Technische Universität Berlin

Lecture 1: Splitting Algorithns

Gabriele Steidl
Applied Mathematics - Imaging Sciences
TU Berlin

Berlin Mathematics Research Center MATH+

Funded under Germany's Excellence Strategy by DFG Deutsche Forschungsgemeinschaft

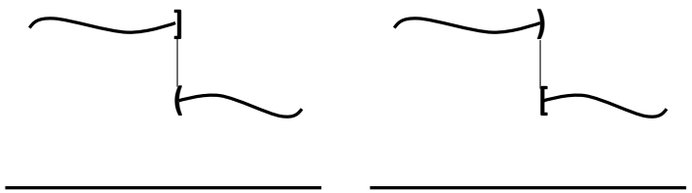
Lecture 1: Splitting Algorithms in Convex Analysis

1. Proximal and Averaged Operators
2. Proximal Point Algorithm (PPA) and Cyclic PPA
3. Forward-Backward Splitting (FBS) and Accelerated FBS
4. Douglas-Rachford Splitting
5. Primal-Dual Algorithms

Plus: Plug-and-Play Algorithm and Proximal Neural Networks

Some notation

- $\Gamma_0(\mathcal{H})$ is a set of proper, convex, and lower semi-continuous (lsc) functions $f : \mathcal{H} \to \mathbb{R} \cup \{+\infty\}$, typically $\mathcal{H} = \mathbb{R}^d$



Not lsc. lsc

- effective domain: $\mathrm{dom} f := \{x \in \mathbb{R}^d : f(x) < +\infty\}$
- ι_S indicator function of $S \subseteq \mathcal{H}$:

$$\iota_S(x) := \begin{cases} 0 & \text{if } x \in S, \\ +\infty & \text{otherwise.} \end{cases}$$

Then $\iota_S \in \Gamma_0(\mathcal{H})$ if S is nonempty, convex, closed. Useful reformulation

$$\operatorname*{argmin}_{x \in \mathcal{H}} \{f(x) + \iota_S(x)\} = \operatorname*{argmin}_{x \in S} f(x)$$

Note that $\inf \emptyset = +\infty$ and $\sup \emptyset = -\infty$, since we want for $s \neq \emptyset$ that $\inf_{x \in S} x < \inf \emptyset$

Proximal Operators

For $f \in \Gamma_0(\mathcal{H})$, the proximal mapping $\text{prox}_{\tau f} : \mathcal{H} \to \mathcal{H}$ is well defined by

$$\text{prox}_{\tau f}(x) := \underset{y \in \mathcal{H}}{\text{argmin}} \left\{ \frac{1}{2\tau} \|y - x\|^2 + f(y) \right\}, \quad \tau > 0$$

and the Moreau-Yoshida envelope $M_{\tau f} : \mathcal{H} \to \mathbb{R}$ by

$$M_{\tau f}(x) := \min_{y \in \mathcal{H}} \left\{ \frac{1}{2\tau} \|y - x\|^2 + f(y) \right\}$$

Remark:

- generalization of projection onto convex sets

$$\text{prox}_{\tau \iota_S}(x) = \underset{y \in \mathcal{H}}{\text{argmin}} \left\{ \frac{1}{2\tau} \|y - x\|_2^2 + \iota_S(y) \right\} = \underset{y \in S}{\text{argmin}} \|y - x\|_2 =: \Pi_S(x)$$

- $\text{prox}_{\tau f}$ acts as denoiser for appropriate f, as e.g. $f = \text{TV}$
- backward scheme for minimizing f

Forward-Backward Operators

Technische Universität Berlin

Gradient flows:

$$x(t_0) = x_0$$
$$\dot{x}(t) = F(x(t)) = -\nabla f(x(t)), \quad t \in [t_0, T]$$

Discretization $t_0 < t_1 < \ldots$, $t_{r+1} = t_r + \tau$

$$x(t_{r+1}) - x(t_r) = \int_{t_r}^{t_{r+1}} F(x(t))\,\mathrm{d}t$$

Apply quadrature rule

$$x(t_{r+1}) \approx x(t_r) - \tau \nabla f(x(t_r)) \quad \text{Euler forward}$$
$$x(t_{r+1}) \approx x(t_r) - \tau \nabla f(x(t_{r+1}) \quad \text{Euler backward}$$
$$x(t_{r+1}) + \tau \nabla f(x(t_{r+1}) \approx x(t_r)$$
$$x(t_{r+1}) \approx x(t_r) - \tau \left(\theta \nabla f(x(t_r)) + (1-\theta) \nabla f(x(t_{r+1}))\right)$$

Crank-Nichelson for $\theta = \frac{1}{2}$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

Example

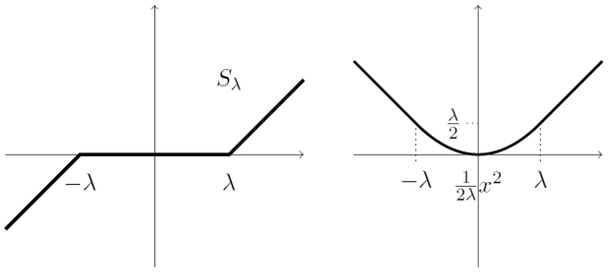
Let $\mathcal{H} = (\mathbb{R}, |\cdot|)$ and $f(x) := |x|$.

$$\text{prox}_{\tau f} = S_\tau(x) := \begin{cases} x - \tau & \text{for } x > \tau, \\ 0 & \text{for } x \in [-\tau, \tau], \\ x + \tau & \text{for } x < -\tau, \end{cases} \qquad \text{(soft shrinkage)}$$

$$M_{\tau|\cdot|}(x) = \begin{cases} x - \frac{\tau}{2} & \text{for } x > \tau, \\ \frac{1}{2\tau}x^2 & \text{for } x \in [-\tau, \tau], \\ -x - \frac{\tau}{2} & \text{for } x < -\tau. \end{cases} \qquad \text{(Huber function)}$$

Hence, $\text{prox}_{\tau f} = \nabla\Phi$, where $\Phi(x) = \frac{x^2}{2} - \tau M_{\tau|\cdot|}(x)$, i.e.,

$$\Phi(x) = \begin{cases} \frac{1}{2}(x - \tau)^2 & \text{for } x > \tau, \\ 0 & \text{for } x \in [-\tau, \tau], \\ \frac{1}{2}(x + \tau)^2 & \text{for } x < -\tau. \end{cases}$$



Properties of the Proximal Operator

i) Variational inequality: $\hat{x} = \operatorname{prox}_{\tau f}(x)$ iff

$$\frac{1}{\tau}\langle x - \hat{x}, y - \hat{x}\rangle + f(\hat{x}) - f(y) \leq 0 \qquad \forall y \in \mathcal{H}$$

ii) Fixed point property: $\hat{x} \in \operatorname{argmin} f$ iff $\hat{x} = \operatorname{prox}_{\tau f}(\hat{x})$

iii) Continuity + convexity of Moreau envelope $M_{\tau f}$:

$$\nabla M_{\tau f}(x) = \frac{1}{\tau}\left(x - \operatorname{prox}_{\tau f}(x)\right)$$

iv) Convex potentials Moreau (1965): $\operatorname{prox} : \mathcal{H} \to \mathcal{H}$ is proximity operator iff it is nonexpansive and $\operatorname{prox} = \nabla\Phi$ for some $\Phi \in \Gamma_0(\mathcal{H})$. One direction:

$$\operatorname{prox}_{\tau f}(x) = x - \tau\nabla M_{\tau f}(x)$$
$$\operatorname{prox}_{\tau f}(x) = \nabla\left(\tfrac{1}{2}\|x\|^2 - \tau M_{\tau f}(x)\right) = \nabla\Phi(x)$$

where $\Phi := \frac{1}{2}\|\cdot\|^2 - \tau M_{\tau f}$ is convex as $\operatorname{prox}_{\tau f}$ is nonexpansive.

v) Moreau decomposition (later):

$$\operatorname{prox}_f(x) + \operatorname{prox}_{f^*}(x) = x, \quad M_f(x) + M_{f^*}(x) = \tfrac{\tau}{2}\|x\|_2^2$$

Example extended

Let $\mathcal{H} = (\mathbb{R}^d, \|\cdot\|_2)$. Then with the componentwise soft shrinkage operator

$$\text{prox}_{\tau\|\cdot\|_1}(x) = \underset{y\in\mathbb{R}^d}{\text{argmin}}\left\{\tfrac{1}{2}\|x-y\|_2^2 + \tau\|y\|_1\right\} = S_\tau(x), \quad \tau > 0$$

Analysis concept in compressed sensing:

$$\text{prox}_{\tau\|T\cdot\|_1}(x) = \underset{y\in\mathbb{R}^d}{\text{argmin}}\left\{\tfrac{1}{2}\|x-y\|_2^2 + \tau\|Ty\|_1\right\}$$

- $T \in \mathbb{R}^{d\times d}$ orthogonal: $\text{prox}_{\tau\|T\cdot\|_1} = T^*\, S_\tau\, T$, see blackbord
- $T \in \mathbb{R}^{n\times d}$ with $n \le d$ and $TT^* = I_n$: $\text{prox}_{\tau\|T\cdot\|_1} = I_d - T^*T + T^*S_\tau T$
- for general T no analytic expression

Proximity Webpage: G. Chierchia, E. Chouzenoux, P. L. Combettes, and J.-C. Pesquet. "The Proximity Operator Repository. User's guide". `http://proximity-operator.net/`

Example extended

Theorem (Neumayer et al. 2020, Combettes 2018)
Let $b \in \mathcal{K}$, $T \in \mathcal{B}(\mathcal{H}, \mathcal{K})$ have closed range and prox : $\mathcal{K} \to \mathcal{K}$ be a proximity operator on $\mathcal{K}$ (for some function $f \in \Gamma_0(\mathcal{K})$). Then, the operator

$$A := T^\dagger \operatorname{prox}(T \cdot + b)$$

is a proximity operator for some $g \in \Gamma_0(\mathcal{H}_T)$, i.e.

$$A = \operatorname*{argmin}_{y \in H_T} \frac{1}{2} \| \cdot - y \|^2_{\mathcal{H}_T} + \tau g(y),$$

where for invertible T we have $\langle x, y \rangle_{\mathcal{H}_T} := \langle Tx, Ty \rangle / \|T\|^2$

Idea of the proof: Use Moreau's result to show that

- A is nonexpansive in $\mathcal{H}_T$ and
- $A = \nabla \Psi$, $\Psi \in \Gamma_0(\mathcal{H}_T)$.
  Candidate: $\Psi = \Phi(T \cdot + b)/\|T\|^2$, where prox $= \nabla \Phi$

Recall

For a Fréchet differentiable function $\Phi\colon \mathcal{H} \to \mathbb{R}$, the gradient $\nabla\Phi(x)$ at $x \in \mathcal{H}$ is defined as the vector satisfying for all $h \in \mathcal{H}$,

$$\langle \nabla\Phi(x), h\rangle = D\Phi(x)h,$$

where $D\Phi\colon \mathcal{H} \to \mathcal{B}(\mathcal{H}, \mathbb{R})$ denotes the Fréchet derivative of Φ, i.e., for all $x, h \in \mathcal{H}$,

$$\Phi(x+h) - \Phi(x) = D\Phi(x)h + o(\|h\|).$$

- The gradient crucially depends on the chosen inner product in $\mathcal{H}$.

Subdifferential

Subdifferential of f at $x_0 \in \mathrm{dom} f$:

$$\partial f(x_0) := \{p \in \mathbb{R}^d : f(x) \geq f(x_0) + \langle p, x - x_0 \rangle \ \forall x \in \mathbb{R}^d\}$$

Fermat's rule: for any $f : \mathbb{R}^d \to \mathbb{R} \cup \{+\infty\}$:
$\hat{x}$ is a global minimizer of f if and only if $0 \in \partial f(\hat{x})$.

Properties of the subdifferential

For $f \in \Gamma_0(\mathbb{R}^d)$ we have

- If f is differentiable at x_0, then $\partial f(x_0) = \{\nabla f(x_0)\}$ and conversely
- $\partial f(x_0)$ is nonempty and bounded iff $x_0 \in \mathrm{int}(\mathrm{dom} f)$
- $\partial(f_1 + f_2) \subseteq \partial f_1 + \partial f_2$
  with equality if $\mathrm{dom} f_1 \cap \mathrm{dom} f_2$ contains a point where one of the functions is continuous

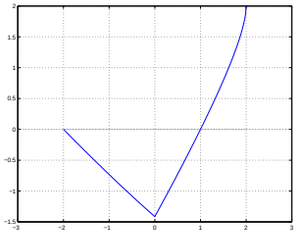
Examples: 1. $f(x) = |x|$

$$\partial f(x) = \begin{cases} 1 & x > 0 \\ -1 & x < 0 \\ [-1, 1] & x = 0 \end{cases}$$

Examples: Subdifferential

2.

$$f(x) := \begin{cases} |x| - (2-x)^{\frac{1}{2}} & \text{if} \quad |x| \leq 2, \\ +\infty & \text{otherwise}, \end{cases}$$



$$\partial f(x) = \begin{cases} \emptyset & \text{if} \quad x = 2, \\ \{1 + \frac{1}{2}(2-x)^{-\frac{1}{2}}\} & \text{if} \quad 0 < x < 2, \\ \left[\sqrt{2}/4 - 1, \sqrt{2}/4 + 1\right] & \text{if} \quad x = 0, \\ \{-1 + \frac{1}{2}(2-x)^{-\frac{1}{2}}\} & \text{if} \quad -2 < x < 0, \\ (-\infty, -3/4] & \text{if} \quad x = -2, \\ \emptyset & \text{otherwise}. \end{cases}$$

By Fermat's rule:

$$\hat{x} = \text{prox}_{\tau f}(x) = \underset{y \in \mathbb{R}^N}{\text{argmin}} \left\{ \frac{1}{2}\|y - x\|_2^2 + \tau f(y) \right\}$$

$$\Leftrightarrow \quad 0 \in \hat{x} - x + \tau \partial f(\hat{x})$$

$$x \in \hat{x} + \tau \partial f(\hat{x}) = (I + \tau \partial f)(\hat{x})$$

$$\hat{x} = (I + \tau \partial f)^{-1}(x)$$

$$\text{prox}_{\tau f}(x) = \underbrace{(I + \tau \partial f)^{-1}}_{\text{resolvent of } \tau \partial f}(x)$$

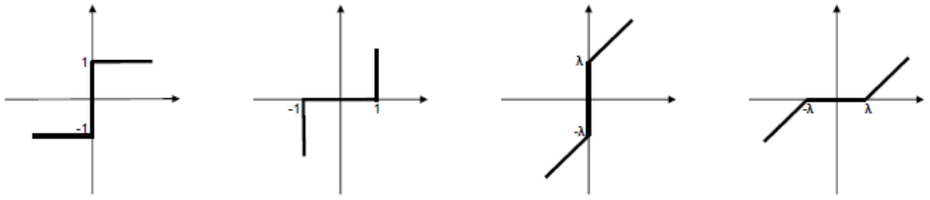


Fig. 5.1 Left to right: $F, F^{-1}, I + \lambda F$ and $(I + \lambda F)^{-1}$.

$$F := \partial f$$

Averaged Operators

An operator $T : \mathcal{H} \to \mathcal{H}$ is called

- Lipschitz continuous if

$$\|Tx - Ty\| \leq L\|x - y\| \quad \forall x, y \in \mathcal{H}$$

- nonexpansive if $L \leq 1$ and contractive if $L < 1$
- Banach's fixed point theorem: If T is a contraction then it has a unique fixed point which is reached starting from any $x^{(0)} \in \mathcal{H}$ by the sequence

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

- averaged iff there exists a nonexpansive mapping R and $\alpha \in (0, 1)$ such that

$$T = \alpha I + (1 - \alpha)R.$$

- Clearly, an averaged operator T is nonexpansive and

$$\mathrm{Fix}(R) = \mathrm{Fix}(T).$$

Examples

1. Reflection operator $\mathbb{R}^2$ which in nonexpansive

$$R := \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

but produces except for $x^{(0)} = (x_1, 0)^\mathsf{T}$ the non-convergent series

$$x^{(1)} = \begin{pmatrix} x_1^{(0)} \\ -x_2^{(0)} \end{pmatrix}, \; x^{(2)} = \begin{pmatrix} x_1^{(0)} \\ x_2^{(0)} \end{pmatrix}, x^{(3)} = \begin{pmatrix} x_1^{(0)} \\ -x_2^{(0)} \end{pmatrix}, \dots$$

A 'better' operator is obtained by averaging

$$T := \alpha I + (1-\alpha)R = \begin{pmatrix} 1 & 0 \\ 0 & 2\alpha - 1 \end{pmatrix}, \quad \alpha \in (0,1)$$

Here we get

$$x^{(r)} \to \begin{pmatrix} x_1^{(0)} \\ 0 \end{pmatrix} \quad \text{as } r \to \infty$$

2. $\text{prox}_{\tau f}$ is an averaged operator with $\alpha = \frac{1}{2}$, a so-called firmly nonexpansive operator, i.e.,

$$\text{prox}_{\tau f} = \frac{1}{2}(I + R)$$

Properties of averaged operators

Theorem (Concatenation of Averaged Operators)
If $T_k : \mathcal{H} \to \mathcal{H}$ are α_k-averaged, $k = 1, \ldots, K$, then $T = T_N \circ \ldots \circ T_0$ is α-averaged with

$$\alpha = \left(1 + \left(\sum \frac{\alpha_k}{1 - \alpha_k}\right)^{-1}\right)^{-1} \leq \frac{K}{(K-1) + (\max \alpha_k)^{-1}}.$$

Theorem (Convergence of Averaged Operators)
Let $T : \mathcal{H} \to \mathcal{H}$ be an averaged operator such that $\mathrm{Fix}(T) \neq \emptyset$. Then, for every $x^{(0)} \in \mathcal{H}$, the sequence $(T^r x^{(0)})_{r \in \mathbb{N}}$ converges weakly to a fixed point of T.

Krasnoselski-Mann iterations (nonexpansive R) converge

$$x^{(r+1)} = \big(\alpha I + (1 - \alpha) R\big)\big(x^{(r)}\big), \qquad \alpha \in (0, 1)$$

- Method cannot be generalized to Banach spaces without additional constraints. Fortunately, the Krasnoselskii–Mann method works on finite dimensional normed spaces.
- Some Results can be extended to uniformly convex spaces. Berlellima, Steidl 2020

Proximal Point Algorithm (PPA)

Let $f \in \Gamma_0(\mathbb{R}^d)$ have a minimizer $\hat{x} \in \operatorname{argmin}_x f(x)$. Then we know already that $\hat{x}$ fulfills the fixed point equation

$$\hat{x} = \operatorname{prox}_{\tau f}(\hat{x})$$

PPA: Initialization $x^{(0)} \in \mathcal{H}$

$$x^{(r+1)} = \operatorname{prox}_{\tau f}(x^{(r)}) = \operatorname*{argmin}_{x \in \mathcal{H}} \left\{ \frac{1}{2\tau} \|x^{(r)} - x\|_2^2 + f(x) \right\}$$

- $(x^{(r)})_r$ converges to fixed point of $\operatorname{prox}_{\tau f}$ = minimizer $\hat{x}$ of f
- $f(x^{(r)}) - f(\hat{x}) \leq \frac{1}{2\tau r} \|x^{(0)} - \hat{x}\|^2$

Generalizations of Proximal Point Algorithm (PPA)

PPA can be generalized for

$$f := \sum_{k=1}^{n} f_k, \quad f_k \in \Gamma_0(\mathcal{H})$$

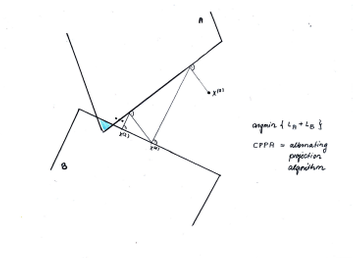
inexact PPA, parallel PPA, by (stochastic) cyclic PPA:

$$x^{(r+1)} = \text{prox}_{\tau_r f_n} \circ \ldots \circ \text{prox}_{\tau_r f_1}(x^{(r)})$$

If $(\tau_r)_r \in \ell_2 \setminus \ell_1$, then $(x^{(r)})_r$ converges to the minimizer of f.
If $\tau_r = \tau$, then $(x^{(r)})_r$ converges to $M_{\tau f_1} + f_2$ $(n = 2)$
and for arbitrary n to ??

- Example: CPPA = alternating projection algorithm

$$\underset{x}{\text{argmin}} \sum_{k=1}^{n} \iota_{C_k}(x) \quad C_k \text{ closed, convex, nonempty intersection}$$



Lecture 1: Splitting Algorithms in Convex Analysis

Operator splitting:

- Origin: linear algebra, PDEs (e.g. Douglas-Rachford 1956): linear, single-valued operators:

$$0 = \nabla f(x) + \nabla g(x) = Ax + Bx$$

- Generalization to inclusion problems by Lions/Mercier 1979: nonlinear, set-valued, Eckstein 1989:

$$0 \in \partial f(x) + \partial g(x) = A(x) + B(x)$$

- Image processing applications by Figueiredo et al. 2007, Combettes/Wajs,Pesquet 2005, 2007

Forward-Backward Splitting

Variational model for inverse problem:

$$\mathcal{J}(x) = \underbrace{\mathcal{D}(F(x), y)}_{\text{data term}} + \lambda \underbrace{\mathcal{R}(x)}_{\text{regularizer}}, \qquad \lambda > 0$$

Task: $h \in \Gamma_0(\mathcal{H})$, $g : \mathcal{H} \to \mathbb{R}$ convex, diff. and $\mathrm{Lip}\nabla g = L < \infty$

$$\hat{x} \in \operatorname*{argmin}_{x \in \mathcal{H}} \{\underbrace{g(x) + h(x)}_{f(x)}\}.$$

Fermat's rule

$$0 \in \nabla g(\hat{x}) + \partial h(\hat{x})$$

$$\hat{x} - \tau \nabla g(\hat{x}) \in \hat{x} + \tau \partial h(\hat{x}) = (I + \tau \partial h)(\hat{x})$$

$$\hat{x} = \mathrm{prox}_{\tau h}\left(\hat{x} - \tau \nabla g(\hat{x})\right)$$

Forward-Backward Splitting (FBS): (also called proximal gradient algorithm)

$$x^{(r+1)} = \mathrm{prox}_{\tau h}\left(x^{(r)} - \tau \nabla g(x^{(r)})\right)$$

Gradient Descent Re-Projection Algorithm for $h = \iota_S$:

$$x^{(r+1)} = \Pi_S\left(x^{(r)} - \tau \nabla g(x^{(r)})\right)$$

Convergence

Theorem: Let $\tau \in (0, \frac{2}{L})$. Then $\{x^{(r)}\}_r$ converges linearly to a minimizer of $g + h$ if it exists.

$$f\left(x^{(r)}\right) - f(\hat{x}) = \mathcal{O}\left(1/r\right).$$

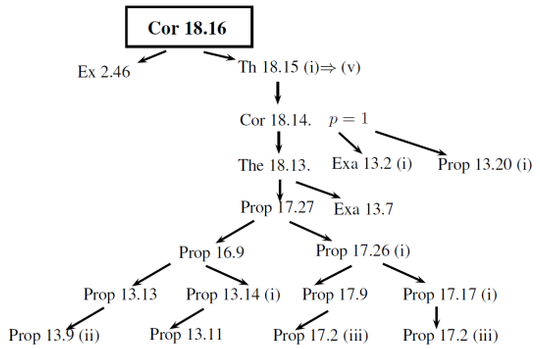
Idea of the Proof:

$$x^{(r+1)} = \underbrace{\text{prox}_{\tau h}}_{\text{averaged}} \circ (I - \tau \nabla g)\left(x^{(r)}\right)$$

By the Baillon-Haddad Theorem $\frac{1}{L}\nabla g$ is firmly nonexpansive. Thus

$$\frac{1}{L}\nabla g = \frac{1}{2}(I + R) \quad \Leftrightarrow \quad I - \tau \nabla g = (1 - \tfrac{\tau L}{2})I + \tfrac{\tau L}{2}(-R)$$

which is averaged if $\frac{\tau L}{2} \in (0, 1)$.



Example: Iterative Soft-Thresholding Algorithm (ISTA)

Consider

$$\underset{x\in\mathbb{R}^d}{\operatorname{argmin}} \{ \underbrace{\frac{1}{2}\|Kx-b\|_2^2}_{g} + \underbrace{\lambda\|x\|_1}_{h} \}$$

Then

$$\nabla g(x) = K^\mathsf{T}(Kx-b)$$

$$\operatorname{prox}_{\lambda\tau\|\cdot\|_1} = S_{\tau\lambda} \quad \text{soft shrinkage}$$

FBS = ISTA (Iterative Soft-Thresholding Algorithm):

$$\begin{aligned} x^{(r+1)} &= \operatorname{prox}_{\tau h}\left(x^{(r)} - \tau\nabla g(x^{(r)})\right) \\ &= \operatorname{prox}_{\tau\tau\|\cdot\|_1}\left(x^{(r)} - \tau K^\mathsf{T}(Kx^{(r)}-b)\right) \\ &= S_{\tau\tau}\left(x^{(r)} - \tau K^\mathsf{T}(Kx^{(r)}-b)\right). \end{aligned}$$

- ISTA: Daubechies, Defrise, De Mol 2004; Combettes, Wajs 2005

Accelerated FBS

By a simple extrapolation trick

$$y^{(r)} = x^{(r)} + \tau_r \left(x^{(r)} - x^{(r-1)} \right), \quad \tau_r = \frac{r-1}{r+2}$$
$$x^{(r+1)} = \mathrm{prox}_{\tau h} \left(y^{(r)} - \tau \nabla g(y^{(r)}) \right)$$

we can achieve best possible rate for $\tau \in (0, \frac{1}{L})$, namely

$$f(x^{(r)}) - f(\hat{x}) = \mathcal{O}\left(1/r^2\right)$$

Equivalent form:

$$y^{(r)} = (1-\theta_r)x^{(r)} + \theta_r z^{(r)}, \quad \theta_r = \frac{2}{r+2}$$
$$x^{(r+1)} = \mathrm{prox}_{\tau h} \left(y^{(r)} - \tau \nabla g(y^{(r)}) \right)$$
$$z^{(r+1)} = x^{(r+1)} + \frac{1}{\theta_r}(x^{(r+1)} - x^{(r)})$$

- Several fashions: Nesterov algorithm, FISTA

Proof: 1. **Progress in one step**: By the L-Lipschitz gradient of g and since $\tau < \frac{1}{L}$

$$g(x^{(r+1)}) \leq g(y^{(r)}) + \langle \nabla g(y^{(r)}), x^{(r+1)} - y^{(r)} \rangle + \frac{1}{2\tau} \|x^{(r+1)} - y^{(r)}\|_2^2$$

By the variational characterization of the proximal operator, we get for all $u \in \mathbb{R}^d$,

$$\begin{aligned} h(x^{(r+1)}) &\leq h(u) + \frac{1}{\tau} \langle y^{(r)} - \tau \nabla g(y^{(r)}) - x^{(r+1)}, x^{(r+1)} - u \rangle \\ &\leq h(u) - \langle \nabla g(y^{(r)}), x^{(r+1)} - u \rangle + \frac{1}{\tau} \langle y^{(r)} - x^{(r+1)}, x^{(r+1)} - u \rangle. \end{aligned}$$

Adding these inequalities and using the convexity of g yields

$$\begin{aligned} f(x^{(r+1)}) &\leq f(u) \underbrace{-g(u) + g(y^{(r)}) + \langle \nabla g(y^{(r)}), u - y^{(r)} \rangle}_{\leq 0} \\ &\quad + \frac{1}{2\tau} \|x^{(r+1)} - y^{(r)}\|_2^2 + \frac{1}{\tau} \langle y^{(r)} - x^{(r+1)}, x^{(r+1)} - u \rangle \\ &\leq f(u) + \frac{1}{2\tau} \|x^{(r+1)} - y^{(r)}\|_2^2 + \frac{1}{\tau} \langle y^{(r)} - x^{(r+1)}, x^{(r+1)} - u \rangle. \end{aligned}$$

Combining these inequalities for $u := \hat{x}$ and $u := x^{(r)}$ with $\theta_r \in [0,1]$ gives

$$\begin{aligned}
&\theta_r \left(f(x^{(r+1)}) - f(\hat{x})\right) + (1-\theta_r)\left(f(x^{(r+1)}) - f(x^{(r)})\right) \\
&= f(x^{(r+1)}) - f(\hat{x}) + (1-\theta_r)\left(f(\hat{x}) - f(x^{(r)})\right) \\
&\leq \frac{1}{2\tau}\|x^{(r+1)} - y^{(r)}\|_2^2 + \frac{1}{\tau}\langle y^{(r)} - x^{(r+1)}, x^{(r+1)} - \theta_r\hat{x} - (1-\theta_r)x^{(r)}\rangle \\
&= \frac{1}{2\tau}\left(\|y^{(r)} - \theta_r\hat{x} - (1-\theta_r)x^{(r)}\|_2^2 - \|x^{(r+1)} - \theta_r\hat{x} - (1-\theta_r)x^{(r)}\|_2^2\right) \\
&= \frac{\theta_r^2}{2\tau}\left(\|z^{(r)} - \hat{x}\|_2^2 - \|z^{(r+1)} - \hat{x}\|_2^2\right).
\end{aligned}$$

Thus, we obtain for a single step

$$\frac{\tau}{\theta_r^2}\left(f(x^{(r+1)}) - f(\hat{x})\right) + \frac{1}{2}\|z^{(r+1)} - \hat{x}\|_2^2 \leq \frac{\tau(1-\theta_r)}{\theta_r^2}\left(f(x^{(r)}) - f(\hat{x})\right) + \frac{1}{2}\|z^{(r)} - \hat{x}\|_2^2.$$

2. **Recursively application** and regarding that $\frac{(1-\theta_r)}{\theta_r^2} \leq \frac{1}{\theta_{r-1}^2}$ we obtain

$$\frac{\tau}{\theta_r^2}\left(f(x^{(r+1)}) - f(\hat{x})\right) \leq \frac{\tau(1-\theta_0)}{\theta_0^2}\left(f(x^{(0)}) - f(\hat{x})\right) + \frac{1}{2}\|z^{(0)} - \hat{x}\|_2^2 = \frac{1}{2}\|x^{(0)} - \hat{x}\|_2^2,$$

$$f(x^{(r+1)}) - f(\hat{x}) \leq \frac{2}{\tau(r+2)^2}\|x^{(0)} - \hat{x}\|_2^2. \qquad \square$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

Plug-and-Play FBS

Idea: Replace the proximity operator wrt the regularizer by a more powerful denoiser

- MMSE by Lebrun/Morel (Denoising cuisine) 2013
- BM3D by Dabov et al. (BM3D) 2008
- Neural Network

FBS and FBS-PnP

Initialization: $x^{(0)} \in \mathbb{R}^m$, $\tau \in (0, \frac{2}{L})$

Iterations: For $r = 0, 1, \ldots$

$$y^{(r+1)} = x^{(r)} - \tau \nabla f(x^{(r)})$$
$$x^{(r+1)} = \text{prox}_{\tau g}(y^{(r+1)})$$

PnP Step: $x^{(r+1)} = D(y^{(r+1)})$

Theorem: Let $\tau \in (0, \frac{2}{L})$ and D be averaged. Then $\{x^{(r)}\}_r$ converges to a minimizer of $g + h$ if it exists.

Refs: see recent results of Kamilov et al.

Proximal Neural Networks (PNNs)

Let

$$B(\cdot; T, b, \alpha) := T^* \underbrace{\sigma_\alpha}_{\text{prox}}(T \cdot + b),$$

with $T/T^* \in \mathrm{St}(n,d) := \{T \in \mathbb{R}^{n,d} : T^*T = I\}$ and stable activation function σ_α (Refs: Combettes/Pesquet 2020)

PNN with K layers:

$$\Phi(\cdot; \theta) = B_K(\cdot; T_K, b_K, \alpha_K) \circ \cdots \circ B_1(\cdot; T_1, b_1, \alpha_1),$$

where $\theta = ((T_k)_{k=1}^K, (b_k)_{k=1}^K, (\alpha_k)_{k=1}^K)$.

Properties:

- Φ is $K/(K+1)$-averaged, since concatenation of K $\frac{1}{2}$-averaged operators
- (scaled) PNNs show a comparable performance as usual convolutional neural networks for denoising
- robustness against adversarial attacks
- No lunch for free: To ensure that the weight matrices are from the Stiefel manifold.
  $\Rightarrow$ optimization on manifold
- construction of convolutional PNNs

Refs: Neumayer, Hertrich et al. 2020, 2021

Stable Activation Functions

- A nonexpansive, monotone increasing function $\sigma : \mathbb{R} \to \mathbb{R}$ with $\sigma(0) = 0$ is called a stable activation function.
- σ is stable activation function iff $\sigma = \text{prox}_g$ for some $g \in \Gamma_0(\mathbb{R})$ with minimizer 0.

Reason:
$\sigma(x) = \Phi'(x)$, where $\Phi(x) := \int_0^x \sigma(t)\,dt$ is continuous, convex function
Moreau $\to \sigma = \text{prox}_g$.
$\sigma(0) = \text{prox}_g(0) = 0$ Thus, 0 is a fixed point of $\text{prox}_g \Leftrightarrow \operatorname{argmin} g$

Name	activation function $\sigma(x)$	$f(x)$ with $\sigma = \text{prox}_f$
Linear activation	x	0
Saturated linear activation (SaLU)	$\begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \le x \le 1 \\ -1 & \text{if } x < -1 \end{cases}$	$\iota_{[-1,1]} = \begin{cases} 0 & \text{if } x \in [-1,1] \\ \infty & \text{if } x \notin [-1,1] \end{cases}$
Soft Thresholding	$\begin{cases} x-\lambda & \text{if } x > \lambda \\ 0 & \text{if } x \in [-\lambda,\lambda] \\ x-\lambda & \text{if } x < -\lambda \end{cases}$	$\lambda\lvert x\rvert$
Saturated linear activation (SaLU)	$\begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \le x \le 1 \\ -1 & \text{if } x < -1 \end{cases}$	$\iota_{[-1,1]} = \begin{cases} 0 & \text{if } x \in [-1,1] \\ \infty & \text{if } x \notin [-1,1] \end{cases}$
Rectified linear unit (ReLU)	$\begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \le 0 \end{cases}$	$\iota_{(0,\infty)} = \begin{cases} 0 & \text{if } x \in [0,\infty) \\ \infty & \text{if } x \notin [0,\infty) \end{cases}$
Parametric rectified linear unit (PReLU)	$\begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \le 0 \end{cases}, \alpha \in (0,1]$	$\begin{cases} 0 & \text{if } x > 0 \\ (\frac{1}{\alpha}-1)\frac{x^2}{2} & \text{if } x \le 0 \end{cases}$
Bent identity activation	$\frac{x+\sqrt{x^2+1}}{2}$	$\begin{cases} x/2 - \ln(x+\frac{1}{2})/4 & \text{if } x > -\frac{1}{2} \\ \infty & \text{if } x \le -\frac{1}{2} \end{cases}$
Inverse square root linear unit	$\begin{cases} x & \text{if } x \ge 0 \\ \frac{x}{\sqrt{x^2+1}} & \text{if } x < 0 \end{cases}$	$\begin{cases} 0 & \text{if } x \ge 0 \\ 1 - x^2/2 - \sqrt{1-x^2} & \text{if } -1 \le x < 0 \\ \infty & \text{if } x < -1 \end{cases}$
Inverse square root unit	$\frac{x}{\sqrt{x^2+1}}$	$\begin{cases} -x^2/2 - \sqrt{1-x^2} & \text{if } \lvert x\rvert \le 1 \\ \infty & \text{if } \lvert x\rvert > 1 \end{cases}$
Arctangent activation	$\frac{2}{\pi}\arctan(x)$	$\begin{cases} -\frac{2}{\pi}\ln(\cos(\frac{\pi x}{2})) - \frac{x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \ge 1 \end{cases}$
Hyperbolic tangent activation	$\tanh(x)$	$\begin{cases} x\operatorname{artanh}(x) + \frac{\ln(1-x^2)-x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \ge 1 \end{cases}$
Elliot activation	$\frac{x}{\lvert x\rvert+1}$	$\begin{cases} -\lvert x\rvert - \ln(1-\lvert x\rvert) - \frac{x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \ge 1 \end{cases}$
Inverse hyperbolic sine	$\operatorname{arcsinh}(x)$	$\cosh(x) - \frac{\lvert x\rvert^2}{2}$
Logarithmic activation	$\operatorname{sgn}(x)\ln(\lvert x\rvert+1)$	$\exp(\lvert x\rvert) - \lvert x\rvert - 1 - \frac{\lvert x\rvert^2}{2}$

Network Architecture

Ingredients for a powerful PNN denoiser:

- Learn the noise instead of the noise-free images ("Residual Learning")
- Learn PNNs with one additional fixed scaling parameter $\gamma \geq 1$.
  $\Rightarrow$ Upper bound for the Lipschitz constant of the PNN
- Start with m_2 copies of the input images.

Summarized, the denoiser has the form

$$\mathcal{D}(x;\theta) = x - \gamma A^{\mathsf{T}} \Phi(Ax;\theta), \quad A = \frac{1}{\sqrt{m_2}} \begin{pmatrix} I_m \\ \vdots \\ I_m \end{pmatrix},$$

where Φ is a PNN.

Parameters: 8 layers, $m_1 = 64$, $m_2 = 128$.

PnP with PNN Denoiser

Recall that we used for denoising the mapping $I_m - \gamma A^\top \Phi(A\cdot; u)$.
$\Rightarrow \Psi = A^\top \Phi(A\cdot; u)$ is averaged.

Lemma (Hertrich et al. 2021)

Let $x^* \in \mathbb{R}^m$ be fixed. Further, let $\Psi\colon \mathbb{R}^m \to \mathbb{R}^m$ be an t-averaged operator with $t \in [\frac{1}{2}, 1]$. For a scaling factor $0 < \gamma < 2$, the mapping

$$\mathcal{D}(x) = \left(1 - \tfrac{1}{1-\gamma+2t\gamma}\right)x^* + \tfrac{1}{1-\gamma+2t\gamma}(I_m - \gamma\Psi(x))$$

is $\tilde{t}$-averaged with $\tilde{t} = \frac{t\gamma}{1-\gamma+2t\gamma}$.

- Numerically the averaging parameter t for denoising-cPNNs is close to $\frac{1}{2}$.
- For $t = \frac{1}{2}$, it we have in the lemma $\mathcal{D}(x) = I_m - \gamma\Psi(x)$.
- If $t > \frac{1}{2}$, we need some oracle.

Numerical Results

- training on 400 training images of the BSDS500 data set, Gaussian noise level $25/255 \approx 0.098$
- test on the BSD68 data set.
- Gaussian noise of noise level $25/255 \approx 0.098$.
- Denoises also images with other noise level

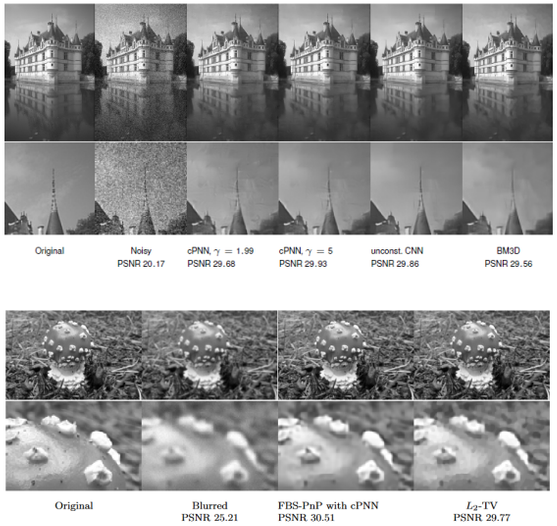


Figure 3.: Deblurring results with blur factor $\tau = 1.5$ and Gaussian noise with $\sigma = 0.01$ using FBS-PnP with a cPNN based denoiser and L_2-TV.

Method	$\sigma = 0.075$,	$\sigma = 0.1$,	$\sigma = 0.125$,	$\sigma = 0.15$
Noisy images	22.50	20.00	18.06	16.48
FBS-PnP with cPNN, $\gamma = 1.99$	30.12	28.80	27.82	27.06
Variational network	30.05	28.72	27.72	26.95
BM3D	29.88	28.50	27.50	26.73

Ref.: Effland, Kobler, Kunisch, Pock. Variational networks: an optimal control approach to early stopping variational methods for image restoration, 2020.

Lecture 1: Splitting Algorithms in Convex Analysis

Douglas-Rachford Algorithm

Let $g, h \in \Gamma_0(\mathbb{R}^d)$ and g or h be continuous at a point in $\operatorname{dom} g \cap \operatorname{dom} h$.

Task:

$$\hat{x} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{g(x) + h(x)\}.$$

Fermat's rule: $0 \in \tau \partial g(\hat{x}) + \tau \partial h(\hat{x}), \quad \tau > 0.$ With

$$\begin{aligned} \hat{x} &= \mathsf{prox}_{\tau g}(\hat{t}) \quad \Leftrightarrow \quad \hat{t} \in \hat{x} + \tau \partial g(\hat{x}) \\ & \qquad\qquad\quad \hat{t} - \hat{x} \in \tau \partial g(\hat{x}) \\ 0 &\in \hat{t} - \hat{x} + \tau \partial h(\hat{x}) \\ 2\hat{x} - \hat{t} &\in \hat{x} + \tau \partial h(\hat{x}) \\ \hat{x} &= \mathsf{prox}_{\tau h}(2\hat{x} - \hat{t}) \end{aligned}$$

Define

$$T := \tfrac{1}{2}\Big(\underbrace{(2\mathsf{prox}_{\tau h} - I)}_{\mathcal{R}_{\tau h}} \circ \underbrace{(2\mathsf{prox}_{\tau g} - I)}_{\mathcal{R}_{\tau g}} + I \Big)$$

with reflections or Cayley operator Then

$$T(\hat{t}) = \tfrac{1}{2}\left((2\mathsf{prox}_{\tau h} - I)\left(2\hat{x} - \hat{t}\right) + \hat{t}\right) = \tfrac{1}{2}\left(2\hat{x} - (2\hat{x} - \hat{t}) + \hat{t}\right) = \hat{t}$$

Technische Universität Berlin

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

Douglas-Rachford Algorithm

$$T := \tfrac{1}{2}\Big(\underbrace{\big(2\mathsf{prox}_{\tau h} - I\big)}_{\mathcal{R}_{\tau h}} \circ \underbrace{\big(2\mathsf{prox}_{\tau g} - I\big)}_{\mathcal{R}_{\tau g}} + I \Big)$$

$$= \mathsf{prox}_{\tau h}\left(\mathcal{R}_{\tau g}(t^{(r)})\right) - \frac{1}{2}\mathcal{R}_{\tau g}(t^{(r)}) + \frac{1}{2}(t^{(r)})$$

DRA: $\tau > 0$

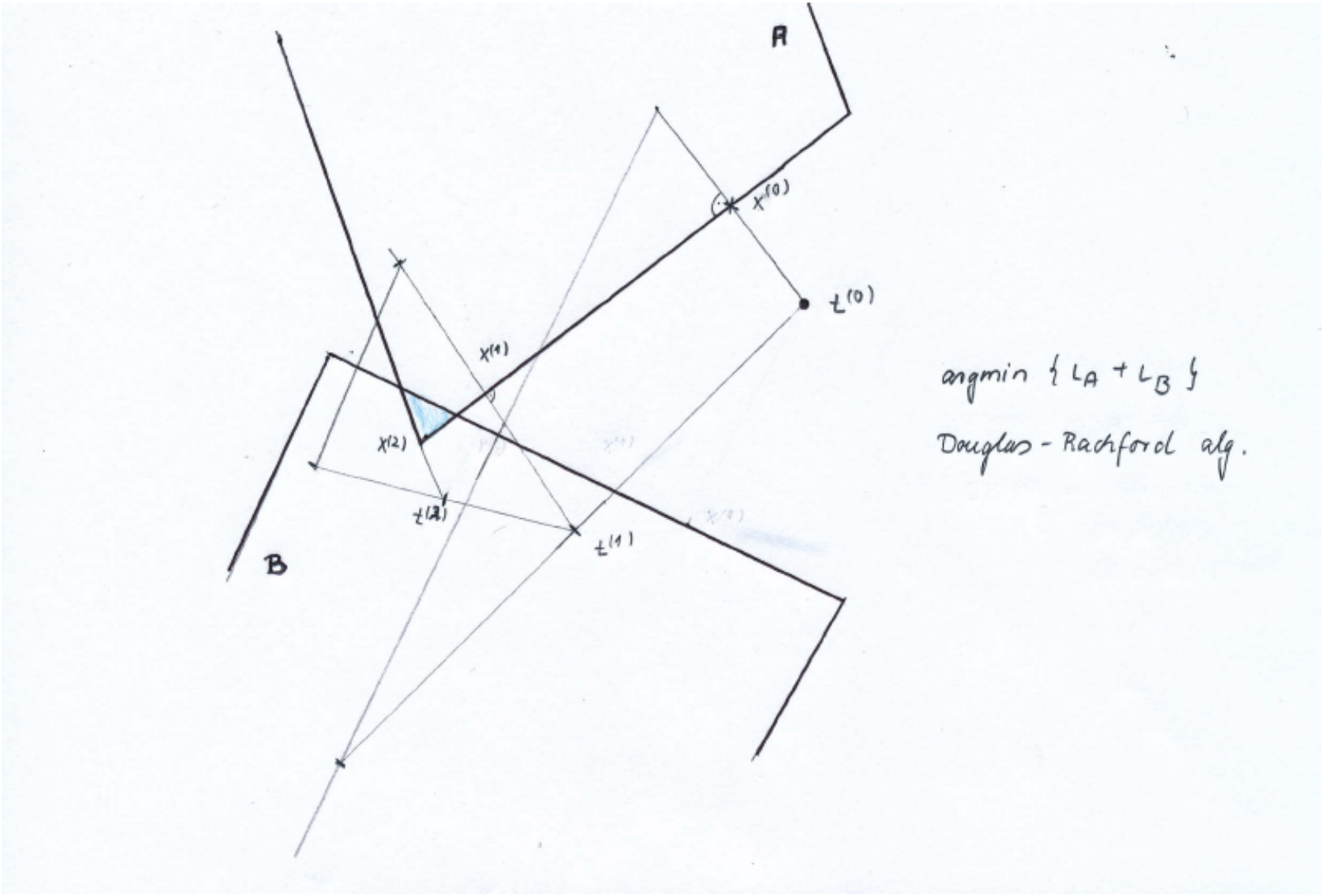
$t^{(0)} \in \mathbb{R}^d$
$x^{(0)} := \mathsf{prox}_{\tau g}(t^{(0)})$

Iterations: For $r = 0, 1, \ldots$

$$\begin{aligned} t^{(r+1)} &= \mathsf{prox}_{\tau h}(2x^{(r)} - t^{(r)}) + t^{(r)} - x^{(r)} \\ &= \mathsf{prox}_{\tau h}\left(\mathcal{R}_{\tau g}(t^{(r)})\right) - \tfrac{1}{2}\mathcal{R}_{\tau g}(t^{(r)}) + \tfrac{1}{2}(t^{(r)}) \end{aligned}$$

$$x^{(r+1)} = \mathsf{prox}_{\tau g}(t^{(r+1)}).$$

Douglas-Rachford Algorithm



Convergence

Theorem: Let $g, h \in \Gamma_0(\mathbb{R}^d)$ where one of the functions is continuous at a point in $\operatorname{dom} g \cap \operatorname{dom} h$. Assume that a solution of $\operatorname{argmin}_{x \in \mathbb{R}^d}\{g(x) + h(x)\}$ exists. Then, for any initial $t^{(0)} \in \mathbb{R}^d$ and any $\tau > 0$, the DRS sequence $(t^{(r)})_r$ converges to a fixed point $\hat{t}$ of T and $(x^{(r)})_r$ to a solution of the minimization problem.

Idea of the Proof: We have

$$T = \tfrac{1}{2}\left((2\mathrm{prox}_{\tau h} - I) \circ \left(2\mathrm{prox}_{\tau g} - I\right) + I\right) \qquad = \tfrac{1}{2}\left(\mathcal{R}_h \circ \mathcal{R}_g + I\right)$$

with the reflection operators

$$\mathcal{R}_h := 2\mathrm{prox}_{\tau h} - I, \quad \mathcal{R}_g := 2\mathrm{prox}_{\tau g} - I$$

The reflections are nonexpansive since the prox operator is firmly nonexpansive, i.e., with some nonexpansive R

$$\mathrm{prox}_{\tau g} = \tfrac{1}{2}(I + R) \quad \Leftrightarrow \quad R = 2\mathrm{prox}_{\tau g} - I.$$

We are done since the DRS steps are Krasnoselski-Mann iterations. □

Extensions of the DRS

Multiple Splitting:

$$\hat{x} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \sum_{k=1}^{n} f_k(x)$$

Setting

$$F(\mathbf{x}) := \sum_{k=1}^{n} f_k(x_k), \quad \mathsf{D} := \left\{\mathbf{x} = (x_1, \ldots, x_n)^{\mathrm{T}} \colon x_1 = \ldots = x_n \in \mathbb{R}^d\right\} \subset \mathbb{R}^{nd}$$

we obtain the minimization problem

$$\operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^{nd}} \left\{ F(\mathbf{x}) + \iota_{\mathsf{D}}(\mathbf{x}) \right\}$$

Parallel DR Algorithm:

$$\mathbf{t}^{(r+1)} := \left((1-\alpha_r)\,\mathrm{Id} + \alpha_r \underbrace{\mathcal{R}_{\tau F}\mathcal{R}_{\tau \iota_{\mathsf{D}}}}_{\text{reflections}}\right)(\mathbf{t}^{(r)}) \qquad \text{(Krasnoselski-Mann iteration)}$$

$$\hat{x} := \mathrm{prox}_{\tau \iota_D}(\hat{t}) = \frac{1}{n}\sum_{k=1}^{n} \hat{t}_k \qquad \text{(final step: mean computation)}$$

Advantage: Separate computation of the proximal mappings

$$\mathrm{prox}_{\tau F}(\mathbf{x}) = \sum_{k=1}^{n} \mathrm{prox}_{\tau f_k}(x_k)$$

Summary so far

Iterative scheme

➡ **Minimization problem :**

$$\hat{\mathrm{x}} \in \operatorname*{Argmin}_{\mathrm{x}} f(\mathrm{x}) + g(\mathrm{x})$$

➡ **Design of a recursive sequence of the form**

$$(\forall k \in \mathbb{N}) \qquad \mathrm{x}^{[k+1]} = \boldsymbol{\Phi}\mathrm{x}^{[k]},$$

Gradient descent	$\boldsymbol{\Phi} = \mathrm{Id} - \tau(\nabla f + \nabla g)$
Proximal point algorithm	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau(f+g)}$
Forward-Backward	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau g}(\mathrm{Id} - \tau\nabla f)$
Peaceman-Rachford	$\boldsymbol{\Phi} = (2\mathrm{prox}_{\tau g} - \mathrm{Id}) \circ (2\mathrm{prox}_{\tau f} - \mathrm{Id})$
Douglas-Rachford	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau g}(2\mathrm{prox}_{\tau f} - \mathrm{Id}) + \mathrm{Id} - \mathrm{prox}_{\tau f}$

Recent work on (linear) convergence rates $\|x^{(r)} - \hat{x}\| \leq c^r x^{(r)} - \hat{x}\|$ see Briceno-Arias, Pustelnik 2021

Lecture 1: Splitting Algorithms in Convex Analysis

1. Proximal and Averaged Operators
2. Proximal Point Algorithm and Cyclic PPA
3. Forward-Backward Splitting and Accelerated FBS
4. Douglas-Rachford Splitting
5. Primal-Dual Algorithms

Conjugate Function

(Fenchel) conjugate function of f:

$$f^*(p) := \sup_{x \in \mathcal{H}} \{\langle p, x \rangle - f(x)\}$$

Properties of conjugate functions

- f^* is always lsc and convex (pointwise supremum of affine functions); if f is proper and convex, then f^* is proper
- $f^{**} = f$ iff $f \in \Gamma_0(\mathcal{H})$
- $f^* = f$ iff $f(x) = \frac{1}{2}\|x\|_2^2$
- $(f_1 + f_2)^* = f_1^* \square f_2^*$
  if $\operatorname{dom} f_1 \cap \operatorname{dom} f_2$ contains a point where one of the functions is continuous. Here we have the infimal convolution

$$(f_1 \square f_2)(x) := \inf_{x=u+v} \{f_1(u) + f_2(v)\}$$

- $\varphi(x) = f(Ax + b) \quad \Rightarrow \quad \varphi^*(p) = f^*\left((A^\mathsf{T})^{-1}p\right) - \langle (A^*)^{-1}p, b \rangle$

Conjugate functions: Examples

1. $f := \|\cdot\|$

$$f^*(p) = \iota_{B_{\|\cdot\|^*}(1)}(p)$$

In particular for $f = \|x\|_1$:

$$f^*(p) = \{p \in \mathbb{R}^d : \|p\|_\infty \leq 1\}$$

Then

$$\operatorname{prox}_{f^*}(x) = \Pi_{[-1,1]}(x)$$

Moreau decomposition:

$$\operatorname{prox}_f(x) = x - \operatorname{prox}_{f^*}(x) = x - \Pi_{[-1,1]}(x) = S_1(x)$$

Technische Universität Berlin

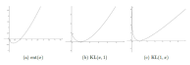
Conjugate Function: Examples

2. Kullback-Leibler Distance on $\mathbb{R}$: (later special case of Bregman distance, resp. φ-divergence)

$$f(x) = \mathrm{KL}(x, y) = \begin{cases} x \log \frac{x}{y} - x + y = x \log x - x \log y - x + y & x, y \geq 0, \\ & x = 0 \text{ if } y = 0 \\ +\infty & \text{otherwise} \end{cases}$$

with $0 \log 0 := 0$ and negative entropy

$$\mathrm{ent(x)} := \begin{cases} x \log x & x \geq 0 \\ +\infty & \text{otherwise} \end{cases}$$



$$f^*(p) = \mathrm{KL}^*(p, y) = \sup_{x \in \mathbb{R}} \{ \underbrace{px - x \log x + x \log y + x - y}_{g(x)} \}$$

$$\begin{aligned} \Rightarrow \quad g'(x) &= p - \log x - 1 + \log y + 1 = 0 \\ \log x &= p + \log y \\ x &= y\, e^p \\ f^*(p) &= \mathrm{KL}^*(p, y) = y e^p - y \end{aligned}$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

Primal-Dual Algorithms

Task:

$$(P) \quad \min_{x \in \mathbb{R}^d} \{g(x) + h(Ax)\}$$

$$(P) \quad \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^m} \{g(x) + h(y) \quad \text{s.t.} \quad Ax = y\}$$

Lagrangian:

$$L(x, y, p) := g(x) + h(y) + \langle p, Ax - y \rangle$$

Primal and dual problem formulations: In general $(D) \leq (P)$

$$(P) \quad \min_{x \in \mathbb{R}^d, y \in \mathbb{R}^m} \sup_{p \in \mathbb{R}^m} \{g(x) + h(y) + \langle p, Ax - y \rangle\}$$

$$(D) \quad \max_{p \in \mathbb{R}^m} \inf_{x \in R^d, y \in \mathbb{R}^m} \{g(x) + h(y) + \langle p, Ax - y \rangle\}$$

Since above in (P),

$$h(Ax) = \sup_{p \in \mathbb{R}^m} \{\langle p, Ax \rangle - h^*(p)\}, \quad \inf_{y \in \mathbb{R}^m} \{h(y) - \langle p, y \rangle\} = -h^*(p)$$

we have

$$(P) \quad \min_{x \in \mathbb{R}^d} \sup_{p \in \mathbb{R}^m} \{g(x) - h^*(p) + \langle p, Ax \rangle\}$$

$$(D) \quad \max_{p \in \mathbb{R}^m} \inf_{x \in \mathbb{R}^d} \{g(x) - h^*(p) + \langle p, Ax \rangle\}$$

Uzawa Method and Method of Multipliers

Uzawa method: alternate minimization of the Lagrangian with respect to (x, y) and gradient ascent for p

$$(x^{(r+1)}, y^{(r+1)}) \in \operatorname*{argmin}_{x \in \mathbb{R}^d, y \in \mathbb{R}^m} L(x, y, p^{(r)})$$

$$p^{(r+1)} = p^{(r)} + \gamma(Ax^{(r+1)} - y^{(r+1)}), \qquad \gamma > 0$$

Drawback: convergence under strong restrictions on g, h

Augmented Lagrangian:

$$L_\gamma(x, y, p) := g(x) + h(y) + \langle p, Ax - y \rangle + \frac{\gamma}{2}\|Ax - y\|_2^2 \quad \gamma > 0$$

$$= g(x) + h(y) + \frac{\gamma}{2}\|Ax - y + \frac{p}{\gamma}\|_2^2 \quad - \frac{1}{2\gamma}\|p\|_2^2$$

Method of Multipliers:

$$(x^{(r+1)}, y^{(r+1)}) \in \operatorname*{argmin}_{x \in \mathbb{R}^d, y \in \mathbb{R}^m} L_\gamma(x, y, p^{(r)})$$

$$p^{(r+1)} = p^{(r)} + \gamma(Ax^{(r+1)} - y^{(r+1)}), \qquad \gamma > 0$$

Drawback: often hard to minimize over x, y at the same time

Alternating Direction Method of Multipliers (ADMM)

Remedy: Alternate the minimization with respect to x and y

Initialization: $y^{(0)} \in \mathbb{R}^m$, $p^{(0)} \in \mathbb{R}^m$
Iterations: For $r = 0, 1, \ldots$

$$
\begin{aligned}
x^{(r+1)} &\in \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ g(x) + \tfrac{\gamma}{2} \| \tfrac{1}{\gamma} p^{(r)} + Ax - y^{(r)} \|_2^2 \right\} \\
y^{(r+1)} &= \operatorname{argmin}_{y \in \mathbb{R}^m} \left\{ h(y) + \tfrac{\gamma}{2} \| \tfrac{1}{\gamma} p^{(r)} + Ax^{(r+1)} - y \|_2^2 \right\} \\
&= \operatorname{prox}_{\frac{1}{\gamma} h}(\tfrac{1}{\gamma} p^{(r)} + Ax^{(r+1)}) \\
p^{(r+1)} &= p^{(r)} + \gamma (Ax^{(r+1)} - y^{(r+1)})
\end{aligned}
$$

- In imaging see S. Osher et al. under the name Alternating Split Bregman Algorithm

Scaled ADMM

Setting

$$b^{(r)} := p^{(r)}/\gamma$$

we obtain the following (scaled) ADMM:

Initialization: $y^{(0)} \in \mathbb{R}^m$, $b^{(0)} \in \mathbb{R}^m$
Iterations: For $r = 0, 1, \ldots$

$$\begin{aligned} x^{(r+1)} &\in \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ g(x) + \tfrac{\gamma}{2} \| b^{(r)} + Ax - y^{(r)} \|_2^2 \right\} \\ y^{(r+1)} &= \operatorname{argmin}_{y \in \mathbb{R}^m} \left\{ h(y) + \tfrac{\gamma}{2} \| b^{(r)} + Ax^{(r+1)} - y \|_2^2 \right\} \\ b^{(r+1)} &= b^{(r)} + Ax^{(r+1)} - y^{(r+1)} \end{aligned}$$

- Glowinski/Marocco 1976 (citations 05/2017: 901)
- Gabay/Mercier 1976 (citations 05/2017: 1260)
- Boyd et al. 2011 (citations 05/2017: 5130)

Convergence

Have a look at the dual problem:

$$(D) \quad \operatorname*{argmax}_{p\in\mathbb{R}^m} \inf_{x\in\mathbb{R}^d} \{g(x) - h^*(p) + \langle p, Ax\rangle\}$$

$$(D) \quad \operatorname*{argmin}_{p\in\mathbb{R}^m} \left\{g^*(-A^\mathsf{T}p) + h^*(p)\right\}.$$

Theorem Let $g \in \Gamma_0(\mathbb{R}^d)$, $h \in \Gamma_0(\mathbb{R}^m)$ and $A \in \mathbb{R}^{m,d}$. Assume that the Lagrangian has a saddle point. Then the sequence $\gamma\left(b^{(r)}\right)_r$ converges to a solution of (D). If in addition the first step in the ADMM algorithm has a unique solution, then $\left(x^{(r)}\right)_r$ converges to a solution of (P).

- Relation between ADMM and DRS

Reference: Eckstein, Berteskas 2009

Multiple Splittings

Task: $A_k \in \mathbb{R}^{d_k,d}$, $k = 1, \ldots, n$, $m = \sum_k d_k$

$$\operatorname*{argmin}_{x \in \mathbb{R}^d} \sum_{k=1}^{n} f_k(A_k x)$$

$$\operatorname*{argmin}_{x \in \mathbb{R}^d, y \in \mathbb{R}^m} \Big\{ \underbrace{\langle x, 0 \rangle}_{g(x)} + \underbrace{\sum_{k=1}^{n} f_k(y_k)}_{h(y)} \quad \text{s.t.} \quad A_k x = y_k,\ k = 1, \ldots, n \Big\}$$

Scaled ADMM $A := (A_1^\mathsf{T}, \ldots, A_n^\mathsf{T})^\mathsf{T}$
Iterations: For $r = 0, 1, \ldots$

$$\begin{aligned} x^{(r+1)} &\in \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ \|b^{(r)} + Ax - y^{(r)}\|_2^2 \right\} \\ y^{(r+1)} &= \operatorname{argmin}_{y \in \mathbb{R}^m} \left\{ \textstyle\sum_{k=1}^n f_k(y_k) + \frac{\gamma}{2} \|b^{(r)} + Ax^{(r+1)} - y\|_2^2 \right\} \\ b^{(r+1)} &= b^{(r)} + Ax^{(r+1)} - y^{(r+1)} \end{aligned}$$

Advantage: Separation of variables in Step 2

$$y_k^{(r+1)} = \operatorname*{argmin}_{y \in \mathbb{R}^{d_k}} \left\{ f_k(y) + \frac{\gamma}{2} \|b_k^{(r)} + A_k x^{(r+1)} - y\|_2^2 \right\}$$

Primal dual Hybrid Gradient Algorithm (PDHG)

First ADMM step requires solution of a linear system.

Arrow-Hurwicz Method: alternate the minimization in the primal and dual problems and add quadratic terms:

$$
\begin{aligned}
x^{(r+1)} &= \operatorname*{argmin}_{x\in\mathbb{R}^d} \left\{ g(x) + \langle p^{(r)}, Ax\rangle + \frac{1}{2\tau}\|x - x^{(r)}\|_2^2 \right\}, \\
&= \mathsf{prox}_{\tau g}(x^{(r)} - \tau A^\top p^{(r)}) \\
p^{(r+1)} &= \operatorname*{argmin}_{p\in\mathbb{R}^m} \left\{ h^*(p) - \langle p, Ax^{(r+1)}\rangle + \frac{1}{2\sigma}\|p - p^{(r)}\|_2^2 \right\}, \\
&= \mathsf{prox}_{\sigma h^*}(p^{(r)} + \sigma A x^{(r+1)})
\end{aligned}
$$

Primal dual Hybrid Gradient Algorithm (PDHG)

$$
\begin{aligned}
x^{(r+1)} &= \operatorname{argmin}_{x\in\mathbb{R}^d} \left\{ g(x) + \tfrac{1}{2\tau}\|x - \left(x^{(r)} - \tau A^\top p^{(r)}\right)\|_2^2 \right\} \\
&= \mathsf{prox}_{\tau g}\left(x^{(r)} - \tau A^\top p^{(r)}\right), \\
y^{(r+1)} &= \operatorname{argmin}_{y\in\mathbb{R}^n} \left\{ h(y) + \tfrac{\sigma}{2}\|\tfrac{1}{\sigma}p^{(r)} + Ax^{(r+1)} - y\|_2^2 \right\} \\
&= \mathsf{prox}_{\frac{1}{\sigma}h}\left(\tfrac{1}{\sigma}p^{(r)} + Ax^{(r+1)}\right), \\
p^{(r+1)} &= p^{(r)} + \sigma(Ax^{(r+1)} - y^{(r+1)}).
\end{aligned}
$$

1	2
3	4
5	6
7	8
9	10
11	12
13	14
15	16
17	18
19	20
21	22
23	24
25	26
27	28
29	30
31	32
33	34
35	36
37	38
39	40
41	42
43	44
45	46
47	48
49	50

Initialization: $y^{(0)}, b^{(0)} = b^{(-1)} \in \mathbb{R}^m$, $\tau, \sigma > 0$ with $\tau\sigma < 1/\|A\|_2^2$ and $\theta \in (0,1]$
Iterations: For $r = 0, 1, \ldots$

$$
\begin{aligned}
x^{(r+1)} &= \operatorname{argmin}_{x \in \mathbb{R}^d} \left\{ g(x) + \tfrac{1}{2\tau} \|x - (x^{(r)} - \tau\sigma A^\mathsf{T} \bar{b}^{(r)})\|_2^2 \right\} \\
y^{(r+1)} &= \operatorname{argmin}_{y \in \mathbb{R}^m} \left\{ h(y) + \tfrac{\sigma}{2} \|b^{(r)} + Ax^{(r+1)} - y\|_2^2 \right\} \\
b^{(r+1)} &= b^{(r)} + Ax^{(r+1)} - y^{(r+1)}. \\
\bar{b}^{(r+1)} &= b^{(r+1)} + \theta(b^{(r+1)} - b^{(r)})
\end{aligned}
$$

Theorem: Let $g \in \Gamma_0(\mathbb{R}^d)$, $h \in \Gamma_0(\mathbb{R}^m)$ and $\theta \in (0,1]$. Let $\tau, \sigma > 0$ fulfill

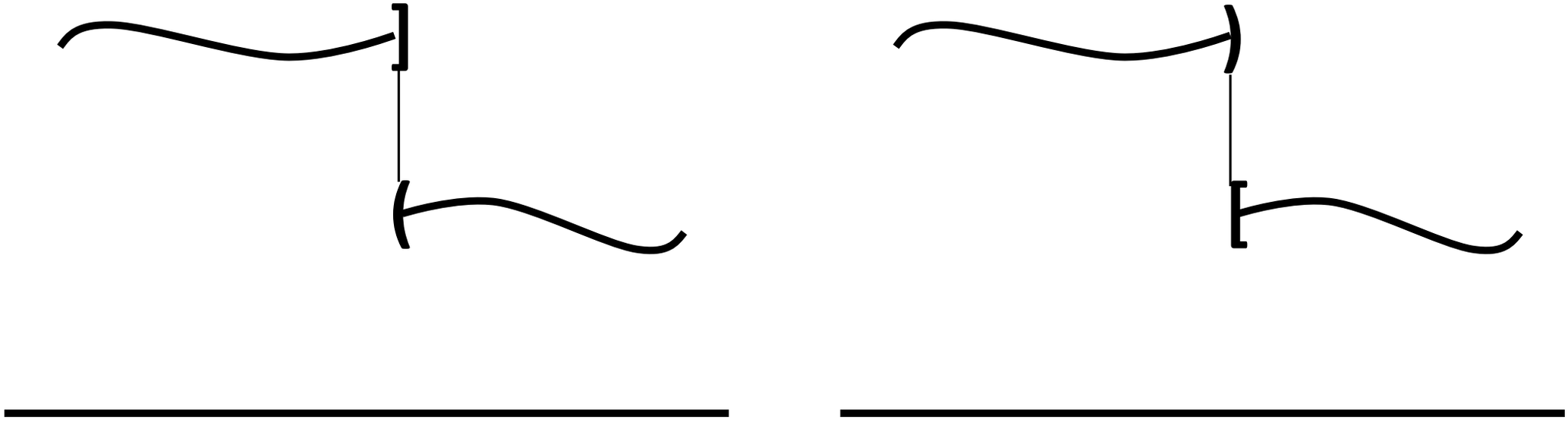
$$\tau\sigma < 1/\|A\|_2^2.$$

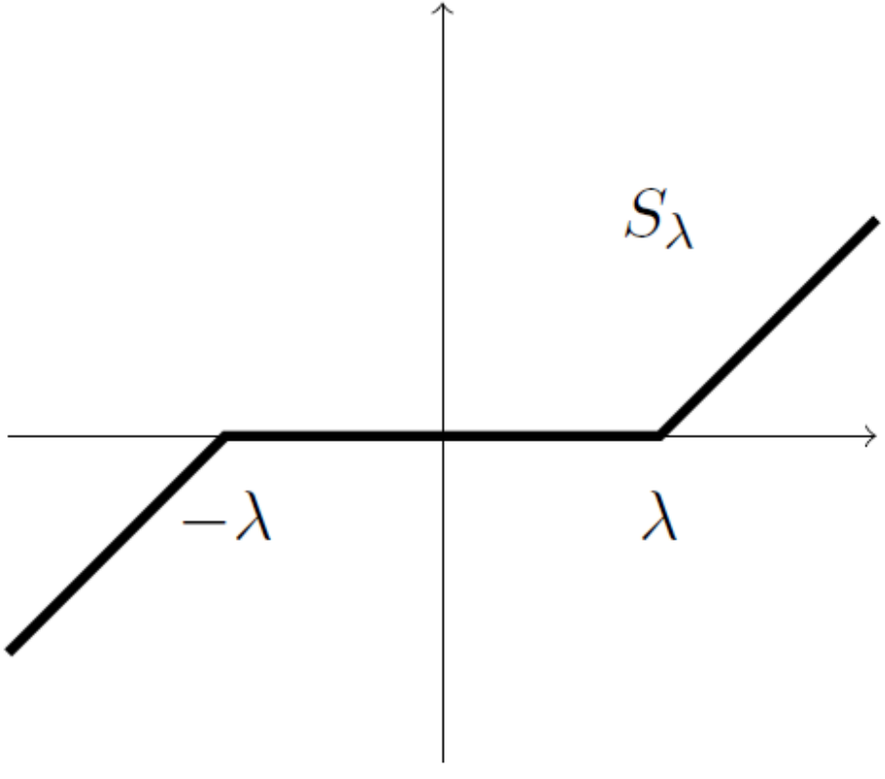
Suppose that the Lagrangian $L(x,p) := g(x) - h^*(p) + \langle Ax, p\rangle$ has a saddle point (x^*, p^*). Then the sequence $\{(x^{(r)}, p^{(r)})\}_r$ produced by Chambolle-Pock algorithm converges to a saddle point of the Lagrangian.

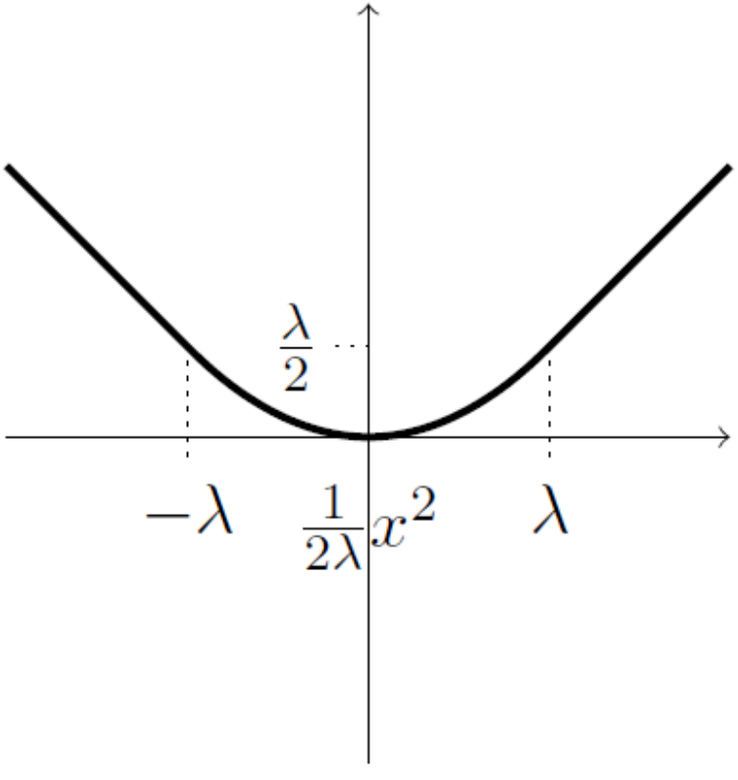
Reference: Pock/Chambolle/Cremers/Bischof 2009, Chambolle/Pock 2011 (citations 05/2017: 1800)

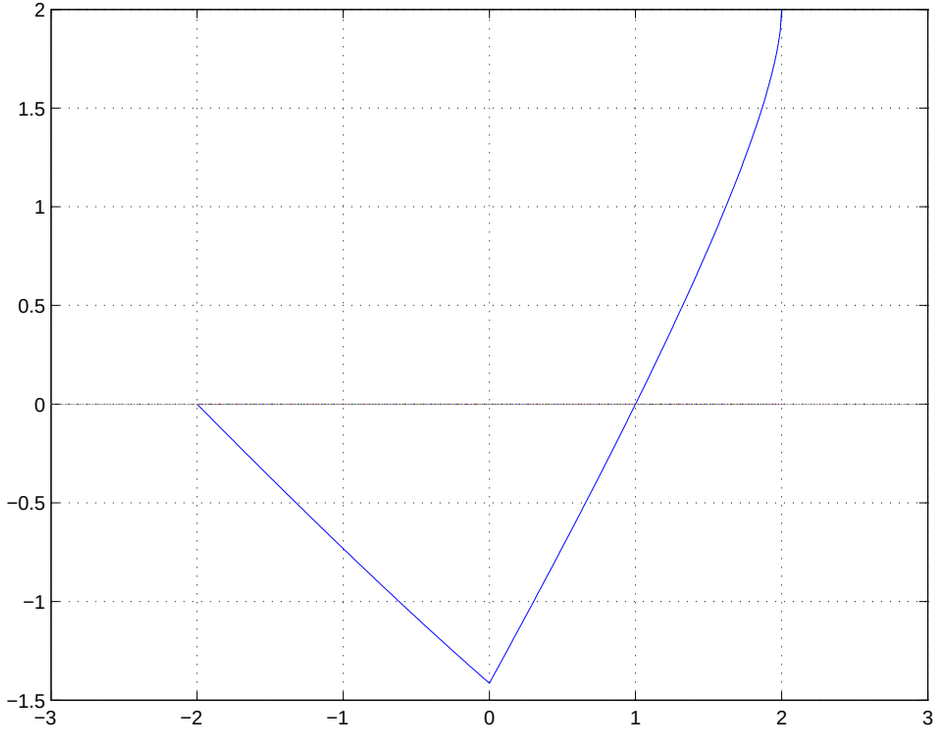
Berlin Mathematics Research Center

Funded under Germany's Excellence Strategy by
DFG Deutsche Forschungsgemeinschaft




S_λ
$-\lambda$
λ


$\frac{\lambda}{2}$
$-\lambda$
$\frac{1}{2\lambda}x^2$
λ


2
1.5
1
0.5
0
−0.5
−1
−1.5
−3
−2
−1
0
1
2
3

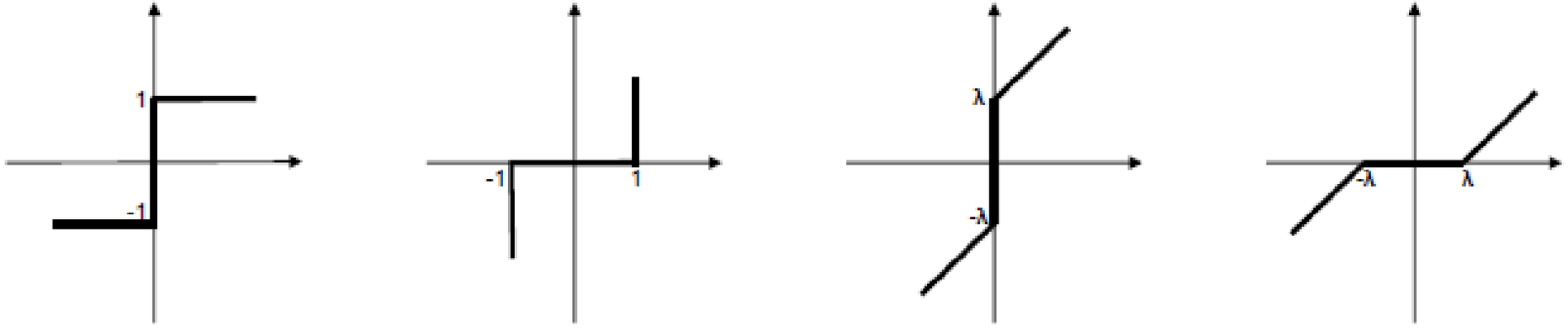
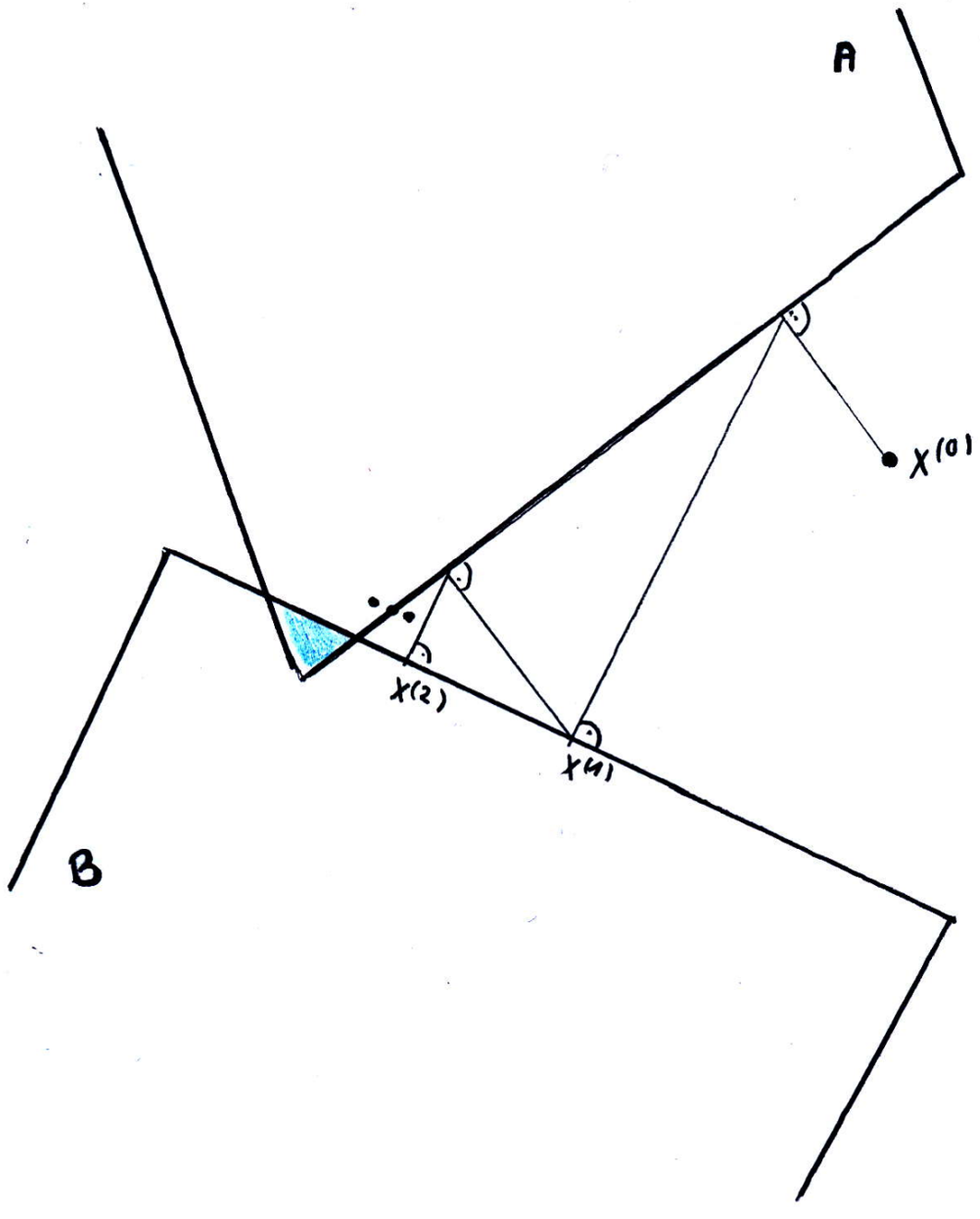
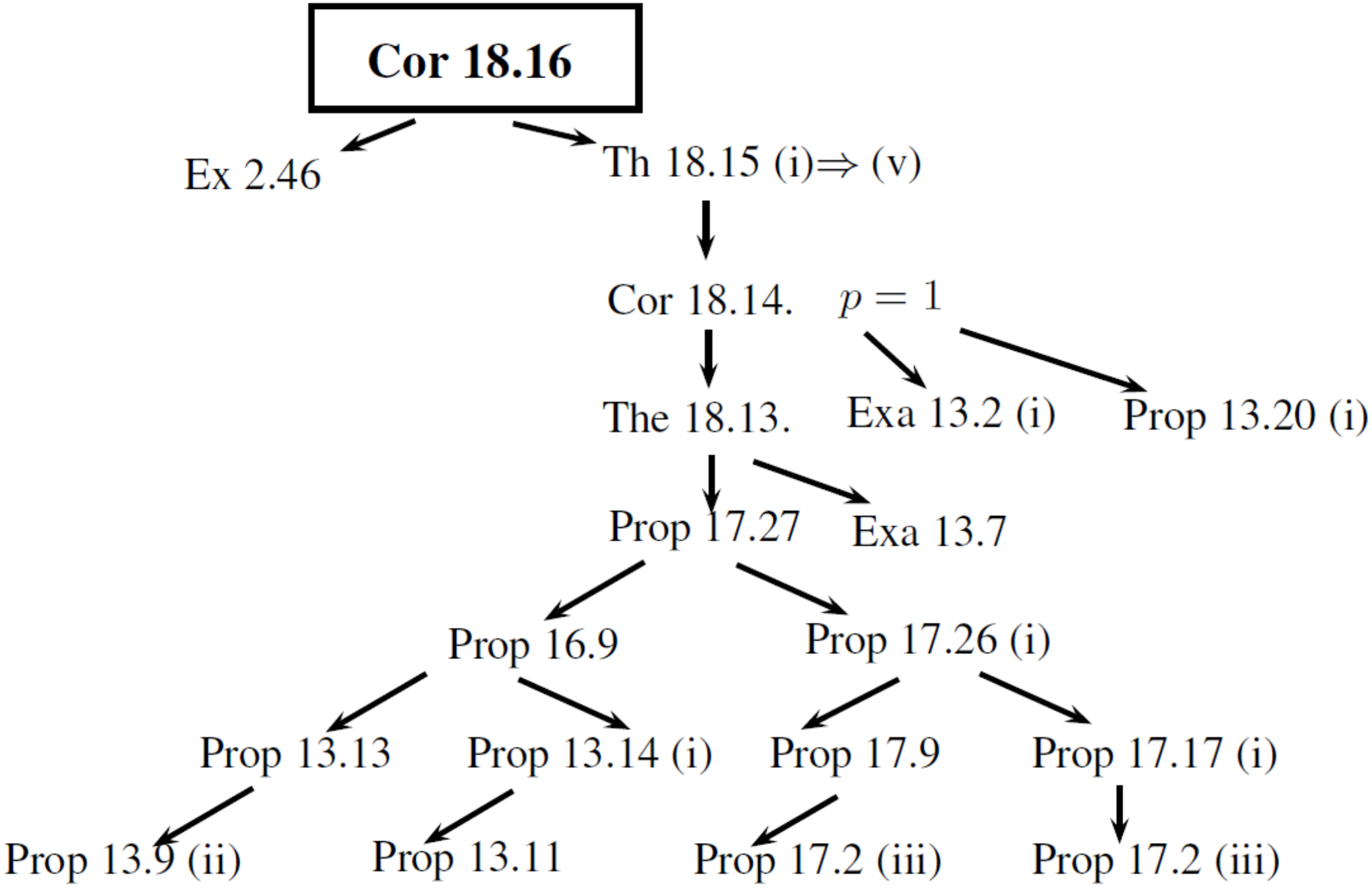


Fig. 5.1 Left to right: F, F^{-1}, $I+\lambda F$ and $(I+\lambda F)^{-1}$.



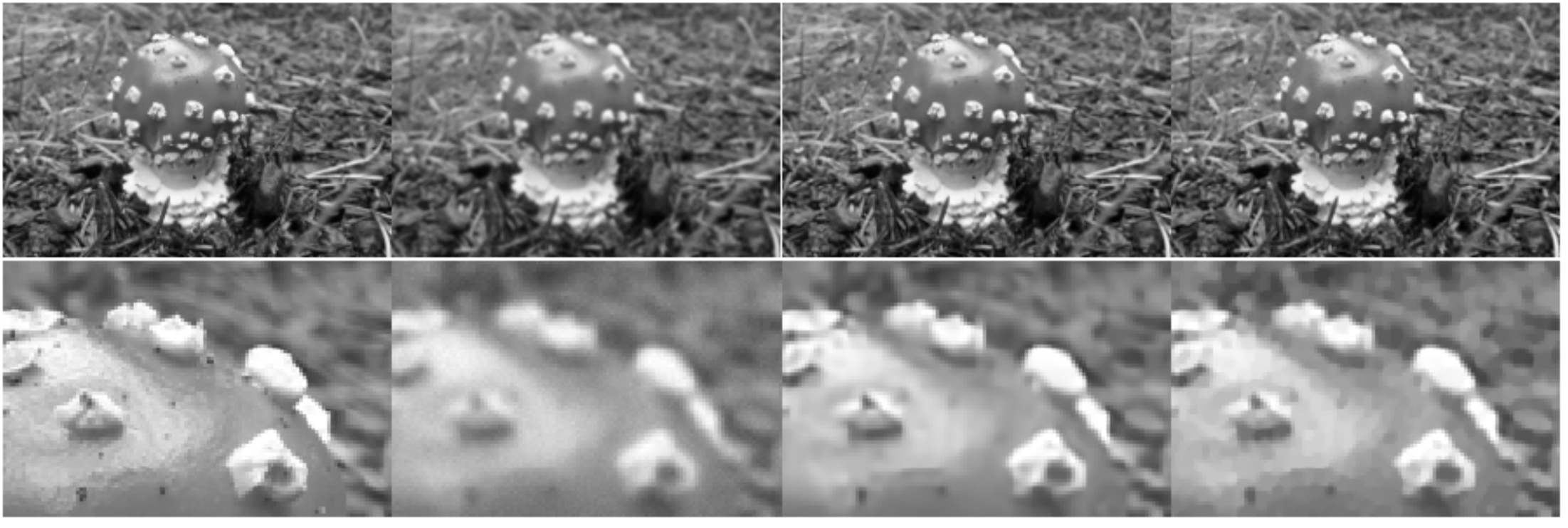
$\text{argmin}\{L_A + L_B\}$

CPPA = alternating projection algorithm


Cor 18.16
Ex 2.46
Th 18.15 (i)⇒ (v)
Cor 18.14. $p = 1$
The 18.13.
Exa 13.2 (i)
Prop 13.20 (i)
Prop 17.27
Exa 13.7
Prop 16.9
Prop 17.26 (i)
Prop 13.13
Prop 13.14 (i)
Prop 17.9
Prop 17.17 (i)
Prop 13.9 (ii)
Prop 13.11
Prop 17.2 (iii)
Prop 17.2 (iii)

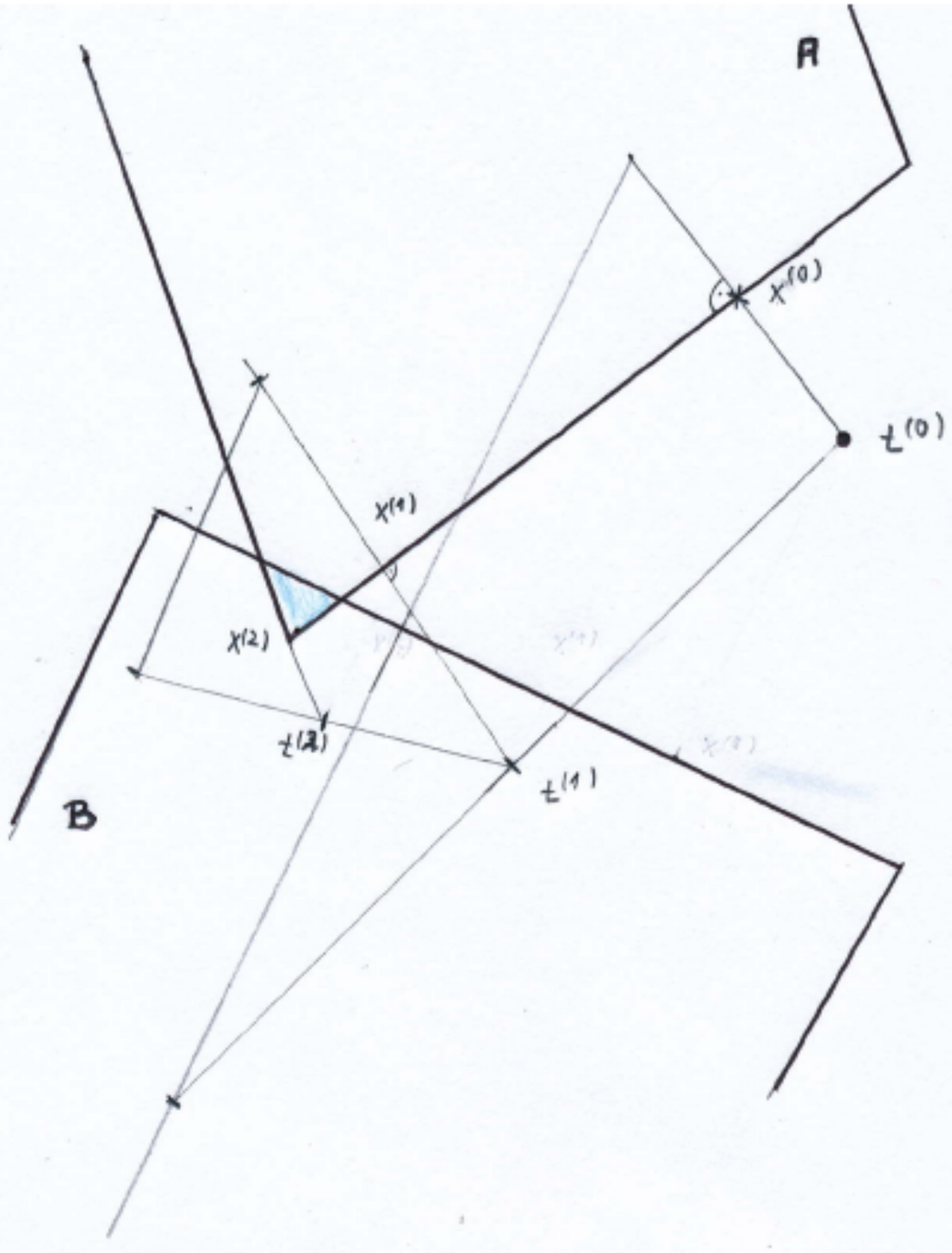
Name	activation function $\sigma(x)$	$f(x)$ with $\sigma = \text{prox}_f$
Linear activation	x	0
Saturated linear activation (SaLU)	$\begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x < -1 \end{cases}$	$\iota_{[-1,1]} = \begin{cases} 0 & \text{if } x \in [-1,1] \\ \infty & \text{if } x \notin [-1,1] \end{cases}$
Soft Thresholding	$\begin{cases} x-\lambda & \text{if } x > \lambda \\ 0 & \text{if } x \in [-\lambda,\lambda] \\ x-\lambda & \text{if } x < -\lambda \end{cases}$	$\lambda\lvert x\rvert$
Saturated linear activation (SaLU)	$\begin{cases} 1 & \text{if } x > 1 \\ x & \text{if } -1 \leq x \leq 1 \\ -1 & \text{if } x < -1 \end{cases}$	$\iota_{[-1,1]} = \begin{cases} 0 & \text{if } x \in [-1,1] \\ \infty & \text{if } x \notin [-1,1] \end{cases}$
Rectified linear unit (ReLU)	$\begin{cases} x & \text{if } x > 0 \\ 0 & \text{if } x \leq 0 \end{cases}$	$\iota_{[0,\infty)} = \begin{cases} 0 & \text{if } x \in [0,\infty) \\ \infty & \text{if } x \notin [0,\infty) \end{cases}$
Parametric rectified linear unit (PReLU)	$\begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases}, \alpha \in (0,1]$	$\begin{cases} 0 & \text{if } x > 0 \\ (\frac{1}{\alpha}-1)\frac{x^2}{2} & \text{if } x \leq 0 \end{cases}$
Bent identity activation	$\frac{x+\sqrt{x^2+1}}{2}$	$\begin{cases} x/2 - \ln(x+\frac{1}{2})/4 & \text{if } x > -\frac{1}{2} \\ \infty & \text{if } x \leq -\frac{1}{2} \end{cases}$
Inverse square root linear unit	$\begin{cases} x & \text{if } x \geq 0 \\ \frac{x}{\sqrt{x^2+1}} & \text{if } x < 0 \end{cases}$	$\begin{cases} 0 & \text{if } x \geq 0 \\ 1 - x^2/2 - \sqrt{1-x^2} & \text{if } -1 \leq x < 0 \\ \infty & \text{if } x < -1 \end{cases}$
Inverse square root unit	$\frac{x}{\sqrt{x^2+1}}$	$\begin{cases} -x^2/2 - \sqrt{1-x^2} & \text{if } \lvert x\rvert \leq 1 \\ \infty & \text{if } \lvert x\rvert > 1 \end{cases}$
Arctangent activation	$\frac{2}{\pi}\arctan(x)$	$\begin{cases} -\frac{2}{\pi}\ln(\cos(\frac{\pi x}{2})) - \frac{x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \geq 1 \end{cases}$
Hyperbolic tangent activation	$\tanh(x)$	$\begin{cases} x\text{arctanh}(x) + \frac{\ln(1-x^2) - x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \geq 1 \end{cases}$
Elliot activation	$\frac{x}{\lvert x\rvert + 1}$	$\begin{cases} -\lvert x\rvert - \ln(1-\lvert x\rvert) - \frac{x^2}{2} & \text{if } \lvert x\rvert < 1 \\ \infty & \text{if } \lvert x\rvert \geq 1 \end{cases}$
Inverse hyperbolic sine	$\text{arcsinh}(x)$	$\cosh(x) - \frac{\lvert x\rvert^2}{2}$
Logarithmic activation	$\text{sgn}(x)\ln(\lvert x\rvert + 1)$	$\exp(\lvert x\rvert) - \lvert x\rvert - 1 - \frac{\lvert x\rvert^2}{2}$

Original | Noisy PSNR 20.17 | cPNN, $\gamma = 1.99$ PSNR 29.68 | cPNN, $\gamma = 5$ PSNR 29.93 | unconst. CNN PSNR 29.86 | BM3D PSNR 29.56



Original | Blurred PSNR 25.21 | FBS-PnP with cPNN PSNR 30.51 | L_2-TV PSNR 29.77

Figure 3.: Deblurring results with blur factor $\tau = 1.5$ and Gaussian noise with $\sigma = 0.01$ using FBS-PnP with a cPNN based denoiser and L_2-TV.



$\operatorname{argmin} \{ L_A + L_B \}$

Douglas - Rachford alg.

Iterative scheme

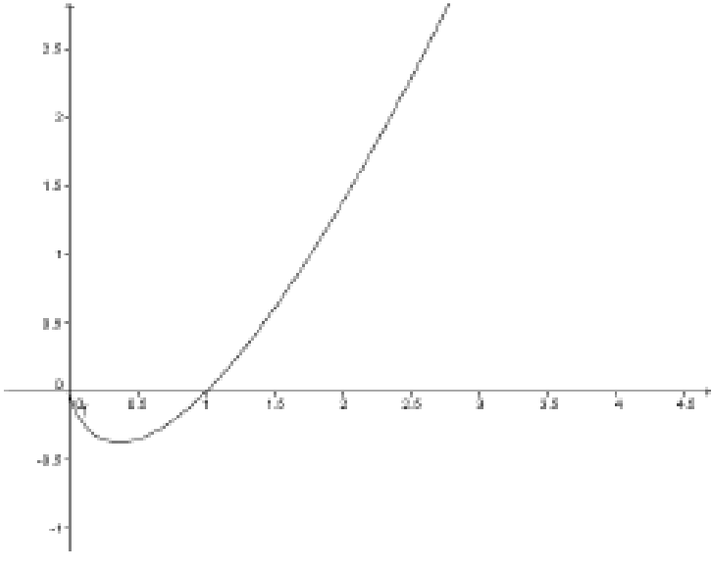
➡ **Minimization problem :**

$$\widehat{\mathrm{x}} \in \operatorname*{Argmin}_{\mathrm{x}} f(\mathrm{x}) + g(\mathrm{x})$$

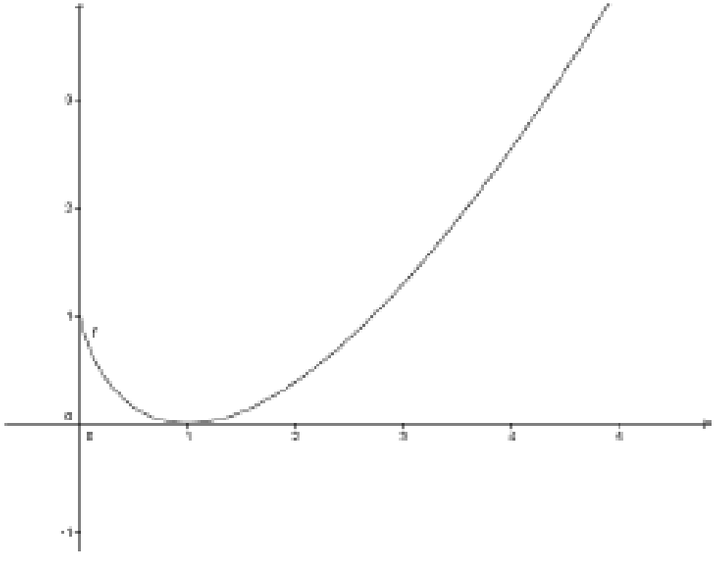
➡ **Design of a recursive sequence of the form**

$$(\forall k \in \mathbb{N}) \qquad \mathrm{x}^{[k+1]} = \boldsymbol{\Phi}\mathrm{x}^{[k]},$$

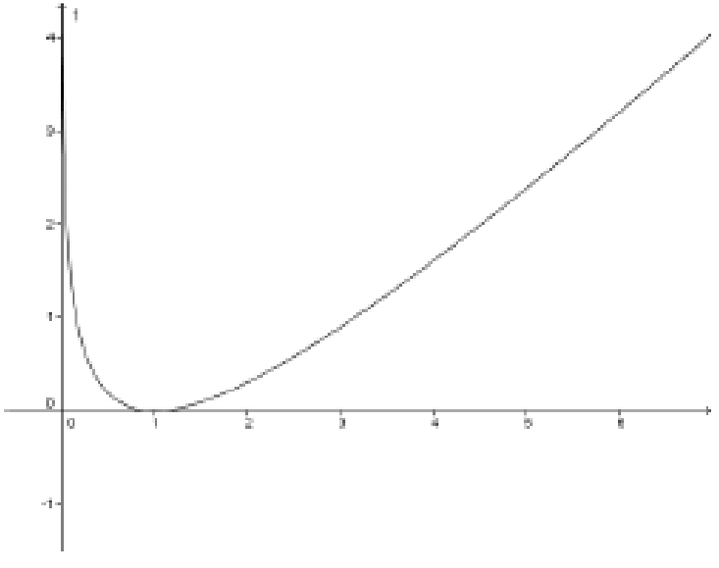
Gradient descent	$\boldsymbol{\Phi} = \mathrm{Id} - \tau(\nabla f + \nabla g)$
Proximal point algorithm	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau(f+g)}$
Forward-Backward	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau g}(\mathrm{Id} - \tau\nabla f)$
Peaceman-Rachford	$\boldsymbol{\Phi} = (2\mathrm{prox}_{\tau g} - \mathrm{Id}) \circ (2\mathrm{prox}_{\tau f} - \mathrm{Id})$
Douglas-Rachford	$\boldsymbol{\Phi} = \mathrm{prox}_{\tau g}(2\mathrm{prox}_{\tau f} - \mathrm{Id}) + \mathrm{Id} - \mathrm{prox}_{\tau f}$



(a) $\mathrm{ent}(x)$



(b) $\mathrm{KL}(x, 1)$



(c) $\mathrm{KL}(1, x)$