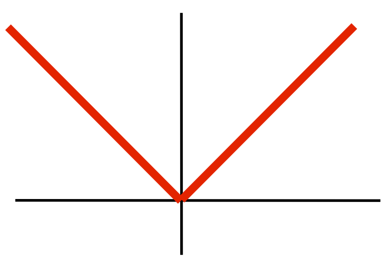
The proximity operator

$$\mathrm{prox}_f : \mathcal{H} \rightarrow \mathcal{H} : x \mapsto \operatorname*{arg\,min}_{x' \in \mathcal{H}} f(x') + \tfrac{1}{2}\|x - x'\|^2$$



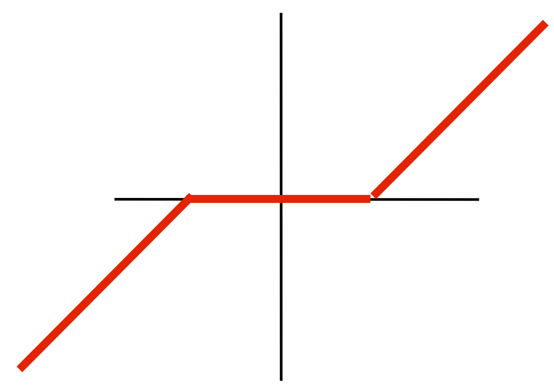
The proximity operator

$$\mathrm{prox}_f : \mathcal{H} \to \mathcal{H} : x \mapsto \underset{x' \in \mathcal{H}}{\arg\min}\, f(x') + \frac{1}{2}\|x - x'\|^2$$



$f(x) = |x|$

☞



$\mathrm{prox}_f(x) = \mathrm{sgn}(x)\max(|x| - 1, 0)$

The proximity operator

$$\mathrm{prox}_f : \mathcal{H} \to \mathcal{H} : x \mapsto \underset{x' \in \mathcal{H}}{\arg\min}\, f(x') + \tfrac{1}{2}\|x - x'\|^2$$

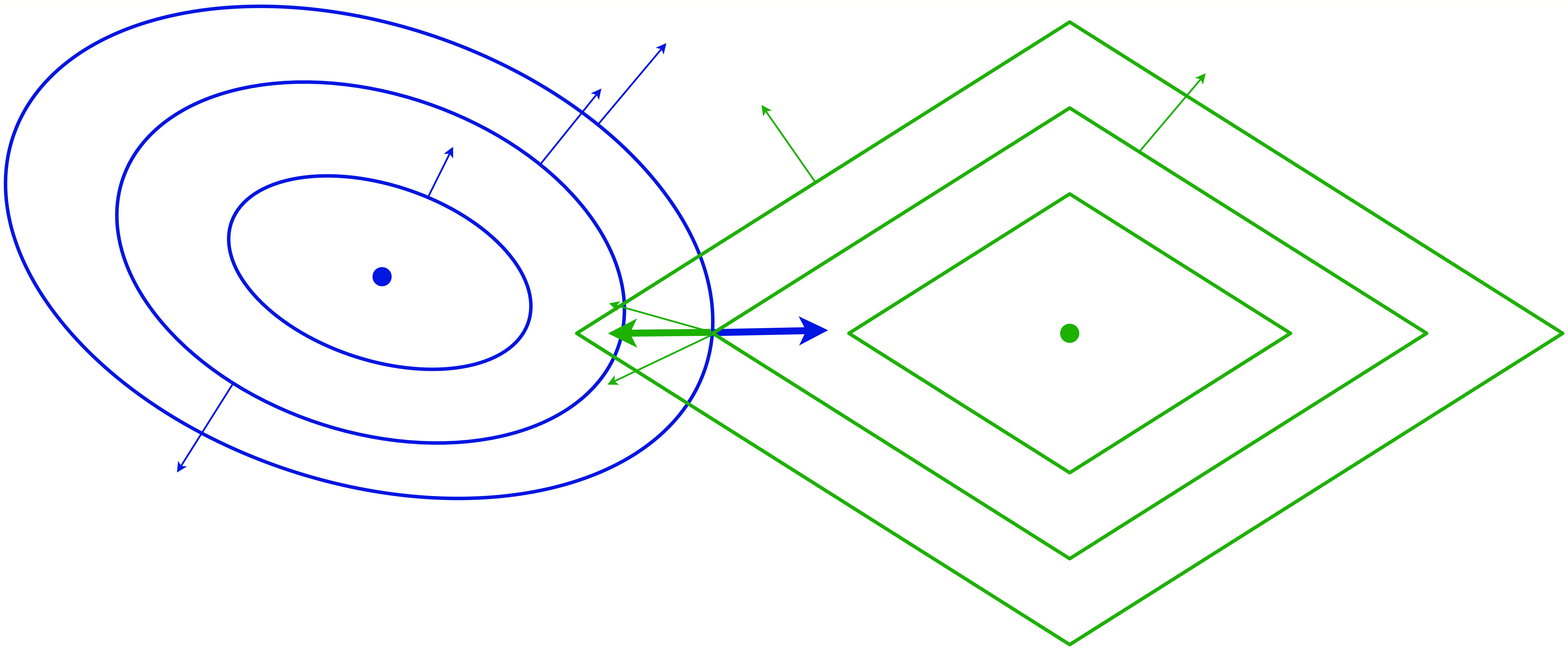
$$\mathrm{prox}_f = (\partial f + \mathrm{Id})^{-1}$$

Goal

$$\text{Find } \tilde{x} \in \arg\min_{x \in \mathcal{X}} \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

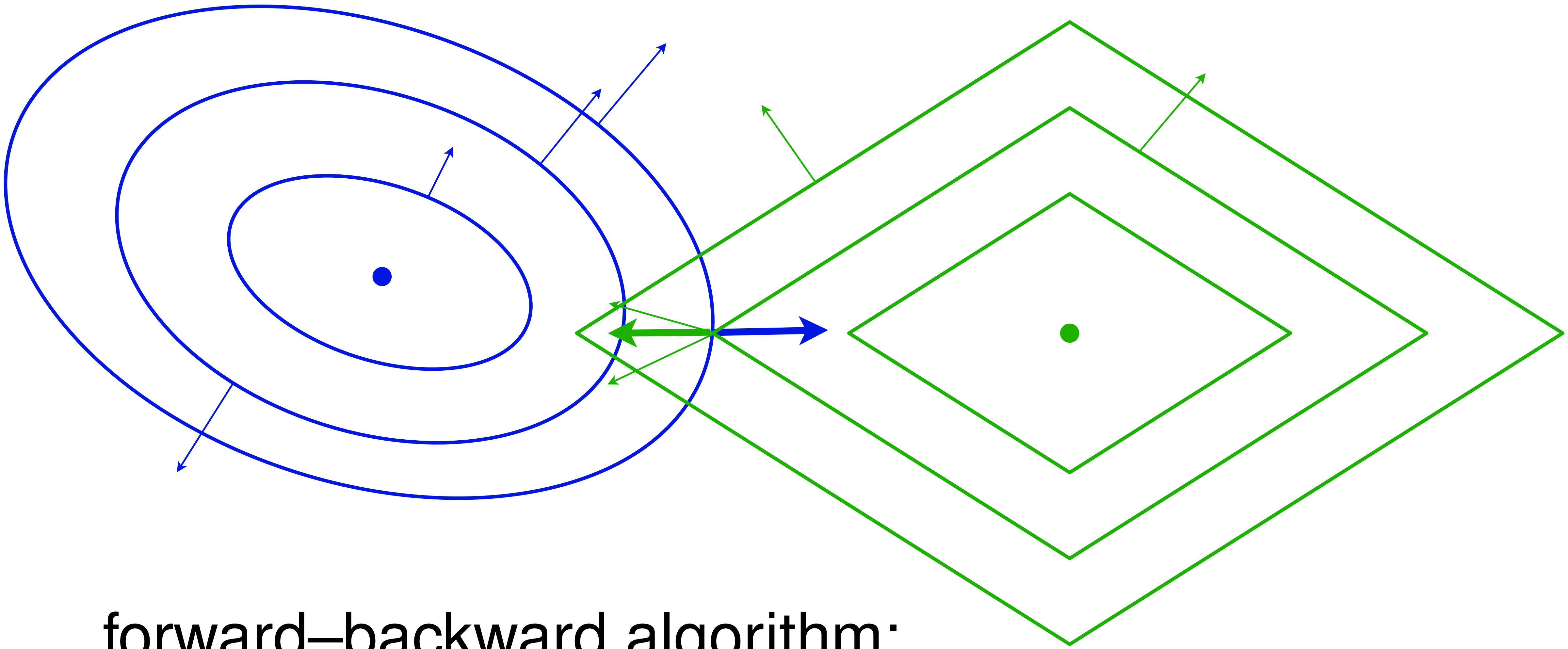
We want full splitting, with individual activation of L_m, L_m^*, the gradient or proximity operator of g_m.

Minimization of 2 functions



$$\min h + f \quad \equiv \quad 0 \in \nabla h(x) + \partial f(x)$$

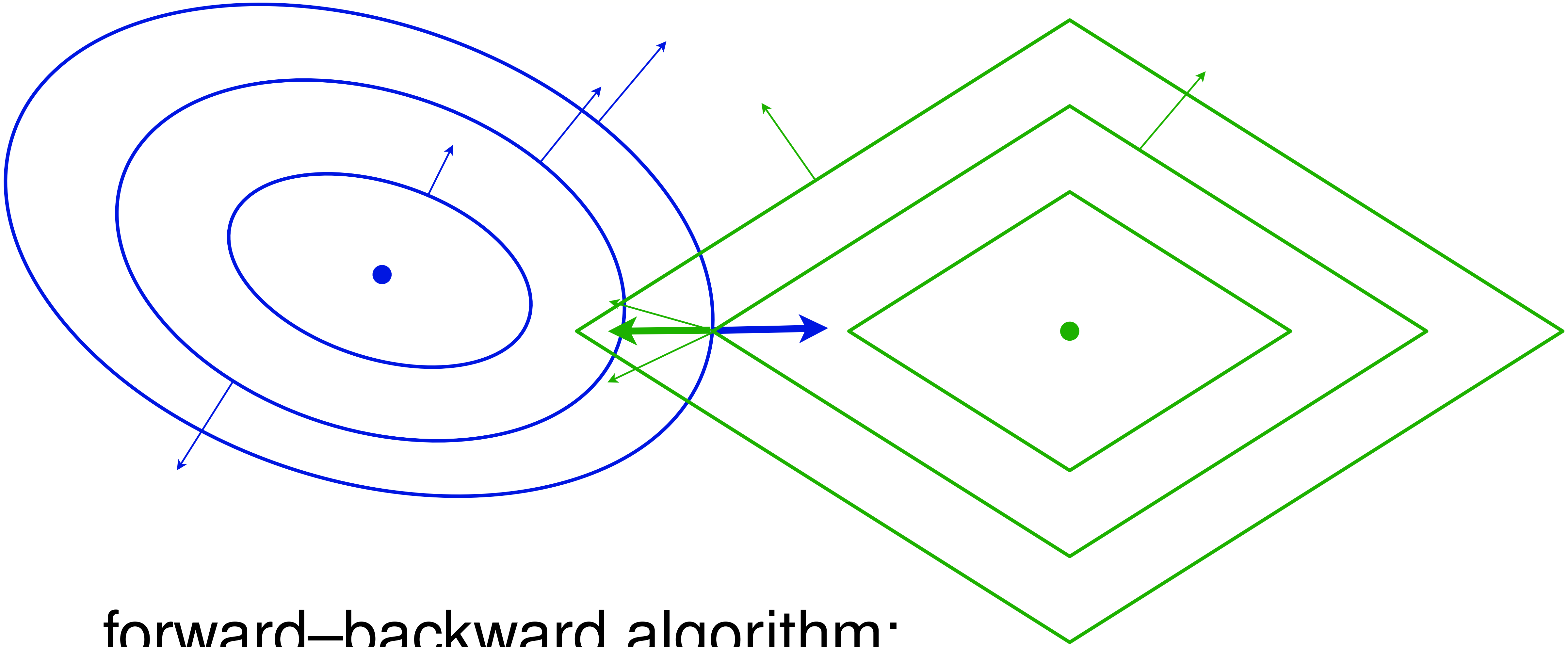
Minimization of 2 functions



forward–backward algorithm:

$$x^{(i+1)} = \operatorname{prox}_{\gamma g}\big(x^{(i)} - \gamma \nabla h(x^{(i)})\big)$$

Minimization of 2 functions



forward–backward algorithm:

$$x^{(i+1)} = \mathrm{prox}_{\gamma g}\big(x^{(i)} - \gamma \nabla h(x^{(i)})\big)$$

$$\equiv \quad 0 \in \partial g(x^{(i+1)}) + \nabla h(x^{(i)}) + (\tfrac{1}{\gamma}\mathrm{Id})(x^{(i+1)} - x^{(i)})$$

Forward-backward splitting

More generally, to solve

$0 \in M(x) + C(x)$

where M is maximally monotone, C is cocoercive, $P \succ 0$

☞ forward–backward algorithm:

$$0 \in M(x^{(i+1)}) + C(x^{(i)}) + P(x^{(i+1)} - x^{(i)})$$

Goal

$$\text{Find } \tilde{x} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \; \Psi(x) = \sum_{m=1}^{M} g_m(L_m x)$$

Goal

$$\text{Find } \tilde{x} \in \operatorname*{arg\,min}_{x \in \mathcal{X}} \left\{ h(x) + f(x) + g(Lx) \right\}$$

using ∇h, prox_f, prox_g

Goal

Find $\tilde{x}$ solution to

$$0 \in \nabla h(x) + \partial f(x) + L^* \partial g(Lx)$$

$$\equiv$$

Find $(\tilde{x}, \tilde{u})$ solution to

$$\begin{cases} u \in \partial g(Lx) \\ 0 \in \nabla h(x) + \partial f(x) + L^* u \end{cases}$$

Goal

Find $\tilde{x}$ solution to

$$0 \in \nabla h(x) + \partial f(x) + L^* \partial g(Lx)$$

$$\equiv$$

Find $(\tilde{x}, \tilde{u})$ solution to

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \begin{pmatrix} \partial f(x) + L^* u \\ -Lx + (\partial g)^{-1}(u) \end{pmatrix} + \begin{pmatrix} \nabla h(x) \\ 0 \end{pmatrix}$$

Goal

Find $\tilde{x}$ solution to

$$0 \in \nabla h(x) + \partial f(x) + L^* \partial g(Lx)$$

$$\equiv$$

Find $(\tilde{x}, \tilde{u})$ solution to

$$\begin{pmatrix} 0 \\ 0 \end{pmatrix} \in \underbrace{\begin{pmatrix} \partial f(x) + L^* u \\ -Lx + (\partial g)^{-1}(u) \end{pmatrix}}_{\text{max. monotone}} + \underbrace{\begin{pmatrix} \nabla h(x) \\ 0 \end{pmatrix}}_{\text{cocoercive}}$$

Primal-dual forward-backward splitting

☞ forward–backward algorithm:

$$\begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(x^{(i+1)}) + L^* u^{(i+1)}\\ -Lx^{(i+1)} + \partial g^*(u^{(i+1)})\end{pmatrix} + \begin{pmatrix}\nabla h(x^{(i)})\\ 0\end{pmatrix} + \begin{pmatrix}? & ?\\ ? & ?\end{pmatrix}\begin{pmatrix}x^{(i+1)} - x^{(i)}\\ u^{(i+1)} - u^{(i)}\end{pmatrix}$$

Primal-dual forward-backward splitting

☞ forward–backward algorithm:

$$\begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(x^{(i+1)}) + L^* u^{(i+1)}\\ -Lx^{(i+1)} + \partial g^*(u^{(i+1)})\end{pmatrix} + \begin{pmatrix}\nabla h(x^{(i)})\\ 0\end{pmatrix} + \begin{pmatrix}\frac{1}{\tau}\mathrm{Id} & ?\\ ? & \frac{1}{\sigma}\mathrm{Id}\end{pmatrix}\begin{pmatrix}x^{(i+1)} - x^{(i)}\\ u^{(i+1)} - u^{(i)}\end{pmatrix}$$

Primal-dual forward-backward splitting

☞ forward–backward algorithm:

$$\begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(x^{(i+1)}) + L^* u^{(i+1)}\\ -Lx^{(i+1)} + \partial g^*(u^{(i+1)})\end{pmatrix} + \begin{pmatrix}\nabla h(x^{(i)})\\ 0\end{pmatrix} + \begin{pmatrix}\frac{1}{\tau}\mathrm{Id} & -L^*\\ ? & \frac{1}{\sigma}\mathrm{Id}\end{pmatrix}\begin{pmatrix}x^{(i+1)} - x^{(i)}\\ u^{(i+1)} - u^{(i)}\end{pmatrix}$$

$$x^{(i+1)} = \mathrm{prox}_{\tau f}\left(x^{(i)} - \tau\nabla h(x^{(i)}) - \tau L^* u^{(i)}\right)$$

Primal-dual forward-backward splitting

☞ forward–backward algorithm:

$$\begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(x^{(i+1)}) + L^* u^{(i+1)}\\ -Lx^{(i+1)} + \partial g^*(u^{(i+1)})\end{pmatrix} + \begin{pmatrix}\nabla h(x^{(i)})\\ 0\end{pmatrix} + \begin{pmatrix}\frac{1}{\tau}\mathrm{Id} & -L^*\\ -L & \frac{1}{\sigma}\mathrm{Id}\end{pmatrix} \begin{pmatrix}x^{(i+1)} - x^{(i)}\\ u^{(i+1)} - u^{(i)}\end{pmatrix}$$

$$\left\lfloor \begin{array}{l} x^{(i+1)} = \mathrm{prox}_{\tau f}\big(x^{(i)} - \tau\nabla h(x^{(i)}) - \tau L^* u^{(i)}\big) \\ u^{(i+1)} = \mathrm{prox}_{\sigma g^*}\big(u^{(i)} + \sigma L(2x^{(i+1)} - x^{(i)})\big) \end{array}\right.$$

Primal-dual forward-backward splitting

☞ forward–backward algorithm:

$$\begin{pmatrix}0\\0\end{pmatrix} \in \begin{pmatrix}\partial f(x^{(i+1)}) + L^* u^{(i+1)}\\ -Lx^{(i+1)} + \partial g^*(u^{(i+1)})\end{pmatrix} + \begin{pmatrix}\nabla h(x^{(i)})\\ 0\end{pmatrix} + \begin{pmatrix}\frac{1}{\tau}\mathrm{Id} & -L^*\\ -L & \frac{1}{\sigma}\mathrm{Id}\end{pmatrix}\begin{pmatrix}x^{(i+1)} - x^{(i)}\\ u^{(i+1)} - u^{(i)}\end{pmatrix}$$

$$\left\lfloor\begin{array}{l} x^{(i+1)} = \mathrm{prox}_{\tau f}\left(x^{(i)} - \tau\nabla h(x^{(i)}) - \tau L^* u^{(i)}\right)\\ u^{(i+1)} = \mathrm{prox}_{\sigma g^*}\left(u^{(i)} + \sigma L(2x^{(i+1)} - x^{(i)})\right)\end{array}\right.$$

Convergence if $\frac{1}{\tau} - \sigma\|L\|^2 \geq \frac{\beta}{2}$

Particular cases

- $g \circ L = 0 \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + h(x) \right\}$

☞ **forward–backward** algorithm

- $h = 0 \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + g(Lx) \right\}$

☞ **Chambolle–Pock** algorithm

- $h = 0$ and $L = \mathrm{Id} \;\rightarrow\; \underset{x \in \mathcal{H}}{\text{minimize}} \left\{ f(x) + g(x) \right\}$

☞ **Douglas–Rachford** ($\equiv$ **ADMM**) algorithm

Summary

splitting methods
to solve monotone inclusions

make it
possible
to design
new

primal-dual proximal algorithms
to solve convex optimization problems

can be
explained
by

Bibliography

[L. Condat, D. Kitahara, A. Contreras, and A. Hirabayashi, "Proximal Splitting Algorithms: Overrelax them all!," preprint, 2019]

[L. Condat, "A primal-dual splitting method for convex optimization involving Lipschitzian, proximable and linear composite terms," *J. Optimization Theory and Applications*, 2013]

[L. Condat, "A Generic Proximal Algorithm for Convex Optimization — Application to Total Variation Minimization," *IEEE Signal Proc. Letters*, 2014]

[P. L. Combettes, L. Condat, J.-C. Pesquet, and B. C. Vũ, "A forward-backward view of some primal-dual optimization methods in image recovery," IEEE ICIP 2014]

[M. Yan, "A New Primal–Dual Algorithm for Minimizing the Sum of Three Functions with a Linear Operator", *J. Sci. Comput.*, 2018]

Thank you!