

OPTIMISATION METHODS FOR COMPUTATIONAL IMAGING

Chapter 4 - Optimisation algorithms

Nelly Pustelnik - ENS Lyon
Audrey Repetti - Heriot-Watt University

Journées SMAI-MODE 2022 – Limoges

Introduction to proximal methods

Gradient and subgradient algorithms

OBJECTIVE: minimise $f(\mathbf{x})$

REMARK: For simplicity, assume f to be convex

- If f has a μ -Lipschitz gradient with $\mu > 0$:

$$Explicit\ step:\quad (\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$$

with $0 < \inf_{k \in \mathbb{N}} \gamma_k$ and $\sup_{k \in \mathbb{N}} \gamma_k < 2\mu^{-1}$

Known as gradient descent algorithm

Gradient and subgradient algorithms

OBJECTIVE: minimise $f(\mathbf{x})$

REMARK: For simplicity, assume f to be convex

- If f has a μ -Lipschitz gradient with $\mu > 0$:

$$\text{Explicit step: } (\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla f(\mathbf{x}_k)$$

with $0 < \inf_{k \in \mathbb{N}} \gamma_k$ and $\sup_{k \in \mathbb{N}} \gamma_k < 2\mu^{-1}$

Known as gradient descent algorithm

- What if f is non-differentiable (but proper, lower-semicontinuous)?

→ Replace gradient with subgradient

$$(\forall \mathbf{x} \in \text{dom } \mathbf{f}) \quad \partial \mathbf{f}(\mathbf{x}) = \left\{ \mathbf{v} \in \mathbb{R}^N \mid (\forall \mathbf{y} \in \mathbb{R}^N) \quad \mathbf{f}(\mathbf{y}) \geq \mathbf{f}(\mathbf{x}) + \langle \mathbf{v} \mid \mathbf{y} - \mathbf{x} \rangle \right\}$$

From the subgradient algorithm...

OBJECTIVE: minimise $f(\mathbf{x})$

REMARK: For simplicity, assume f to be convex

👉 Subgradient algorithm [Shor, 1979]

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \mathbf{v}_k \quad \text{where } \mathbf{v}_k \in \partial f(\mathbf{x}_k)$$

with $(\forall k \in \mathbb{N}) \gamma_k > 0$ such that $\sum_{k=0}^{+\infty} \gamma_k^2 < +\infty$ and $\sum_{k=0}^{+\infty} \gamma_k = +\infty$

From the subgradient algorithm...

OBJECTIVE: minimise $f(\mathbf{x})$

REMARK: For simplicity, assume f to be convex

👉 Subgradient algorithm [Shor, 1979]

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \mathbf{v}_k \quad \text{where } \mathbf{v}_k \in \partial f(\mathbf{x}_k)$$

with $(\forall k \in \mathbb{N}) \gamma_k > 0$ such that $\sum_{k=0}^{+\infty} \gamma_k^2 < +\infty$ and $\sum_{k=0}^{+\infty} \gamma_k = +\infty$

☞ Implicit form

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \mathbf{v}_{k+1} \quad \text{where } \mathbf{v}_{k+1} \in \partial f(\mathbf{x}_{k+1})$$

$$\Leftrightarrow \mathbf{x}_k - \mathbf{x}_{k+1} \in \gamma_k \partial f(\mathbf{x}_{k+1})$$

... to the origins of the proximity operator!

OBJECTIVE: $\underset{\mathbf{x} \in \mathbb{R}^N}{\text{minimise}} f(\mathbf{x})$

REMARK: For simplicity, assume f to be convex

PROPERTY:

Let $\psi: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a convex, proper, lower-semicontinuous function. For all $\mathbf{x} \in \mathbb{R}^N$, there exists a unique vector $\hat{\mathbf{x}} \in \mathbb{R}^N$ such that $\mathbf{x} - \hat{\mathbf{x}} \in \partial\psi(\hat{\mathbf{x}})$.

- $\hat{\mathbf{x}} = \text{prox}_{\psi}(\mathbf{x})$
 - $\text{prox}_{\psi}: \mathbb{R}^N \rightarrow \mathbb{R}^N \rightsquigarrow \text{proximity operator}$

👉 Proximal point algorithm

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_k - \mathbf{x}_{k+1} \in \gamma_k \partial f(\mathbf{x}_{k+1}) \\ \Leftrightarrow \mathbf{x}_{k+1} = \text{prox}_{\gamma_k f}(\mathbf{x}_k)$$

with $\inf_{k \in \mathbb{N}} \gamma_k > 0$ such that $\sum_{k=0}^{+\infty} \gamma_k = +\infty$

Proximity operators

Definition: Proximity operator [Moreau, 1965]

Let $f \in \Gamma_0(\mathbb{R}^N)$ and $\bar{x} \in \mathbb{R}^N$. The proximal operator of f at \bar{x} is the unique minimiser of $f + \frac{1}{2}\|\cdot - \bar{x}\|^2$, i.e.

$$\text{prox}_f(\bar{x}) = \operatorname{argmin}_{x \in \mathbb{R}^N} f(x) + \frac{1}{2}\|x - \bar{x}\|^2$$

Characterisation of the proximity operator

Let $f \in \Gamma_0(\mathbb{R}^N)$ and $\bar{x} \in \mathbb{R}^N$. We have

$$\bar{p} = \text{prox}_f(\bar{x}) \Leftrightarrow \bar{x} - \bar{p} \in \partial f(\bar{p})$$

Proximity operators: Remarks

- The proximity operator can be seen as a *generalisation* of the projection operator .

Let \mathcal{C} be a closed, convex and non-empty subset of \mathbb{R}^N , and let $\bar{\mathbf{x}} \in \mathbb{R}^N$. The projection of $\bar{\mathbf{x}}$ onto \mathcal{C} is given by

$$\Pi_{\mathcal{C}}(\bar{\mathbf{x}}) = \operatorname{argmin}_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \iota_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 = \operatorname{prox}_{\iota_{\mathcal{C}}}(\bar{\mathbf{x}})$$

Proximity operators: Remarks

- The proximity operator can be seen as a *generalisation* of the projection operator.

Let \mathcal{C} be a closed, convex and non-empty subset of \mathbb{R}^N , and let $\bar{\mathbf{x}} \in \mathbb{R}^N$. The projection of $\bar{\mathbf{x}}$ onto \mathcal{C} is given by

$$\Pi_{\mathcal{C}}(\bar{\mathbf{x}}) = \operatorname{argmin}_{\mathbf{x} \in \mathcal{C}} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \iota_{\mathcal{C}}(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \bar{\mathbf{x}}\|^2 = \operatorname{prox}_{\iota_{\mathcal{C}}}(\bar{\mathbf{x}})$$

- Additively separable functions: Let $f \in \Gamma_0(\mathbb{R}^N)$ be such that

$$(\forall \mathbf{x} = (x^{(n)})_{1 \leq n \leq N} \in \mathbb{R}^N) \quad f(\mathbf{x}) = \sum_{n=1}^N \psi_n(x^{(n)}). \text{ Let}$$

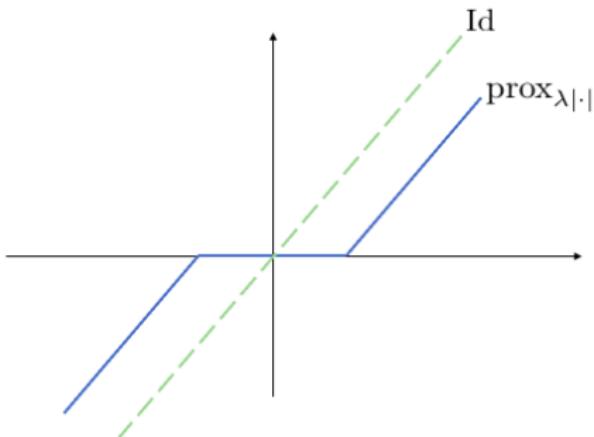
$$\bar{\mathbf{x}} = (\bar{x}^{(n)})_{1 \leq n \leq N} \in \mathbb{R}^N. \text{ Then we have } \operatorname{prox}_f(\bar{\mathbf{x}}) = \left(\operatorname{prox}_{\psi_n}(\bar{x}^{(n)}) \right)_{1 \leq n \leq N}$$

Proximity operator examples: ℓ_1 norm

Let $f = \lambda \|\cdot\|_1$ with $\lambda > 0$. For every $\bar{x} \in \mathbb{R}^N$, we have

$$\text{prox}_f(\bar{x}) = (\text{prox}_{\lambda|\cdot|}(\bar{x}^{(n)}))_{1 \leq n \leq N}$$

where, for every $n \in \{1, \dots, N\}$, $\text{prox}_{\lambda|\cdot|}(\bar{x}^{(n)}) = \text{sign}(\bar{x}^{(n)}) \max\{|\bar{x}^{(n)}| - \lambda, 0\}$

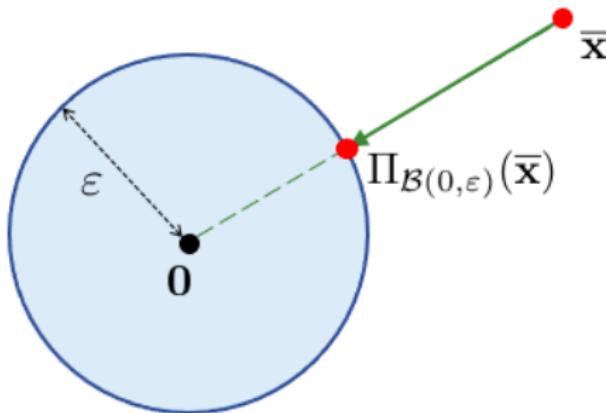


REMARK: Also known as *soft-thresholding* operator

Proximity operator examples: ℓ_2 ball

Let $f = \iota_{\mathcal{B}(\mathbf{0}, \varepsilon)}$ with $\mathcal{B}(\mathbf{0}, \varepsilon)$ denoting the ℓ_2 ball centred at $\mathbf{0}$ with radius $\varepsilon > 0$.
For every $\bar{\mathbf{x}} \in \mathbb{R}^N$, we have

$$\text{prox}_f(\bar{\mathbf{x}}) = \Pi_{\mathcal{B}(\mathbf{0}, \varepsilon)}(\bar{\mathbf{x}}) = \min\{\varepsilon, \|\bar{\mathbf{x}}\|\} \frac{\bar{\mathbf{x}}}{\|\bar{\mathbf{x}}\|}$$



Proximity operator: Denoising interpretation

GAUSSIAN DENOISING PROBLEM

$$\mathbf{z} = \bar{\mathbf{x}} + \mathbf{w}$$

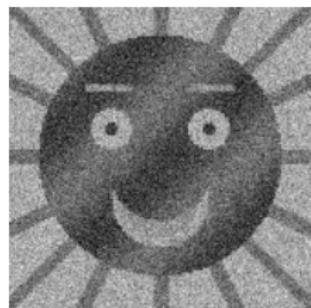
- $\mathbf{w} \in \mathbb{R}^N$ is a realisation of a white Gaussian noise with variance $\sigma > 0$

MAP ESTIMATE: Find $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \underbrace{\frac{1}{2} \|\mathbf{x} - \mathbf{z}\|^2}_{\text{data-fidelity}} + \underbrace{F(\mathbf{x})}_{\text{regularisation}} = \operatorname{prox}_F(\mathbf{z})$

Original \mathbf{x}



Noisy \mathbf{z}



Proximity operator: Online toolbox

- Matlab codes for most of existing functions used in data science can be found on the *prox repository*: <http://proximity-operator.net>
- The ODL library on Python provides functions for some proximity operators and algorithms: <https://odlgroup.github.io/odl/>

Forward-Backward algorithm
acceleration techniques

“Semi-smooth” minimisation problem

MINIMISATION PROBLEM

$$\text{Find } \hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) = h(\mathbf{x}) + g(\mathbf{x})\}$$

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex, proper and β -Lipschitz differentiable
- $g \in \Gamma_0(\mathbb{R}^N)$

EXAMPLE: $\mathbf{z} = \Phi \bar{\mathbf{x}} + \mathbf{w}$

- $\Phi: \mathbb{R}^N \rightarrow \mathbb{R}^M$ measurement operator
- $\mathbf{w} \in \mathbb{R}^M$ realisation of white Gaussian noise

$$\underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{minimise}} \underbrace{\frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2}_{h(\mathbf{x})} + \underbrace{\eta \|\Psi \mathbf{x}\|_1}_{g(\mathbf{x})}$$

- $\eta > 0$ is a regularisation parameter
- $\Psi: \mathbb{R}^N \rightarrow \mathbb{R}^L$ sparsifying operator (e.g. wavelet transform)

“Semi-smooth” minimisation problem

MINIMISATION PROBLEM

$$\text{Find } \hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{ f(\mathbf{x}) = h(\mathbf{x}) + g(\mathbf{x}) \}$$

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex, proper and β -Lipschitz differentiable
- $g \in \Gamma_0(\mathbb{R}^N)$

REMARK: Usually prox_f does not have a closed form solution

IDEA: Use splitting methods to handle h and f separately

FB algorithm

OBJECTIVE: Find a minimiser of $f = h + g$

Let $\mathbf{x}_0 \in \text{dom } g$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2\beta^{-1}[$

For $k = 0, 1, \dots$

$$\lfloor \mathbf{x}_{k+1} = \text{prox}_{\gamma_k g} (\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k))$$

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be a sequence generated by the FB algorithm. Let, for every $k \in \mathbb{N}$, $0 < \gamma_k < 2\beta^{-1}$, where β is the Lipschitz constant of ∇h . Then

- $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a minimiser \mathbf{x}^* of f
- $(f(\mathbf{x}_k))_{k \in \mathbb{N}}$ is a non-increasing sequence converging to $f(\mathbf{x}^*)$.

FB algorithm: Limits

OBJECTIVE: Find a minimiser of $f = h + g$

Let $\mathbf{x}_0 \in \text{dom } g$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2\beta^{-1}[$

For $k = 0, 1, \dots$

$$\lfloor \mathbf{x}_{k+1} = \text{prox}_{\gamma_k g} (\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k))$$

- Convergence may be slow in practice...
 - Use Nesterov acceleration (*inertia*)
 - Use second order information (*preconditioning*)
- What if $\text{prox}_{\gamma_k g}$ does not have a closed form?
 - Use sub-iterations (e.g. dual FB algorithm)
 - Use more advanced methods (e.g. primal-dual algorithms)
 - Use more advanced methods (e.g. primal-dual algorithms)

FB acceleration: Inertia

What is inertia?

Let $k \in \mathbb{N}$ be the current iteration of an iterative algorithm.

Inertia aims to use information from the **previous iterate(s)** $(\mathbf{x}_{k'})_{k' \leq k}$ to build the next iterate \mathbf{x}_{k+1} .

WHY? Use memory to go faster!

For FB we have:

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{T}_k(\mathbf{x}_k) \text{ where } \mathbf{T}_k = \text{prox}_{\gamma_k g} \circ (\text{Id} - \gamma_k \nabla h)$$

Introducing inertia would lead to

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \tilde{\mathbf{T}}_k(\mathbf{x}_1, \dots, \mathbf{x}_k)$$

QUESTION: How to choose $\tilde{\mathbf{T}}_k$?

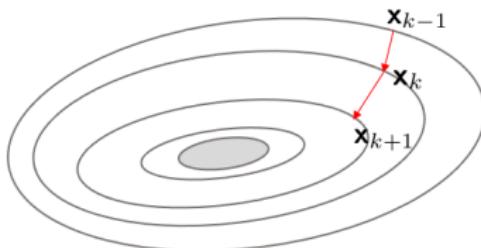
REMARK: In general $\tilde{\mathbf{T}}_k$ only depends on $(\mathbf{x}_k, \mathbf{x}_{k-1})$ to avoid memory issues

Particular case: Gradient Descent algorithm

Let $g \equiv 0$. In this case $\text{prox}_g = \text{Id}$.

The *path* taken by the iterates $(\mathbf{x}_k)_{k \in \mathbb{N}}$ is determined by the opposite of the gradient direction:

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)$$



- >We have, for every $k \in \mathbb{N}$, $h(\mathbf{x}_{k+1}) \leq h(\mathbf{x}_k)$
- This is the **Gradient Descent (GD) algorithm**

Particular case: Inertia for GD algorithm

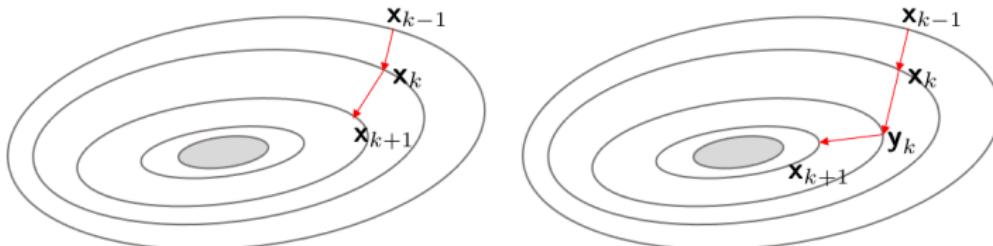
Let $g \equiv 0$. In this case $\text{prox}_g = \text{Id}$.

The *path* taken by the iterates $(\mathbf{x}_k)_{k \in \mathbb{N}}$ is determined by the opposite of the gradient direction:

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)$$

ACCELERATION: *Nesterov-type accelerated GD algorithm* [Nesterov, 83]

$$(\forall k \in \mathbb{N}) \quad \begin{aligned} \mathbf{x}_{k+1} &= \mathbf{y}_k - \gamma \nabla h(\mathbf{y}_k) \quad \text{with } \gamma \in]0, 1/\beta[\\ \mathbf{y}_{k+1} &= \mathbf{x}_{k+1} + \frac{k}{k+\alpha} (\mathbf{x}_{k+1} - \mathbf{x}_k) \end{aligned}$$



Particular case: Inertia for GD algorithm

Let $g \equiv 0$. In this case $\text{prox}_g = \text{Id}$.

The *path* taken by the iterates $(\mathbf{x}_k)_{k \in \mathbb{N}}$ is determined by the opposite of the gradient direction:

$$(\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} = \mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)$$

ACCELERATION: *Nesterov-type accelerated GD algorithm* [Nesterov, 83]

$$\begin{aligned} (\forall k \in \mathbb{N}) \quad \mathbf{x}_{k+1} &= \mathbf{y}_k - \gamma \nabla h(\mathbf{y}_k) \quad \text{with } \gamma \in]0, 1/\beta[\\ \mathbf{y}_{k+1} &= \mathbf{x}_{k+1} + \frac{k}{k+\alpha} (\mathbf{x}_{k+1} - \mathbf{x}_k) \end{aligned}$$

- Each iteration takes nearly the same computational cost as GD
- **not** a *descent* method (i.e. we may not have $h(\mathbf{x}_{k+1}) \leq h(\mathbf{x}_k)$)
- $\alpha = 3$ is the smallest constant that guarantees $O(1/k^2)$ convergence, and can be replaced by any other $\alpha \geq 3$

Particular case: Inertia for GD algorithm - example

MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h_1(\mathbf{x}) + h_2(\mathbf{x})$

- h_1 is an ℓ_2 -norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad h_1(\mathbf{x}) = \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2$ with $\Phi \in \mathbb{R}^{M \times N}$ and $\mathbf{z} \in \mathbb{R}^M$ are realisations of i.i.d. Gaussian random variables.
- h_2 is the Huber function: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad h_2(\mathbf{x}) = \sum_{n=1}^N \varphi_n(x^{(n)})$ with,

$$(\forall n \in \{1, \dots, N\}) \quad \varphi_n(x^{(n)}) = \begin{cases} \frac{(x^{(n)})^2}{2\delta} & \text{if } |x^{(n)}| \leq \delta \\ |x^{(n)}| - \frac{\delta}{2} & \text{otherwise} \end{cases}$$

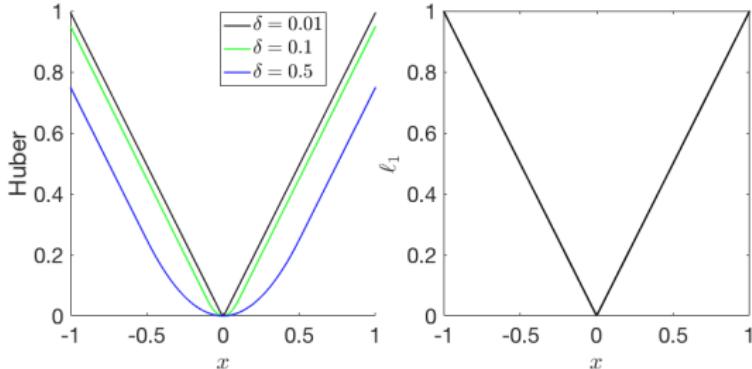
Particular case: Inertia for GD algorithm - example

MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h_1(\mathbf{x}) + h_2(\mathbf{x})$

- h_1 is an ℓ_2 -norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad h_1(\mathbf{x}) = \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2$ with $\Phi \in \mathbb{R}^{M \times N}$ and $\mathbf{z} \in \mathbb{R}^M$ are realisations of i.i.d. Gaussian random variables.
- h_2 is the Huber function: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad h_2(\mathbf{x}) = \sum_{n=1}^N \varphi_n(x^{(n)})$ with,

$$(\forall n \in \{1, \dots, N\}) \quad \varphi_n(x^{(n)}) = \begin{cases} \frac{(x^{(n)})^2}{2\delta} & \text{if } |x^{(n)}| \leq \delta \\ |x^{(n)}| - \frac{\delta}{2} & \text{otherwise} \end{cases}$$

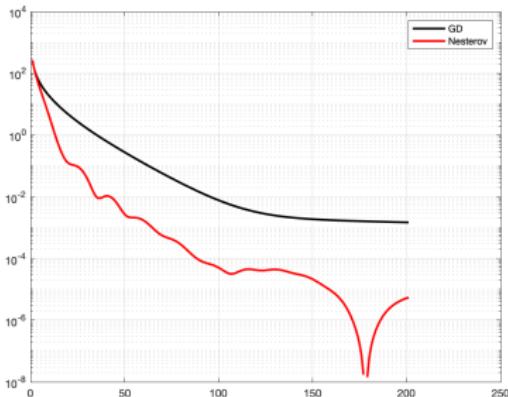
REMARK: The Huber function is a smoothed approximation of the ℓ_1 function:



Particular case: Inertia for GD algorithm - example

MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h_1(\mathbf{x}) + h_2(\mathbf{x})$

- h_1 is an ℓ_2 -norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad h_1(\mathbf{x}) = \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2$ with $\Phi \in \mathbb{R}^{M \times N}$ and $\mathbf{z} \in \mathbb{R}^M$ are realisations of i.i.d. Gaussian random variables.
- h_2 is the Huber function
- h is Lipschitz differentiable, with Lipschitz constant $\beta = \|\Phi\|^2 + \frac{1}{\delta}$
- $N = 1000$ and $M = 500$



Extension to FB algorithm: FISTA [Beck and Teboulle, 2009]

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

Let $\mathbf{x}_0 \in \text{dom } g$ and $\mathbf{y}_0 = \mathbf{x}_0$. Let $\theta_0 = 1$.

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 1/\beta]$

$$\mathbf{x}_{k+1} = \text{prox}_{\gamma_k g} \left(\mathbf{y}_k - \gamma_k \nabla h(\mathbf{y}_k) \right)$$

$$\theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}$$

$$\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}} (\mathbf{x}_{k+1} - \mathbf{x}_k)$$

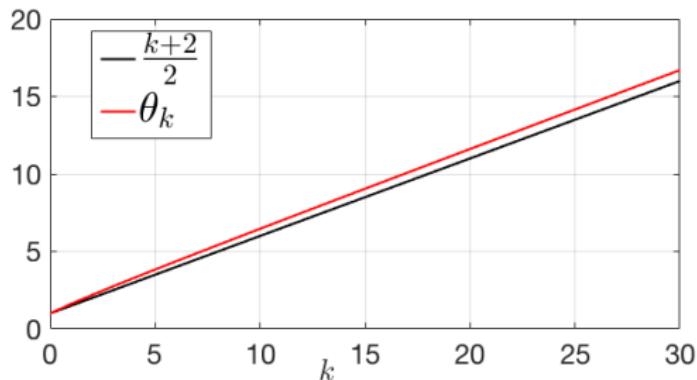
- Adopt the inertia (momentum) strategy proposed by Nesterov

Behaviour of FISTA' momentum parameter

- Momentum parameter: $\beta_k = \frac{\theta_k - 1}{\theta_{k+1}}$

where $\theta_0 = 1$, and $(\forall k \geq 1) \quad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}$

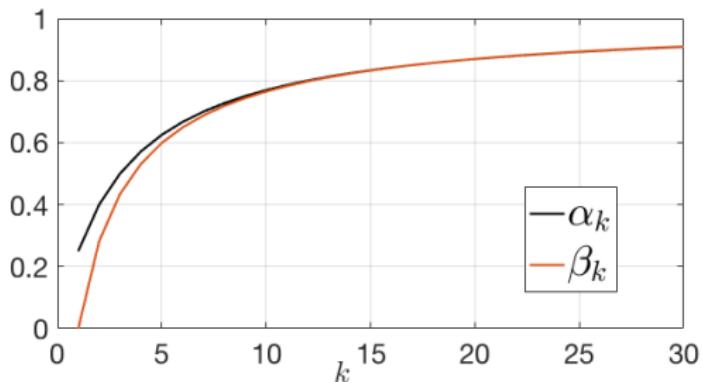
For every $k \in \mathbb{N}$, we have $\theta_k \geq \frac{k+2}{2}$



Link between FISTA and Nesterov momentum parameters

- Nesterov: $\alpha_k = \frac{k}{k+3}$
- FISTA: $\beta_k = \frac{\theta_k - 1}{\theta_{k+1}}$ where $\theta_0 = 1$, and $(\forall k \geq 1) \quad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}$

$(\alpha_k)_{k \in \mathbb{N}}$ and $(\beta_k)_{k \in \mathbb{N}}$ have similar behaviour when k is (approximately) larger than 15, and both converge to 1 when $k \rightarrow +\infty$



Convergence rate of FISTA

We analyse the **convergence rate** of an iterative algorithm in the sense of “*where the algorithm is after a large number of iterations k* ” (for objective value).

Convergence rate of FISTA

We analyse the **convergence rate** of an iterative algorithm in the sense of “*where the algorithm is after a large number of iterations k* ” (for objective value).

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be generated by **FB iterations** with $(\gamma_k)_{k \in \mathbb{N}}$ in $]0, 1/\beta^{-1}[$.
 $(f(\mathbf{x}_k))_{k \in \mathbb{N}}$ converges to $f(\mathbf{x}^*)$ at the rate $O(1/k)$:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\beta}{k+1} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be generated by **FISTA**.
 $(f(\mathbf{x}_k))_{k \in \mathbb{N}}$ converges to $f(\mathbf{x}^*) = \min_{\mathbb{R}^N} f$ at the rate $O(1/k^2)$:

$$f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{2\beta}{(k+1)^2} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

Proof: A complete proof is provided in [Beck and Teboulle, 2009].

Convergence rate of FISTA

We analyse the **convergence rate** of an iterative algorithm in the sense of “*where the algorithm is after a large number of iterations k* ” (for objective value).

- Improved iteration complexity:
 - FB: $f(\mathbf{x}_k) - f(\mathbf{x}^*) \approx O(1/k)$
 - FISTA: $f(\mathbf{x}_k) - f(\mathbf{x}^*) \approx O(1/k^2)$
- (Almost) Same computational complexity per iteration as FB

ISSUES:

- No convergence guarantees of the sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$
- Lipschitz constant β of h can still impact (a lot) the practical convergence rate (or β is unknown)
- Still need to implement the prox efficiently (or need sub-iterations)

Example: Image deconvolution [Beck and Teboulle, 2009]

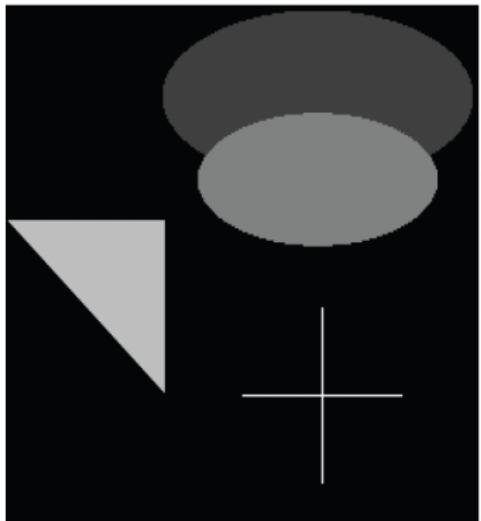
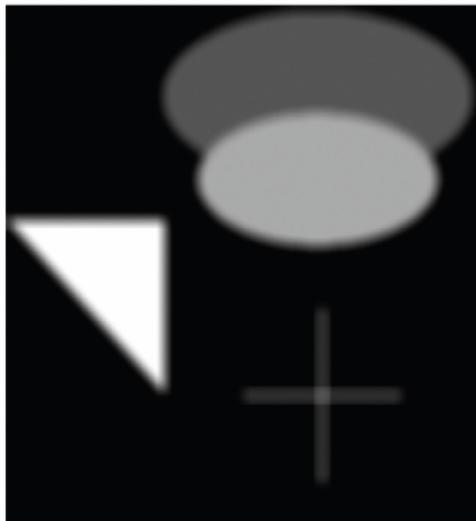
INVERSE PROBLEM: $\mathbf{z} = \Phi\bar{\mathbf{x}} + \mathbf{w}$

- \mathbf{w} is a realisation on a zero-mean white Gaussian noise
- $\Phi = \mathbf{R}\mathbf{D}$ where \mathbf{R} is the matrix representing a Gaussian blur operator and \mathbf{D} is the inverse of a Haar wavelet transform

MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} F(\mathbf{x}) + R(\mathbf{x})$

- Data-fidelity: F is an ℓ_2 -norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad F(\mathbf{x}) = \frac{1}{2} \|\Phi\mathbf{x} - \mathbf{z}\|^2$
~~~ Lipschitz-differentiable
- Regularisation:  $R$  is an  $\ell_1$ -norm:  $(\forall \mathbf{x} \in \mathbb{R}^N) \quad R(\mathbf{x}) = \eta \|\mathbf{x}\|_1$   
~~~  $\text{prox}_R$  is the soft-thresholding operator

Example: Image deconvolution [Beck and Teboulle, 2009]

 \bar{x}  $z = \Phi \bar{x} + w$

Example: Image deconvolution [Beck and Teboulle, 2009]

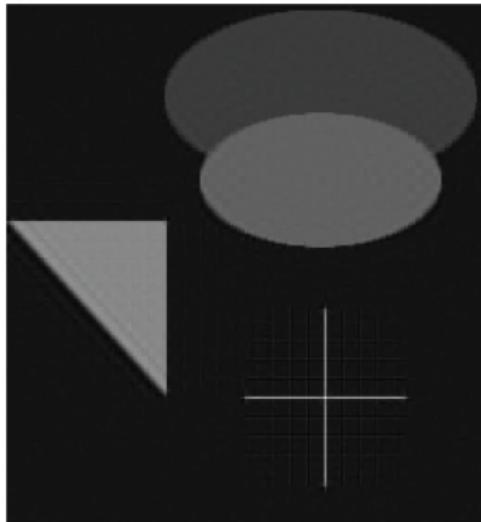


FB: \mathbf{x}_k with $k = 100$



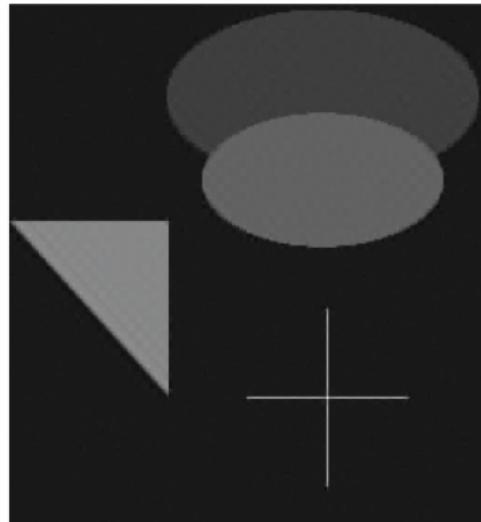
\mathbf{x}_k with $k = 200$

Example: Image deconvolution [Beck and Teboulle, 2009]



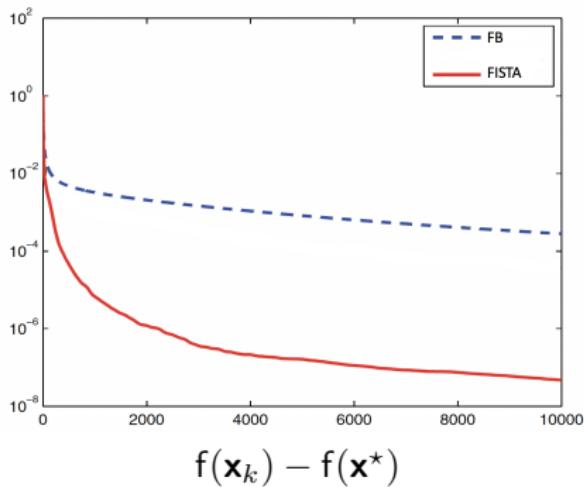
FISTA:

\mathbf{x}_k with $k = 100$



\mathbf{x}_k with $k = 200$

Example: Image deconvolution [Beck and Teboulle, 2009]



Convergent FISTA

“general” FISTA [Beck and Teboulle, 2009]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond_θ)

Let $\mathbf{x}_0 \in \text{dom } g$ and $\mathbf{y}_0 = \mathbf{x}_0$. Let $\theta_0 = 1$.

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 1/\beta]$

$$\mathbf{x}_{k+1} = \text{prox}_{\gamma_k g} \left(\mathbf{y}_k - \gamma_k \nabla h(\mathbf{y}_k) \right)$$

Choose θ_{k+1} using rule (Cond_θ)

$$\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}} (\mathbf{x}_{k+1} - \mathbf{x}_k)$$

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be a sequence generated by FISTA. Under (Cond_θ), $(\mathbf{x}_k)_{k \in \mathbb{N}}$ satisfies

$$(\forall k \in \mathbb{N}) \quad f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\beta}{2\theta_k^2} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

“general” FISTA [Beck and Teboulle, 2009]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond_θ)

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be a sequence generated by FISTA. Under (Cond_θ), $(\mathbf{x}_k)_{k \in \mathbb{N}}$ satisfies

$$(\forall k \in \mathbb{N}) \quad f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\beta}{2\theta_k^2} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

Let $(\theta)_{k \in \mathbb{N}}$ satisfying (Cond_θ). Then $(\forall k \in \mathbb{N}) \quad \theta_k \geq k + 1$.

Hence, there exists $v \in]0, +\infty[$ such that $(\forall k \in \mathbb{N}) \quad f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{v}{(k+1)^2}$
where v depends on the exact choice of $(\theta_k)_{k \in \mathbb{N}}$

In other words, FISTA has a convergence rate $O(1/k^2)$ for any choice of $(\theta_k)_{k \in \mathbb{N}}$ satisfying (Cond_θ).

REMARK: Best choice to obtain the smaller constant v will be when (Cond_θ) is **saturated**, i.e. when the inequality becomes an equality.

“general” FISTA [Beck and Teboulle, 2009]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond_θ)

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be a sequence generated by FISTA. Under (Cond_θ), $(\mathbf{x}_k)_{k \in \mathbb{N}}$ satisfies

$$(\forall k \in \mathbb{N}) \quad f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\beta}{2\theta_k^2} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$$

The sequence $(\theta_k)_{k \in \mathbb{N}}$ defined by $\begin{cases} \theta_0 = 1 \\ (\forall k \in \mathbb{N}) \quad \theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2} \end{cases}$

achieves the equality in (Cond_θ)

Under this condition, we have $(\forall k \in \mathbb{N}) \quad f(\mathbf{x}_k) - f(\mathbf{x}^*) \leq \frac{\beta}{2(k+1)^2} \|\mathbf{x}_0 - \mathbf{x}^*\|^2$

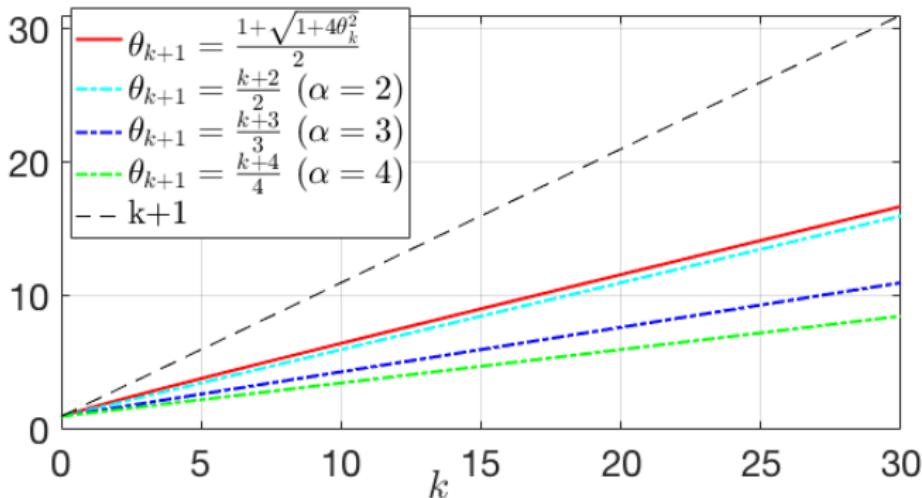
In other words, this is the tightest condition to obtain the best convergence rate.

QUESTION: Are there other (possibly better) choices?

Convergent FISTA [Chambolle and Dossal, 2015]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond _{θ})

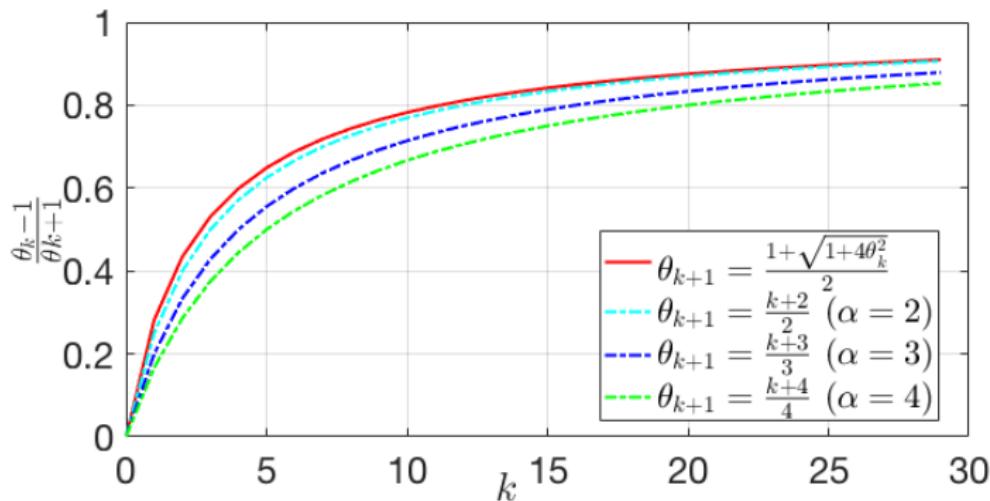
Let $\alpha \geq 2$ and let $(\forall k \in \mathbb{N}) \quad \theta_k = \frac{k+\alpha}{\alpha}$.



Convergent FISTA [Chambolle and Dossal, 2015]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond _{θ})

Let $\alpha \geq 2$ and let $(\forall k \in \mathbb{N}) \quad \theta_k = \frac{k+\alpha}{\alpha}$.



Convergent FISTA [Chambolle and Dossal, 2015]

Define $(\theta_k)_{k \in \mathbb{N}}$ as $\begin{cases} \theta_0 = 1 \\ (\forall k \geq 1) \quad \theta_k^2 - \theta_k \leq \theta_{k-1}^2 \end{cases}$ (Cond _{θ})

Let $\alpha \geq 2$ and let $(\forall k \in \mathbb{N}) \quad \theta_k = \frac{k+\alpha}{\alpha}$.

The sequence $(\theta_k)_{k \in \mathbb{N}}$ defined above satisfies (Cond _{θ})

Proof: Let $(\forall k \geq 1) \quad \rho_k = \theta_{k-1}^2 - \theta_k^2 + \theta_k$.

$$\begin{aligned} \rho_k &= \frac{1}{\alpha^2} \left((k+\alpha-1)^2 - (k+\alpha)^2 + \alpha(k+\alpha) \right) \\ &= \frac{1}{\alpha^2} \left((\alpha-2)k + (\alpha-1)^2 \right) \\ &\geq 0 \end{aligned}$$

Since $\alpha \geq 2$, we have $\rho_k \geq 0$, hence the result. \square

CONSEQUENCE: $(\exists v > 0)(\forall k \in \mathbb{N}) \quad \mathbf{f}(\mathbf{x}_k) - \mathbf{f}(\mathbf{x}^*) \leq \frac{v}{(k+1)^2}$

Convergent FISTA: Convergence of $(\mathbf{x}_k)_{k \in \mathbb{N}}$

Let $\alpha > 2$ and let $(\forall k \in \mathbb{N}) \quad \theta_k = \frac{k+\alpha}{\alpha}$.

Let $(\mathbf{x}_k)_{k \in \mathbb{N}}$ be a sequence generated by FISTA, using the $(\theta_k)_{k \in \mathbb{N}}$ defined above. Then the sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a minimiser of f .

Proof: A complete proof is provided in [Chambolle and Dossal, 2015].

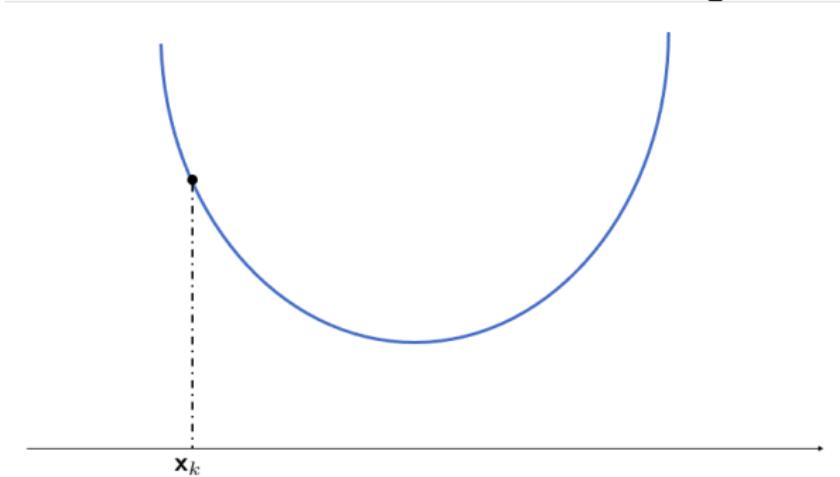
Backtracking: FISTA with
backtracking

Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

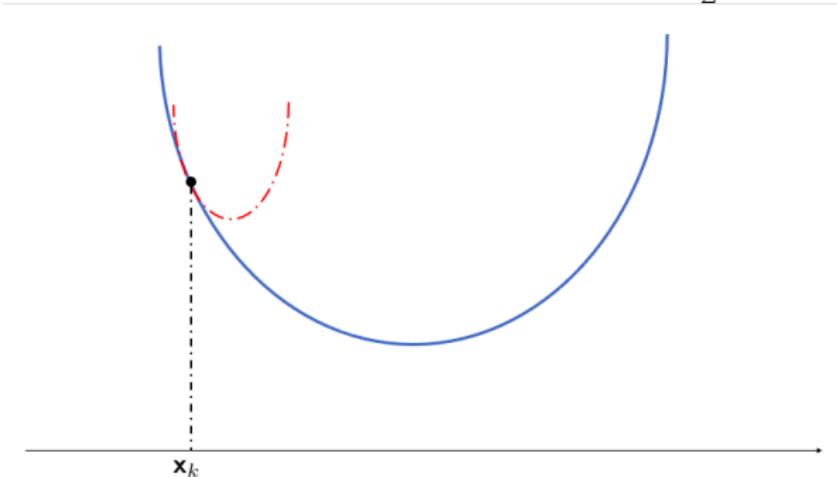


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

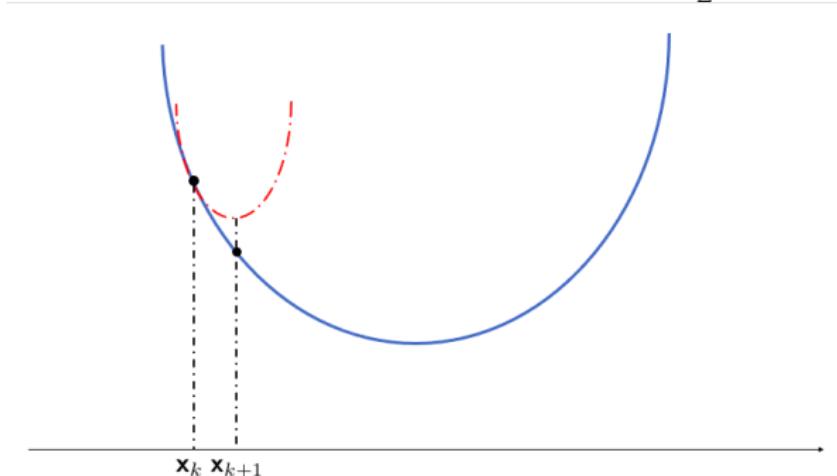


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

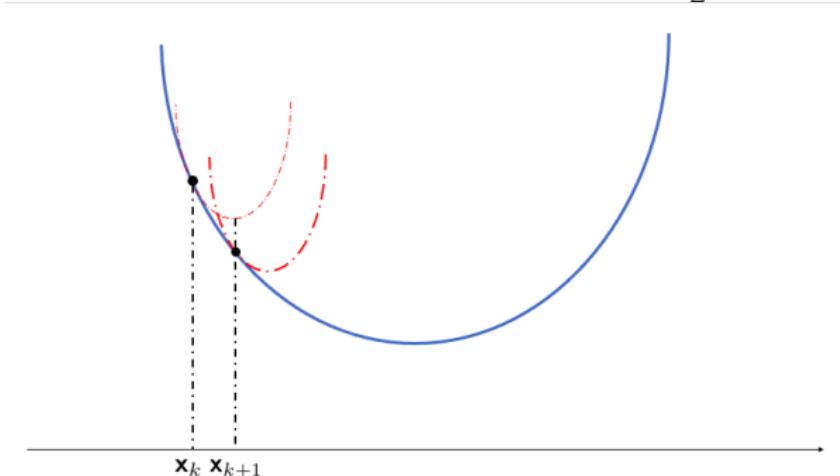


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

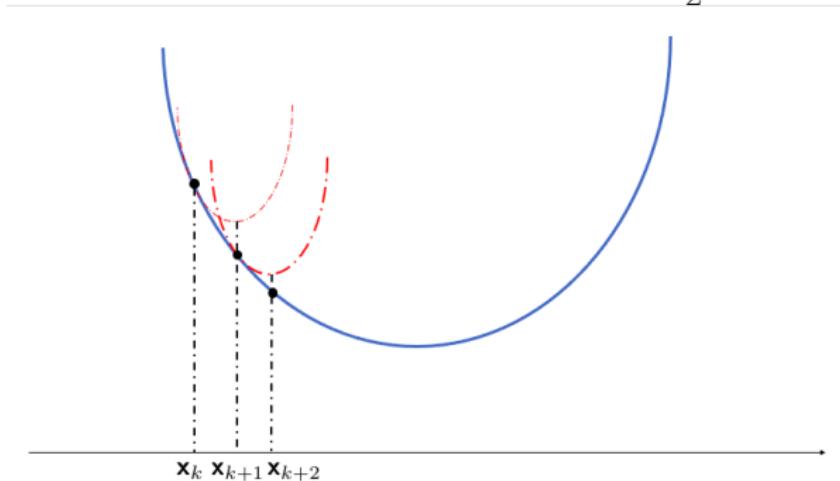


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

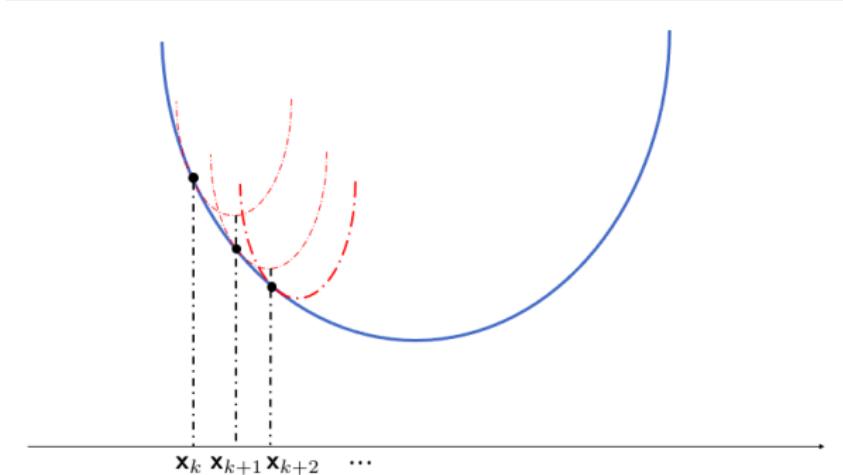


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

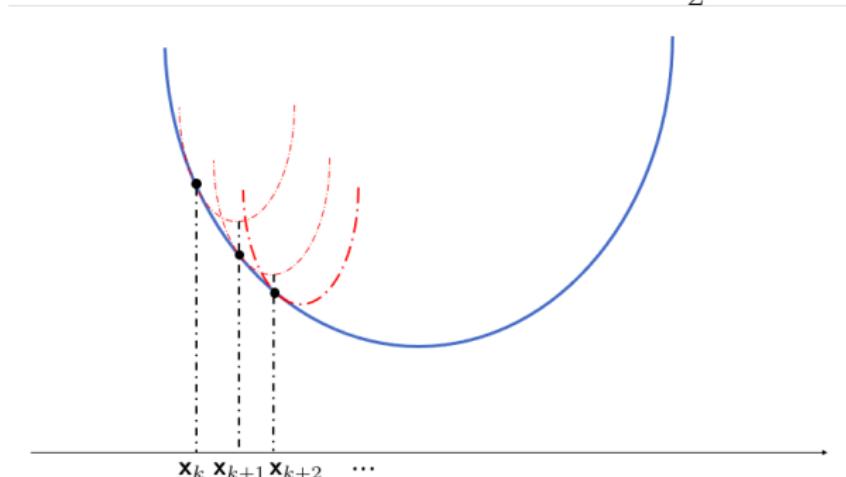


Case when β is unknown: Backtracking (1/2)

REMARK: Link between FB iterations and Majorisation-Minimisation methods:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



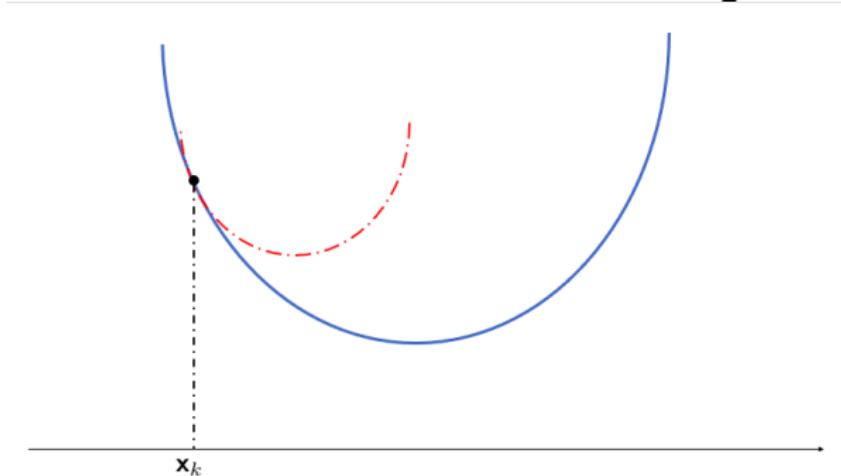
IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that local quadratic majorisation of f is satisfied.

Case when β is unknown: Backtracking (2/2)

IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that **local quadratic majorisation of f** is satisfied.

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta_k}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

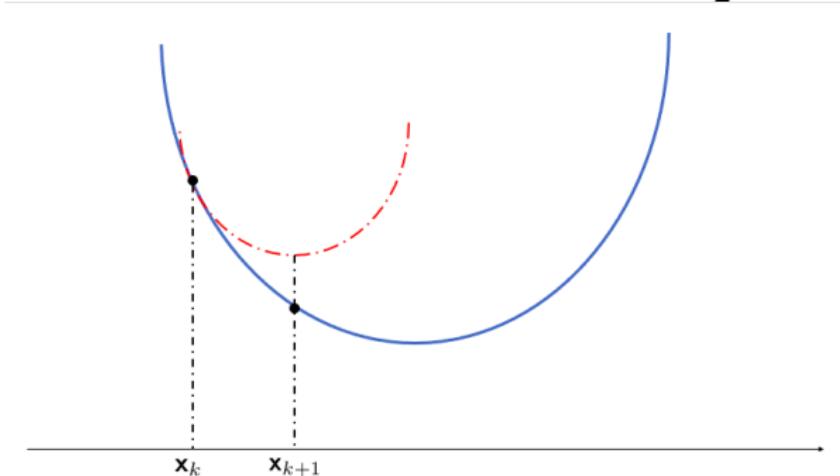


Case when β is unknown: Backtracking (2/2)

IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that **local quadratic majorisation of f** is satisfied.

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta_k}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

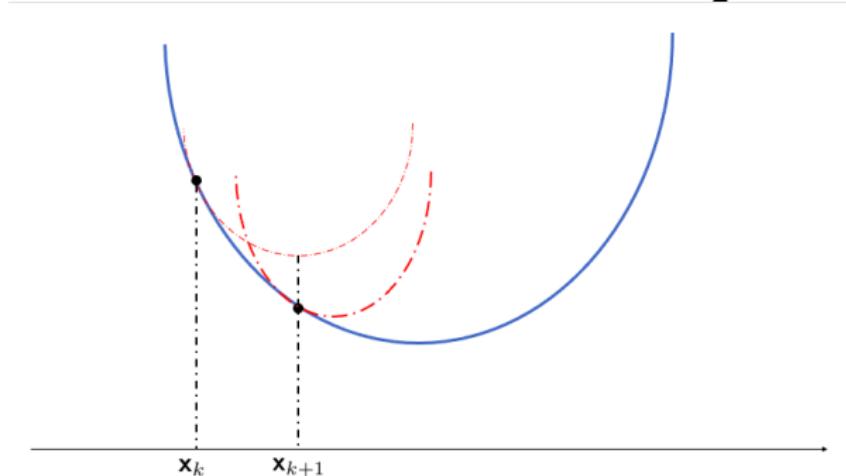


Case when β is unknown: Backtracking (2/2)

IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that **local quadratic majorisation of f** is satisfied.

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta_k}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

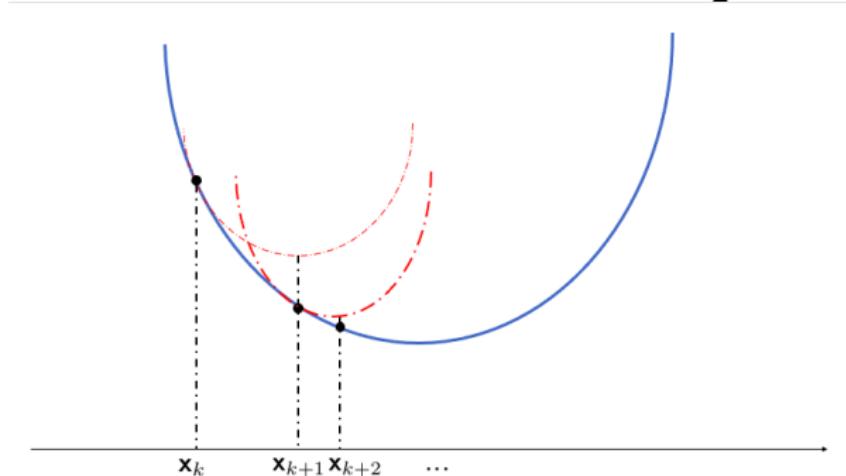


Case when β is unknown: Backtracking (2/2)

IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that **local quadratic majorisation of f** is satisfied.

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q_k(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta_k}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



FISTA with Backtracking

Let $\bar{\beta} > 0$, and $\mathbf{x} \in \mathbb{R}^N$. Define $p_{\bar{\beta}}(\mathbf{x}) = \text{prox}_{\bar{\beta}^{-1}g}(\mathbf{x} - \bar{\beta}^{-1}\nabla h(\mathbf{x}))$
and $(\forall \mathbf{x}^+ \in \mathbb{R}^N) \quad q_{\bar{\beta}}(\mathbf{x}^+, \mathbf{x}) = h(\mathbf{x}) + \langle \mathbf{x}^+ - \mathbf{x} \mid \nabla h(\mathbf{x}) \rangle + \frac{\bar{\beta}}{2} \|\mathbf{x}^+ - \mathbf{x}\|^2 + g(\mathbf{x}^+)$

FISTA with backtracking

Let $\mathbf{x}_0 \in \text{dom } g$ and $\mathbf{y}_0 = \mathbf{x}_0$. Let $\theta_0 = 1$. Let $\beta_0 > 0$ and $\eta > 1$.

For $k = 0, 1, \dots$

Find the smallest $i_k \in \mathbb{N}$ such that $\begin{cases} f(p_{\bar{\beta}}(\mathbf{y}_k)) \leq q_{\bar{\beta}}(p_{\bar{\beta}}(\mathbf{y}_k), \mathbf{y}_k) \\ \bar{\beta} = \eta^{i_k} \beta_k \end{cases}$

Set $\beta_{k+1} = \eta^{i_k} \beta_k$ and $\mathbf{x}_{k+1} = p_{\beta_{k+1}}(\mathbf{y}_k)$

$$\theta_{k+1} = \frac{1 + \sqrt{1 + 4\theta_k^2}}{2}$$

$$\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}} (\mathbf{x}_{k+1} - \mathbf{x}_k)$$

FISTA with Backtracking: How it works

Let $\bar{\beta} > 0$, and $\mathbf{x} \in \mathbb{R}^N$. Define $p_{\bar{\beta}}(\mathbf{x}) = \text{prox}_{\bar{\beta}^{-1}g}(\mathbf{x} - \bar{\beta}^{-1}\nabla h(\mathbf{x}))$
and $(\forall \mathbf{x}^+ \in \mathbb{R}^N) \quad q_{\bar{\beta}}(\mathbf{x}^+, \mathbf{x}) = h(\mathbf{x}) + \langle \mathbf{x}^+ - \mathbf{x} \mid \nabla h(\mathbf{x}) \rangle + \frac{\bar{\beta}}{2} \|\mathbf{x}^+ - \mathbf{x}\|^2 + g(\mathbf{x}^+)$

At each iteration $k \in \mathbb{N}$, start with $i_k = 0$ and:

- [Step 1]

Fix $\bar{\beta} = \eta^{i_k} \beta_k$ and compute $p_{\bar{\beta}}(\mathbf{y}_k) = \text{prox}_{\bar{\beta}^{-1}g}(\mathbf{y}_k - \bar{\beta}^{-1}\nabla h(\mathbf{y}_k))$

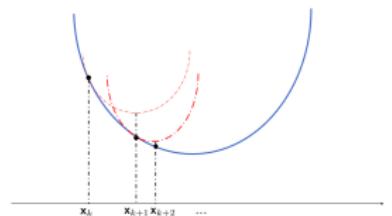
- [Condition to move to Step 2]

If $f(p_{\bar{\beta}}(\mathbf{y}_k)) \leq q_{\bar{\beta}}(p_{\bar{\beta}}(\mathbf{y}_k), \mathbf{y}_k)$, then go to [Step 2].
Otherwise go back to [Step 1] with $i_k = i_k + 1$.

- [Step 2]

Update $\beta_{k+1} = \eta^{i_k} \beta_k$, $\mathbf{x}_{k+1} = p_{\beta_{k+1}}(\mathbf{y}_k)$.

Compute FISTA inertia step.



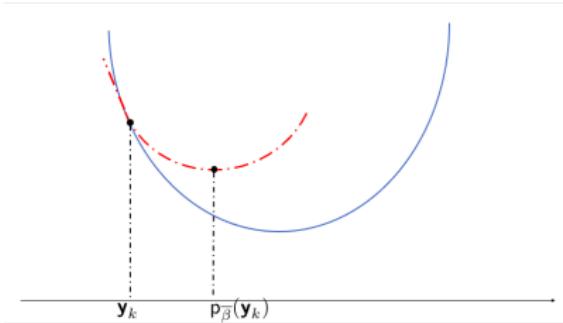
FISTA with Backtracking: Remarks

- The approximations $q_{\bar{\beta}}(\cdot, \mathbf{y}_k)$ of f at \mathbf{y}_k do not need to be a majorant function of f on \mathbb{R}^N .

We only need to satisfy the majorant condition at $p_{\bar{\beta}}(\mathbf{y}_k)$:

$$f(p_{\bar{\beta}}(\mathbf{y}_k)) \leq q_{\bar{\beta}}(p_{\bar{\beta}}(\mathbf{y}_k), \mathbf{y}_k)$$

Allows larger step-sizes than $1/\beta$



- Backtracking strategy can also be used with FB

FB acceleration: Preconditioning
and variable metric

FB as majorisation-minimisation algorithm

OBJECTIVE: Find a minimiser of $f = h + g$

$$\begin{aligned}\mathbf{x}_{k+1} &= \text{prox}_{\gamma_k g} (\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)) \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} g(\mathbf{x}) + \frac{1}{2\gamma_k} \|\mathbf{x} - \mathbf{x}_k + \gamma_k \nabla h(\mathbf{x}_k)\|^2 \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} g(\mathbf{x}) + \frac{1}{2\gamma_k} \|\mathbf{x} - \mathbf{x}_k\|^2 + \langle \mathbf{x} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\gamma_k}{2} \|\nabla h(\mathbf{x}_k)\|^2 \\ &= \underset{\mathbf{x}}{\operatorname{argmin}} g(\mathbf{x}) + h(\mathbf{x}_k) + \langle \mathbf{x} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \underbrace{\frac{1}{2\gamma_k} \|\mathbf{x} - \mathbf{x}_k\|^2}_{:=q(\mathbf{x}, \mathbf{x}_k)}\end{aligned}$$

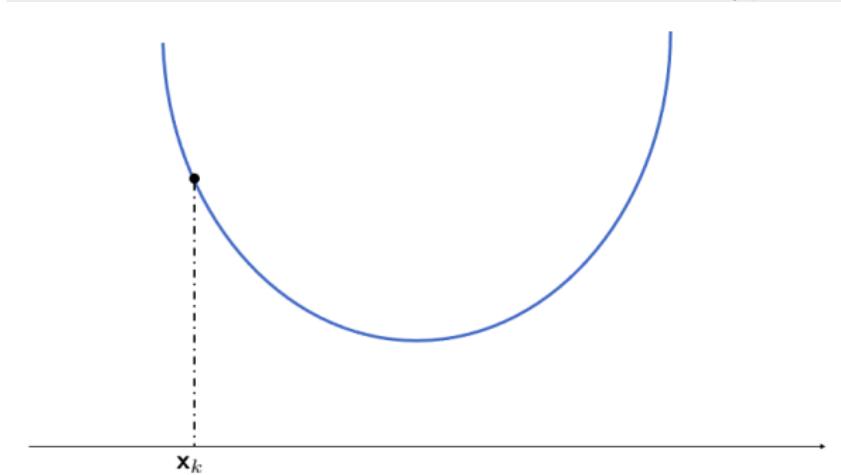
- $\mathbf{x}_{k+1} = \underset{\mathbf{x}}{\operatorname{argmin}} g(\mathbf{x}) + q(\mathbf{x}, \mathbf{x}_k)$
- $(\forall \mathbf{x} \in \mathbb{R}^N) f(\mathbf{x}) \leq g(\mathbf{x}) + q(\mathbf{x}, \mathbf{x}_k)$ for $\gamma_k \in]0, 1/\beta[$
- $f(\mathbf{x}_k) = g(\mathbf{x}_k) + q(\mathbf{x}_k, \mathbf{x}_k)$

From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

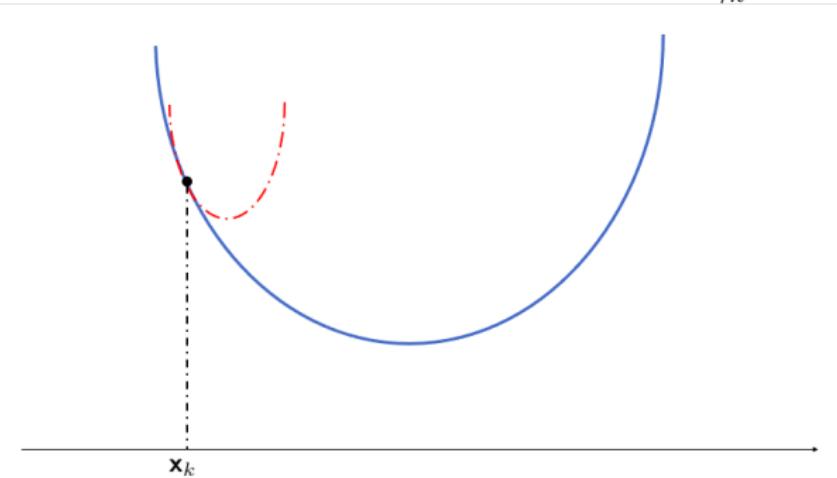


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

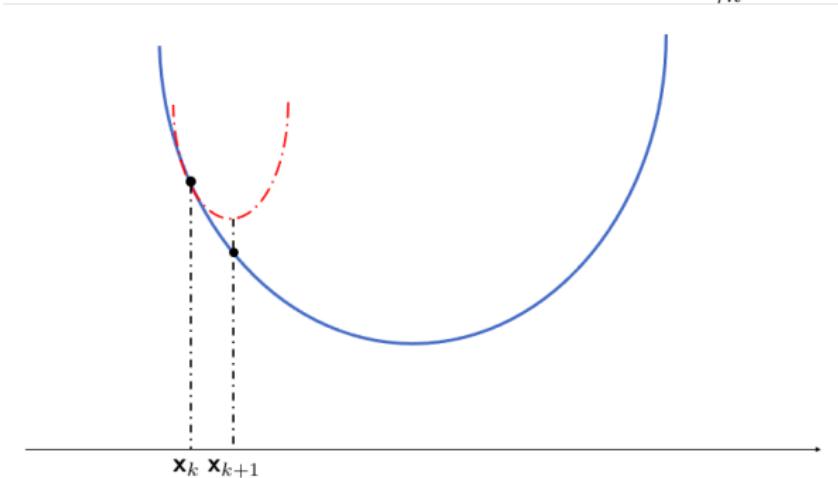


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

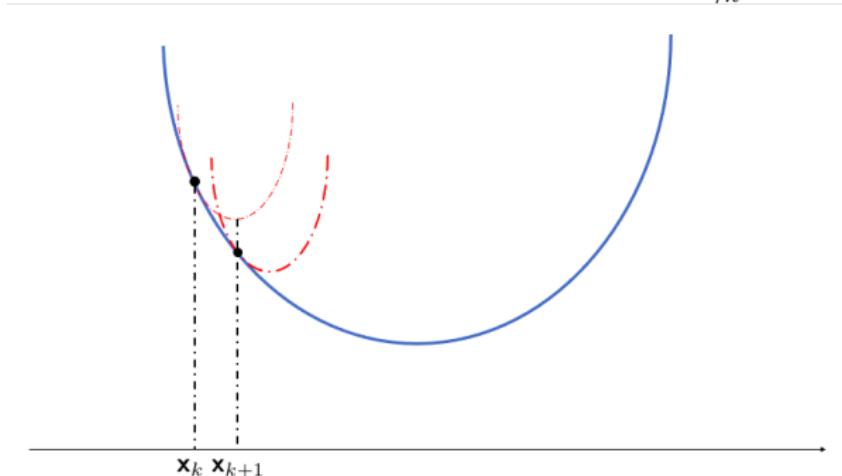


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

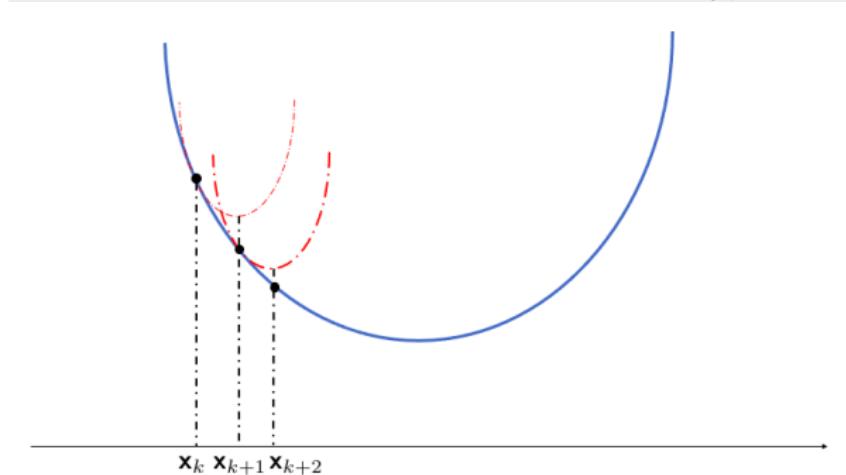


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

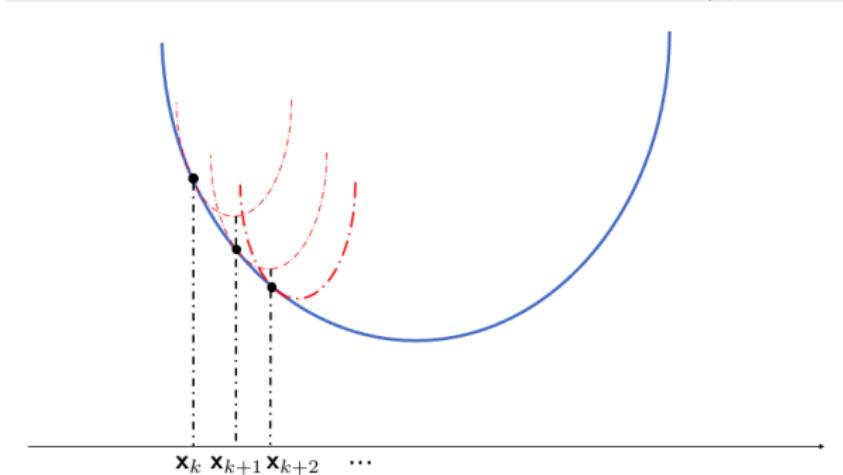


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$

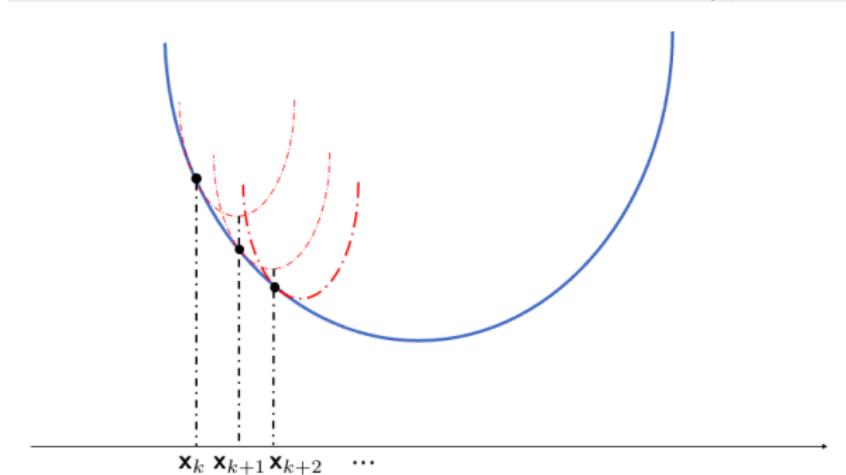


From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



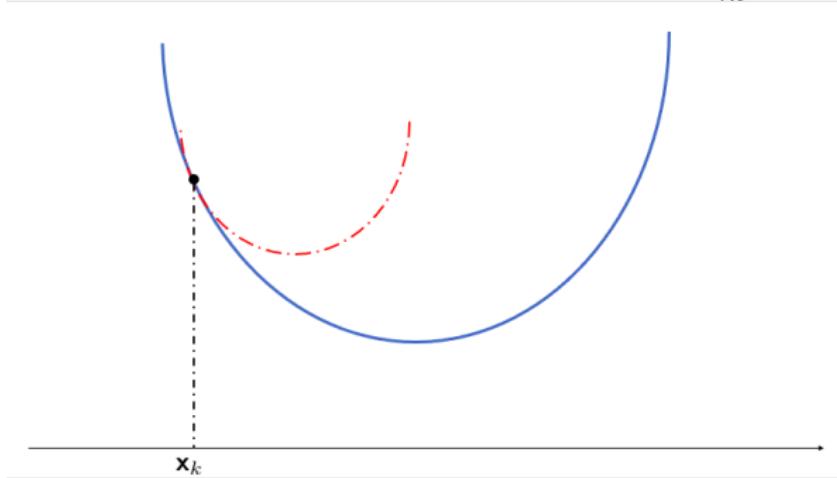
IDEA: Do not use the Lipschitz constant β to determine the step-size γ , but verify that local quadratic majorisation of f is satisfied.

... to Variable Metric FB

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} and $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ SPD such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{A}_k}^2 + g(\mathbf{x}_{k+1})$$



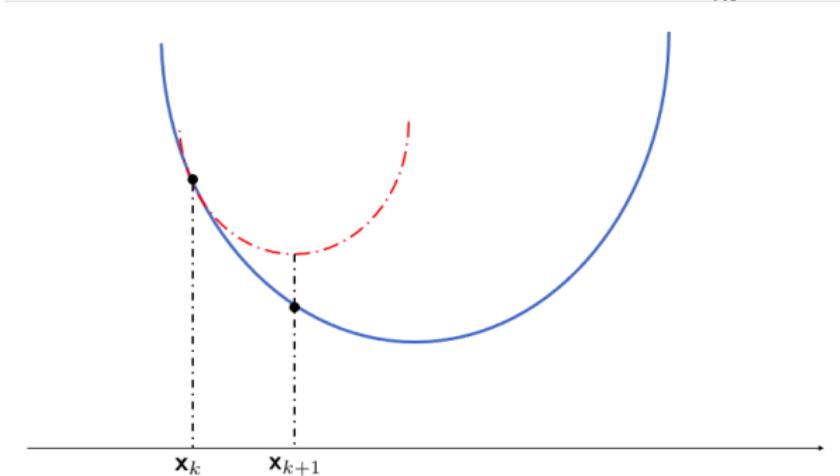
- ($\forall \mathbf{x} \in \mathbb{R}^N$) $\|\mathbf{x}\|_{\mathbf{A}_k}^2 = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \rangle$

... to Variable Metric FB

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} and $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ SPD such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{A}_k}^2 + g(\mathbf{x}_{k+1})$$



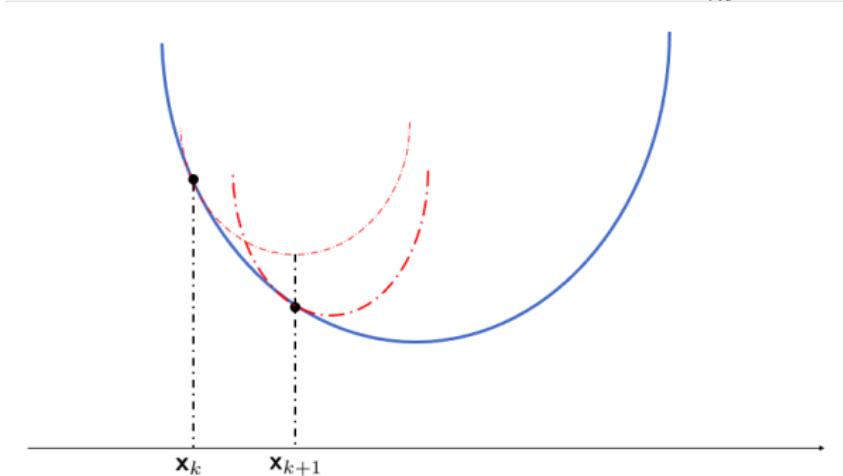
- ($\forall \mathbf{x} \in \mathbb{R}^N$) $\|\mathbf{x}\|_{\mathbf{A}_k}^2 = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \rangle$

... to Variable Metric FB

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} and $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ SPD such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{A}_k}^2 + g(\mathbf{x}_{k+1})$$



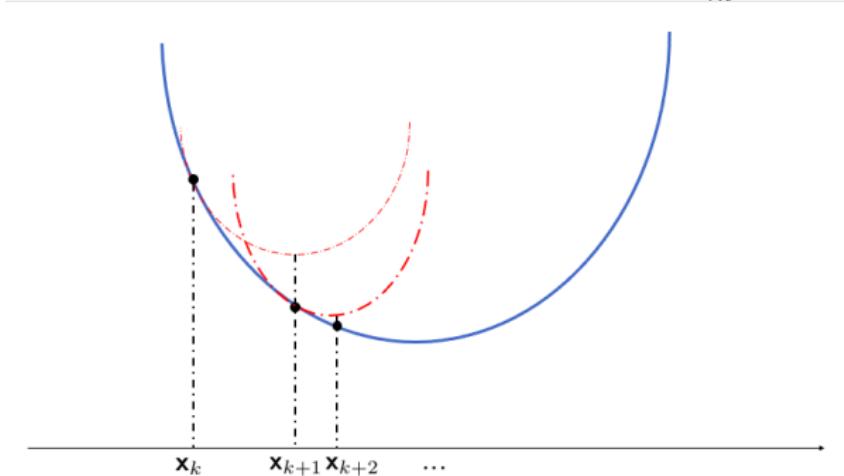
- ($\forall \mathbf{x} \in \mathbb{R}^N$) $\|\mathbf{x}\|_{\mathbf{A}_k}^2 = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \rangle$

... to Variable Metric FB

REMARK: FB iterations as Majorisation-Minimisation scheme:

Define \mathbf{x}_{k+1} and $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ SPD such that

$$f(\mathbf{x}_{k+1}) \leq q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|_{\mathbf{A}_k}^2 + g(\mathbf{x}_{k+1})$$



- ($\forall \mathbf{x} \in \mathbb{R}^N$) $\|\mathbf{x}\|_{\mathbf{A}_k}^2 = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x} \rangle$

Variable Metric FB

OBJECTIVE: Find a minimiser of $f = h + g$

Let $\mathbf{x}_0 \in \text{dom } g$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2[$

For $k = 0, 1, \dots$

$$\lfloor \mathbf{x}_{k+1} = \text{prox}_{\gamma_k g}^{\mathbf{A}_k} (\mathbf{x}_k - \gamma_k \mathbf{A}_k^{-1} \nabla h(\mathbf{x}_k))$$

- $\mathbf{A}_k \in \mathbb{R}^{N \times N}$ is symmetric positive definite such that,
 $(\exists (\underline{\nu}, \bar{\nu}) \in]0, +\infty[^2) (\forall k \in \mathbb{N}) (\forall \mathbf{x} \in \mathbb{R}^N) \quad \underline{\nu} \|\mathbf{x}\|^2 \leq \mathbf{x}^\top \mathbf{A}_k \mathbf{x} \leq \bar{\nu} \|\mathbf{x}\|^2$
- $(\forall \tilde{\mathbf{x}} \in \mathbb{R}^N) \quad \text{prox}_{\gamma_k g}^{\mathbf{A}_k}(\tilde{\mathbf{x}}) = \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{argmin}} \gamma_k g(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \tilde{\mathbf{x}}\|_{\mathbf{A}_k}^2$

- $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a minimiser \mathbf{x}^* of f
- $(f(\mathbf{x}_k))_{k \in \mathbb{N}}$ is a non-increasing sequence converging to $f(\mathbf{x}^*)$.

FB & Non-convex problems

OBJECTIVE: Find a minimiser of $f = h + g$

QUESTION: Convergence when h and/or g are not convex?

EXAMPLES: NON CONVEX DATA FIDELITY ($F \equiv h$ differentiable)

- *Phase reconstruction:* $\mathbf{z} = |\Phi\bar{\mathbf{x}}| + \mathbf{w}$

$$h(\mathbf{x}) = \frac{1}{2}$$

- *Signal-dependent Gaussian noise:* $\mathbf{z} = \Phi\bar{\mathbf{x}} + \mathbf{w}$ with

$$\begin{cases} \mathbf{w}^{(m)} \sim \mathcal{N}(0, \sigma^{(m)}) \\ \sigma^{(m)} = \sqrt{a^{(m)}[\Phi\bar{\mathbf{x}}]^{(m)} + b^{(m)}} \end{cases}$$

$$h(\mathbf{x}) = \sum_{m=1}^M h_m([\Phi\mathbf{x}]^{(m)}) \text{ with } h_m(u) = \frac{1}{2} \frac{(u - z^{(m)})^2}{a^{(m)}u + b^{(m)}} + \frac{1}{2} \log(a^{(m)}u + b^{(m)})$$

FB & Non-convex problems

OBJECTIVE: Find a minimiser of $f = h + g$

QUESTION: Convergence when h and/or g are not convex?

EXAMPLES: NON CONVEX DATA FIDELITY ($F \equiv h$ differentiable)

- *Phase reconstruction:* $\mathbf{z} = |\Phi\bar{\mathbf{x}}| + \mathbf{w}$

$$h(\mathbf{x}) = \frac{1}{2}$$

- *Signal-dependent Gaussian noise:* $\mathbf{z} = \Phi\bar{\mathbf{x}} + \mathbf{w}$ with

$$\begin{cases} \mathbf{w}^{(m)} \sim \mathcal{N}(0, \sigma^{(m)}) \\ \sigma^{(m)} = \sqrt{a^{(m)}[\Phi\bar{\mathbf{x}}]^{(m)} + b^{(m)}} \end{cases}$$

$$h(\mathbf{x}) = \sum_{m=1}^M h_m([\Phi\mathbf{x}]^{(m)}) \text{ with } h_m(\mathbf{u}) = \frac{1}{2} \frac{(\mathbf{u} - \mathbf{z}^{(m)})^2}{a^{(m)}\mathbf{u} + b^{(m)}} + \frac{1}{2} \log(a^{(m)}\mathbf{u} + b^{(m)})$$

- *(Semi) Blind deconvolution:* $\mathbf{z} = \Phi\bar{\mathbf{x}} + \mathbf{w}$

$$h(\mathbf{x}, \Phi) = \frac{1}{2} \|\Phi\mathbf{x} - \mathbf{z}\|^2$$



Necessitate block-coordinate approaches... (see Part 2b)

FB & Non-convex problems

OBJECTIVE: Find a minimiser of $f = h + g$

QUESTION: Convergence when h and/or g are not convex?

EXAMPLES: NON CONVEX REGULARISATION ($R \equiv g$ non differentiable)

- *ℓ_0 pseudo norm:* $g(\mathbf{x}) = \ell_0(\mathbf{x})$
- *log-sum penalisation:* $g(\mathbf{x}) = \sum_{n=1}^N \log(|\mathbf{x}^{(n)}| + \varepsilon)$
- *Cauchy penalisation:* $g(\mathbf{x}) = \sum_{n=1}^N \log(|\mathbf{x}^{(n)}|^2 + \varepsilon)$
- etc.

FB & Non-convex problems: Convergence

OBJECTIVE: Find a minimiser of $f = h + g$

QUESTION: Convergence when h and/or g are not convex?

- FB iterates $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converge to a critical point \mathbf{x}^* of f (i.e., $\mathbf{0} \in \partial f(\mathbf{x}^*)$)
 - if g non convex, $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 1\beta^{-1}[$
 - if g convex, $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2\beta^{-1}[$
- Same results for VMFB
- 👉 Proof based on **Kurdyka-Łojasiewicz inequality**

Kurdyka-Łojasiewicz inequality

Function f satisfies the **Kurdyka-Łojasiewicz inequality** on a bounded subset E of \mathbb{R}^N iff, $(\forall \xi \in \mathbb{R})(\exists(\kappa, \zeta, \theta) \in]0, +\infty[^2 \times [0, 1[)$ such that

$$(\forall \mathbf{t} \in \partial f(\mathbf{x})) \quad \|\mathbf{t}\| \geq \kappa |f(\mathbf{x}) - \xi|^\theta,$$

for every $\mathbf{x} \in E$ such that $|f(\mathbf{x}) - \xi| \leq \zeta$

- ★ Note that other forms of the KL inequality can be found in the literature
[Bolte *et al.*, 2007][Bolte *et al.*, 2010]
- ★ Satisfied for a wide class of functions :
 - real analytic functions
 - semi-algebraic functions
 - ...

Example: Image reconstruction [Chouzenoux *et al*, 2014]

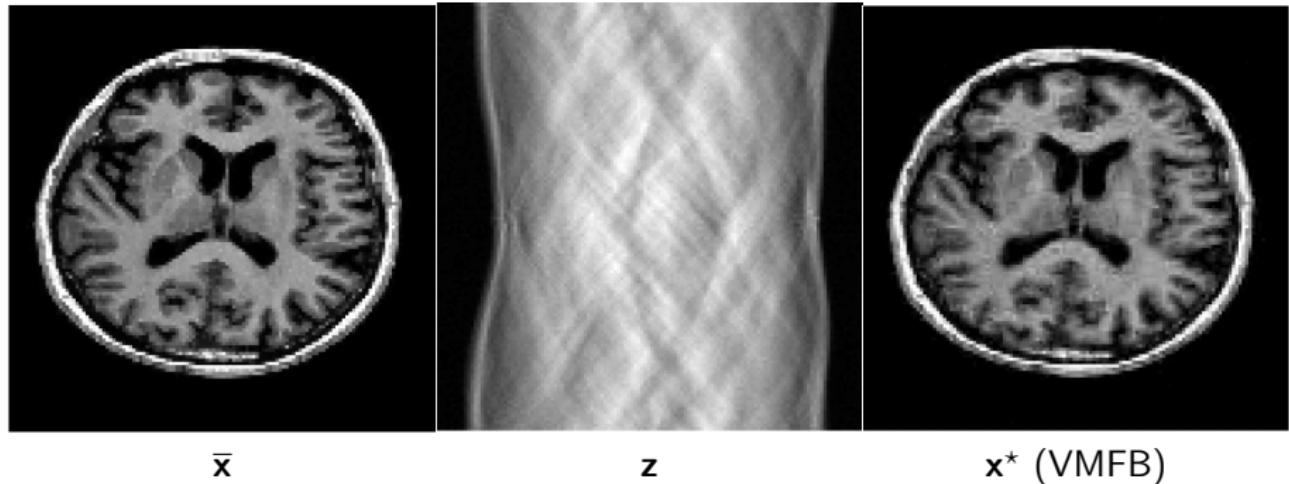
INVERSE PROBLEM: $\mathbf{z} = \Phi\bar{\mathbf{x}} + \mathbf{w}$

- **Measurement operator:** Φ is a Radon matrix modelling $M = 16384$ parallel projections (128 acquisition lines and 128 angles)
- **Signal-dependent Gaussian noise (approximation of Gauss-Poisson):**
 $(\forall m \in \{1, \dots, M\}) \mathbf{w}^{(m)} \sim \mathcal{N}(0, \sigma^{(m)})$ with $\sigma^{(m)} = \sqrt{a^{(m)}[\Phi\bar{\mathbf{x}}]^{(m)} + b^{(m)}}$

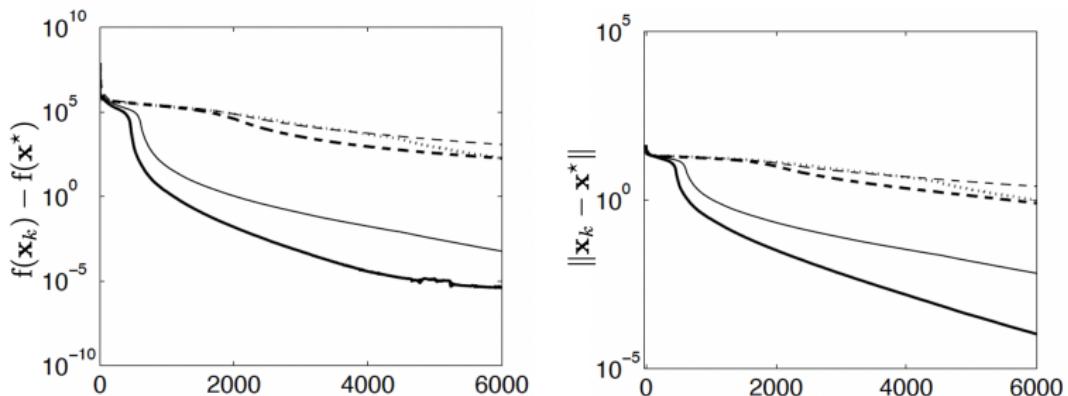
MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} F(\mathbf{x}) + R(\mathbf{x})$

- **Data-fidelity:** $F(\mathbf{x}) = \sum_{m=1}^M h_m([\Phi\mathbf{x}]^{(m)})$ with $h_m(\mathbf{u}) = \frac{1}{2} \frac{(\mathbf{u} - \mathbf{z}^{(m)})^2}{a^{(m)}\mathbf{u} + b^{(m)}} + \frac{1}{2} \log(a^{(m)}\mathbf{u} + b^{(m)})$
~~~ Lipschitz-differentiable
- **Regularisation:**  $R(\mathbf{x}) = \eta \|\Psi\mathbf{x}\|_1 + \iota_{[0,+\infty]^2}(\mathbf{x})$  with  $\eta > 0$ , and  $\Psi$  redundant (undecimated) wavelet (DB8 over three resolution levels)  
~~~  $\operatorname{prox}_R$  computed with sub-iterations (dual FB algorithm)

Example: Image reconstruction [Chouzenoux *et al*, 2014]



Example: Image reconstruction [Chouzenoux *et al*, 2014]



- *Dashed:* FB with $\gamma_k = 1.9\beta^{-1}$ (**bold**) and $\gamma_k = \beta^{-1}$ (**thin**)
- *Dotted:* FISTA with $\gamma_k = \beta^{-1}$ (**thin**)
- *Solid:* VMFB with $\gamma_k = 1.9\beta^{-1}$ (**bold**) and $\gamma_k = \beta^{-1}$ (**thin**)

Primal-Dual methods

Minimization problem

Find $\mathbf{x}^* \in \text{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{Lx})$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable
- $f \in \Gamma_0(\mathbb{R}^N)$
- $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

EXAMPLE: $\mathbf{z} = \Phi \bar{\mathbf{x}} + \mathbf{w}$

- $\Phi: \mathbb{R}^N \rightarrow \mathbb{R}^M$ measurement operator
- $\mathbf{w} \in \mathbb{R}^M$ realisation of white Gaussian noise

$$\underset{\mathbf{x} \in \mathbb{R}^N}{\text{minimise}} \underbrace{\frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2}_{h(\mathbf{x})} + \underbrace{\iota_C(\mathbf{x})}_{f(\mathbf{x})} + \underbrace{\eta \|\Psi \mathbf{x}\|_1}_{g(\mathbf{Lx})}$$

- C convex, closed, non-empty subset of \mathbb{R}^N (e.g. $[0, +\infty]$)
- $\eta > 0$ is a regularisation parameter
- $\Psi: \mathbb{R}^N \rightarrow \mathbb{R}^L$ sparsifying operator (e.g. wavelet transform or TV)

Minimization problem

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{Lx})$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable
- $f \in \Gamma_0(\mathbb{R}^N)$
- $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

Use FB algorithm?

For $k = 0, 1, \dots$

$$\left[\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k(f+g \circ \mathbf{L})} (\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)) \right]$$

► **QUESTION:** How to compute $\operatorname{prox}_{\gamma_k(f+g \circ \mathbf{L})}$?

► Use primal-dual methods

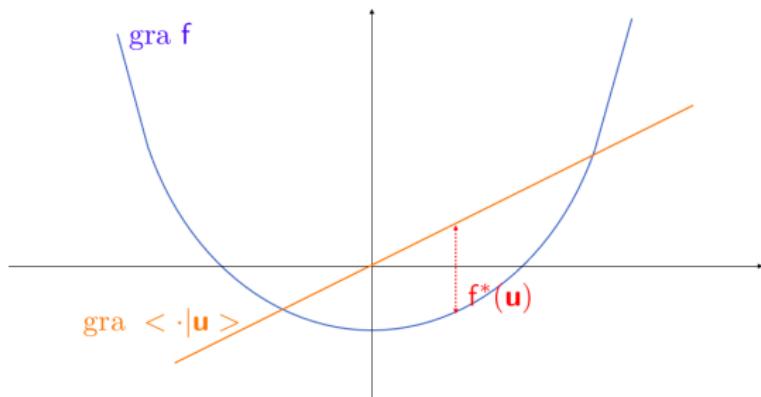
Fenchel-Legendre duality

Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

Graphical illustration: $f^*(\mathbf{u})$ is the supremum of the signed vertical distance between the graph of f and that of the continuous linear functional $\langle \cdot | \mathbf{u} \rangle$

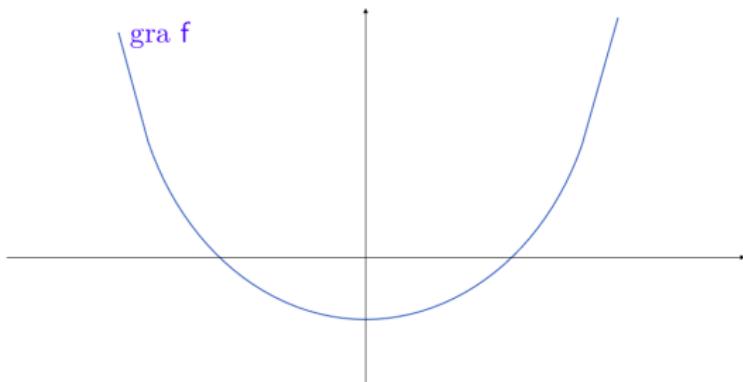


Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

Graphical illustration: Second interpretation - How to build $f^*(\mathbf{u})$?

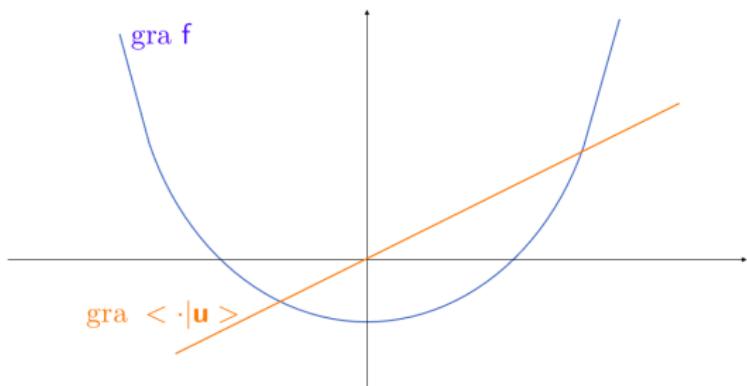


Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

Graphical illustration: Second interpretation - How to build $f^*(\mathbf{u})$?

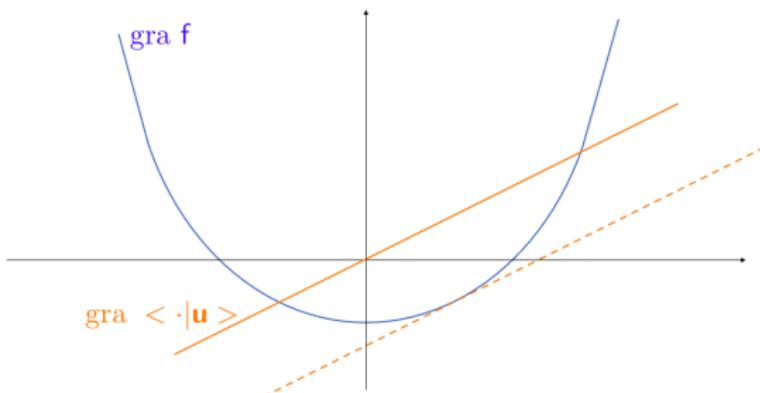


Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

Graphical illustration: Second interpretation - How to build $f^*(\mathbf{u})$?

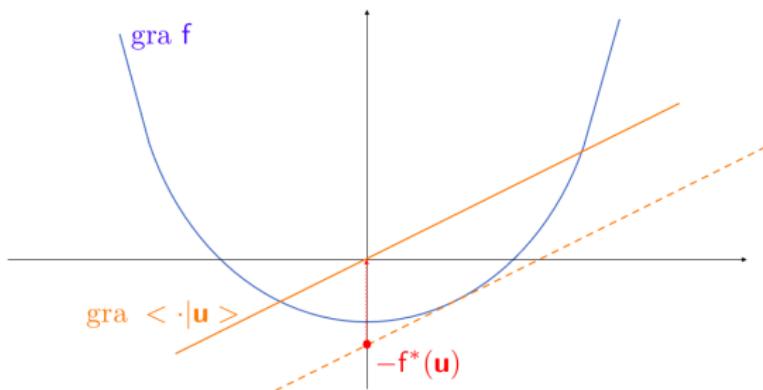


Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

Graphical illustration: Second interpretation - How to build $f^*(\mathbf{u})$?



Conjugate function

The **conjugate** of a function $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ is the function f^* defined as

$$\begin{aligned} f^*: \quad \mathbb{R}^N &\rightarrow [-\infty, +\infty] \\ \mathbf{u} &\mapsto \sup_{\mathbf{x} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) \end{aligned}$$

REMARK:

- The function f is proper (i.e. $\text{dom } f \neq \emptyset$) iff $(\forall \mathbf{u} \in \mathbb{R}^N) \quad f^*(\mathbf{u}) \neq -\infty$

Proof: Assume that $(\exists \mathbf{u} \in \mathbb{R}^N)$ such that $f^*(\mathbf{u}) = -\infty$.

So, we have $\sup_{\mathbf{x}} \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}) = -\infty$.

This happens iff $(\forall \mathbf{x} \in \mathbb{R}^N) f(\mathbf{x}) = +\infty$, i.e. $\text{dom } f = \emptyset$. □

Some properties

Fenchel-Young Inequality

Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a proper function. Then

$$(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2) \quad f(\mathbf{x}) + f^*(\mathbf{u}) \geq \langle \mathbf{x} | \mathbf{u} \rangle$$

Proof: This is a direct application of the definition of f^* :

$$(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2) \quad f^*(\mathbf{u}) \geq \langle \mathbf{x} | \mathbf{u} \rangle - f(\mathbf{x}).$$



Some properties

Fenchel-Young Inequality

Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a proper function. Then

$$(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2) \quad f(\mathbf{x}) + f^*(\mathbf{u}) \geq \langle \mathbf{x} | \mathbf{u} \rangle$$

- Let $f^{**} = (f^*)^*$ be the **biconjugate** of f . Then we have $f^{**} \leq f$.

Proof: We can assume that f is proper (otherwise the inequality trivially holds).

According to Fenchel-Young inequality, for every $\mathbf{x} \in \mathbb{R}^N$ we have

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \langle \mathbf{x} | \mathbf{u} \rangle - f^*(\mathbf{u}) \leq f(\mathbf{x})$$

$$\text{In particular, } \sup_{\mathbf{u} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f^*(\mathbf{u}) \leq f(\mathbf{x})$$

The result is obtained by noticing that $f^{**}(\mathbf{x}) = \sup_{\mathbf{u} \in \mathbb{R}^N} \langle \mathbf{x} | \mathbf{u} \rangle - f^*(\mathbf{u})$. □

Some properties

Fenchel-Young Inequality

Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a proper function. Then

$$(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2) \quad f(\mathbf{x}) + f^*(\mathbf{u}) \geq \langle \mathbf{x} | \mathbf{u} \rangle$$

- Let $f^{**} = (f^*)^*$ be the **biconjugate** of f . Then we have $f^{**} \leq f$.
- Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ and $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ such that $f \leq g$. We have $f^* \geq g^*$ and $f^{**} \leq g^{**}$.
- $f^{***} = f^*$

Proof: We have $f^{***} = (f^*)^{**} \leq f^*$.

In addition, since $f^{**} \leq f$, using second property we have $(f^{**})^* \geq f^*$.

Hence the equality. □

Some properties

Fenchel-Young Inequality

Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a proper function. Then

$$(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2) \quad f(\mathbf{x}) + f^*(\mathbf{u}) \geq \langle \mathbf{x} | \mathbf{u} \rangle$$

- Let $f^{**} = (f^*)^*$ be the **biconjugate** of f . Then we have $f^{**} \leq f$.
- Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ and $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ such that $f \leq g$. We have $f^* \geq g^*$ and $f^{**} \leq g^{**}$.
- $f^{***} = f^*$

Moreau-Fenchel Theorem

Let $f: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ be a proper function.

$$f \text{ is l.s.c. and convex} \Leftrightarrow f^{**} = f$$

Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

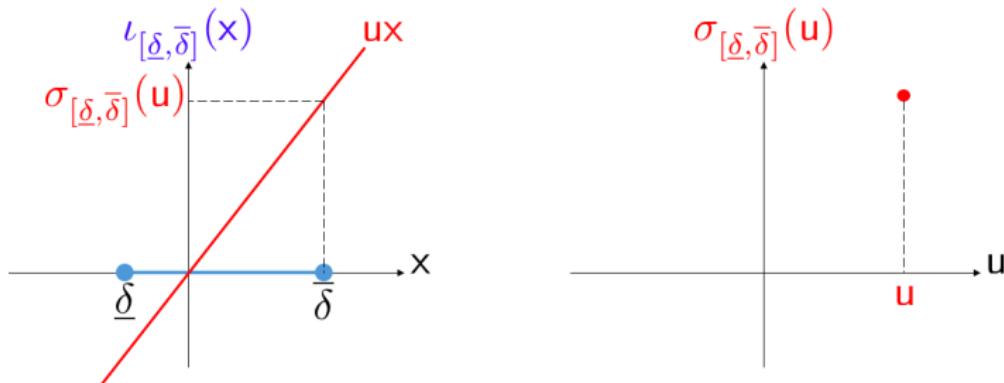
$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} \mid \mathbf{u} \rangle$$

Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

EXAMPLE: $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$

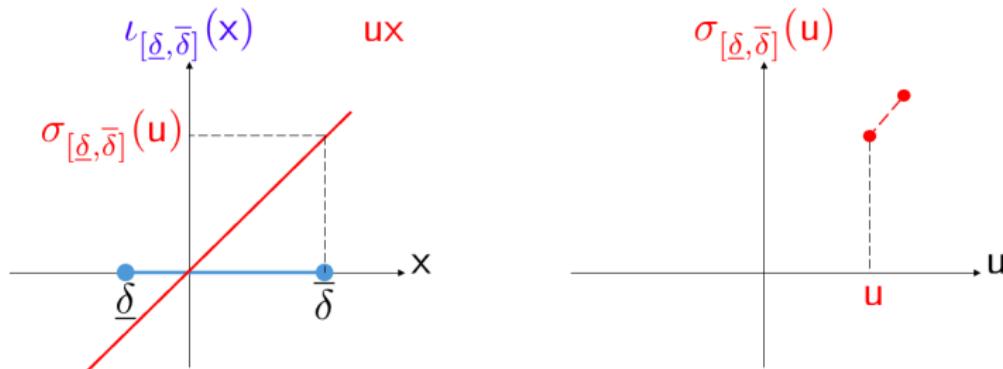


Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

EXAMPLE: $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$

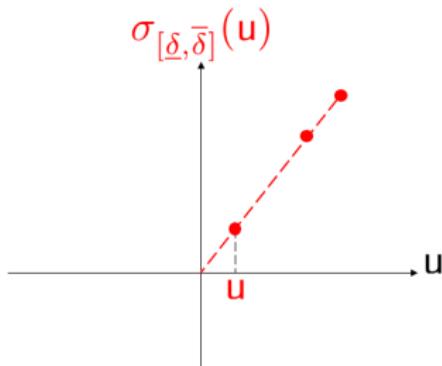
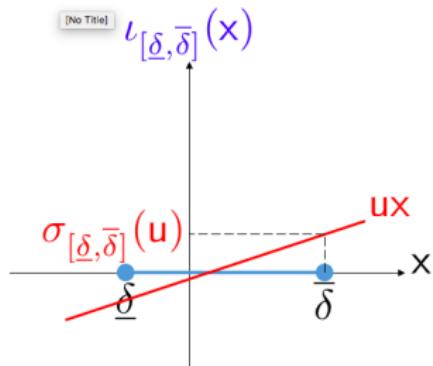


Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

EXAMPLE: $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$

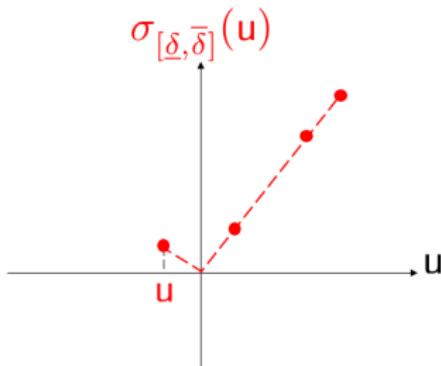
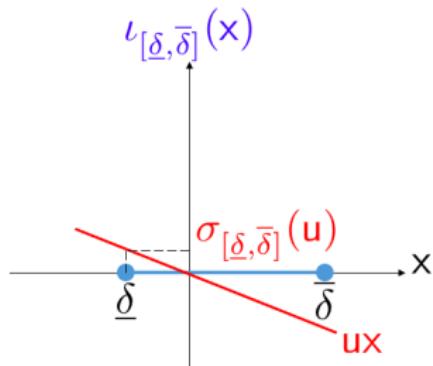


Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

EXAMPLE: $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$

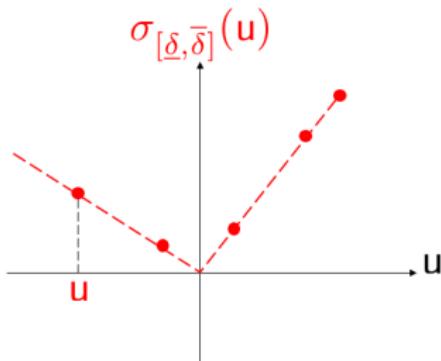
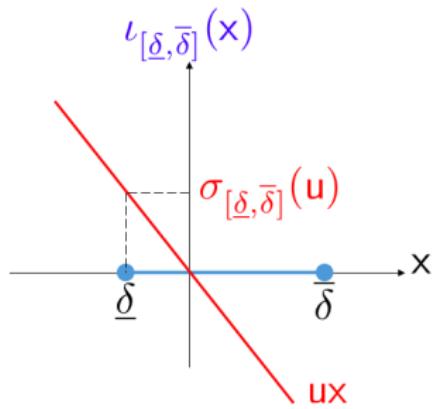


Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

EXAMPLE: $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$



Example: Support function

Let \mathcal{C} be a subset of \mathbb{R}^N . The support function of \mathcal{C} , denoted by $\sigma_{\mathcal{C}}$, is

$$(\forall \mathbf{u} \in \mathbb{R}^N) \quad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x} \in \mathcal{C}} \langle \mathbf{x} | \mathbf{u} \rangle$$

REMARKS:

- We have $\sigma_{\mathcal{C}} = \iota_{\mathcal{C}}^*$
- If \mathcal{C} is a closed, convex, non-empty subset of \mathbb{R}^N , then $\sigma_{\mathcal{C}}^* = \iota_{\mathcal{C}}^{**} = \iota_{\mathcal{C}}$

- Let $-\infty \leq \underline{\delta} < \bar{\delta} \leq +\infty$, and $\mathcal{C} = [\underline{\delta}, \bar{\delta}]$. Then

$$(\forall x \in \mathbb{R}) \quad \sigma_{\mathcal{C}}(x) = \begin{cases} \underline{\delta}x & \text{if } x < 0 \\ 0 & \text{if } x = 0 \\ \bar{\delta}x & \text{if } x > 0 \end{cases}$$

As a consequence we have $(\forall \delta > 0)(\forall x \in \mathbb{R}) \quad f(x) = \delta|x| = \sigma_{[-\delta, +\delta]}(x)$

and $f^* = \iota_{[-\delta, +\delta]} = \iota_{B_\infty(0, \delta)}$

More generally, for $f = \delta \|\cdot\|_1$, we have $f^* = \iota_{B_\infty(0, \delta)}$

Example: norms

Let $p \in]1, +\infty[$. The conjugate function of the $f = \frac{1}{p}|\cdot|^p$ is given by

$$f^* = \frac{1}{q}|\cdot|^q \text{ with } q = \frac{p}{p-1}$$

Proof: We consider the case $u \geq 0$ (the case $u \leq 0$ can be handled analogously).

$$(\forall u \in [0, +\infty[) \quad f^*(u) = \sup_{x \in \mathbb{R}} ux - \frac{1}{p}|x|^p = \sup_{x \geq 0} ux - \frac{1}{p}x^p$$

The function $\varphi: x \in [0, +\infty[\mapsto ux - \frac{1}{p}x^p$ being concave, we can obtain its maximum using the Fermat's rule:

$$\partial\varphi(x) = 0 \iff u - x^{p-1} = 0 \iff x = u^{\frac{1}{p-1}}$$

$$\text{Hence, } f^*(u) = \sup_{x \geq 0} ux - \frac{1}{p}x^p = u^{1+\frac{1}{p-1}} - \frac{1}{p}u^{\frac{p}{p-1}} = (1 - \frac{1}{p})u^{\frac{p}{p-1}}.$$

By setting $q = \frac{p}{p-1}$, we obtain $f^*(u) = \frac{1}{q}|u|^q$.

□

Example: norms

Let $p \in]1, +\infty[$. The conjugate function of the $f = \frac{1}{p}|\cdot|^p$ is given by

$$f^* = \frac{1}{q}|\cdot|^q \text{ with } q = \frac{p}{p-1}$$

REMARKS:

- Let $f = \|\cdot\|_p^p$. The conjugate function of f is $f^* = \|\cdot\|_q^q$ with $q = \frac{p}{p-1}$.
- For $q = \frac{p}{p-1}$, the ℓ_q -norm is called the **dual norm** of ℓ_p .

Fenchel-Rockafellar duality

Let $f: \mathbb{R}^N \rightarrow]-\infty]$, $g: \mathbb{R}^M \rightarrow]-\infty, +\infty]$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$.

The **primal problem** associated with the composite function $f + g \circ \mathbf{L}$ is

$$\underset{\mathbf{x} \in \mathbb{R}^N}{\text{minimise}} \quad f(\mathbf{x}) + g(\mathbf{Lx})$$

and its **dual problem** is

$$\underset{\mathbf{u} \in \mathbb{R}^M}{\text{minimise}} \quad f^*(-\mathbf{L}^*\mathbf{u}) + g^*(\mathbf{u})$$

Let $\mathbf{x}^* \in \mathbb{R}^N$ be a solution to the primal problem, and $\mathbf{u}^* \in \mathbb{R}^M$ be a solution to the dual problem.

The **primal optimal value** is $\mu = f(\mathbf{x}^*) + g(\mathbf{Lx}^*)$.

The **dual optimal value** is $\mu^* = f^*(-\mathbf{L}^*\mathbf{u}^*) + g^*(\mathbf{u}^*)$.

The **duality gap** is $\Delta(f, g, \mathbf{L}) = \begin{cases} 0, & \text{if } \mu = -\mu^* \\ \mu + \mu^*, & \text{otherwise} \end{cases}$

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

Let $f: \mathbb{R}^N \rightarrow]-\infty]$ and $g: \mathbb{R}^M \rightarrow]-\infty, +\infty]$ be **proper** functions, and let $\mathbf{L} \in \mathbb{R}^{M \times N}$. We have $\mu \geq -\mu^*$.

Proof: According to Fenchel-Young inequality, for every $\mathbf{x} \in \mathbb{R}^N$ and $\mathbf{u} \in \mathbb{R}^M$, we have $f(\mathbf{x}) + f^*(-\mathbf{L}^* \mathbf{u}) \geq \langle \mathbf{x} | -\mathbf{L}^* \mathbf{u} \rangle$ and $g(\mathbf{L}\mathbf{x}) + g^*(\mathbf{u}) \geq \langle \mathbf{L}\mathbf{x} | \mathbf{u} \rangle$.

Hence $f(\mathbf{x}) + g(\mathbf{L}\mathbf{x}) + f^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u}) \geq \langle \mathbf{x} | -\mathbf{L}^* \mathbf{u} \rangle + \langle \mathbf{L}\mathbf{x} | \mathbf{u} \rangle = 0$. □

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

Let $f: \mathbb{R}^N \rightarrow]-\infty]$ and $g: \mathbb{R}^M \rightarrow]-\infty, +\infty]$ be **proper** functions, and let $\mathbf{L} \in \mathbb{R}^{M \times N}$. We have $\mu \geq -\mu^*$.

Strong duality

Let $f \in \Gamma_0(\mathbb{R}^N)$ and $g \in \Gamma_0(\mathbb{R}^M)$, and let $\mathbf{L} \in \mathbb{R}^{M \times N}$.
If $0 \in \text{int}(\text{dom } g - \mathbf{L}(\text{dom } f))$, then $\mu = -\mu^*$.

REMARKS:

- If $\text{int}(\text{dom } g) \cap \mathbf{L}(\text{dom } f) \neq \emptyset$ OR $(\text{dom } g) \cap \text{int}(\mathbf{L}(\text{dom } f)) \neq \emptyset$ then $0 \in \text{int}(\text{dom } g - \mathbf{L}(\text{dom } f))$ holds.
- In addition, if $\mathbf{u}^* \in \mathbb{R}^M$ is a solution to the dual problem such that f^* is differentiable at $-\mathbf{L}^* \mathbf{u}^*$, then
 - either the primal problem has no solution,
 - OR it has a **unique solution given by** $\mathbf{x}^* = \nabla f^*(-\mathbf{L}^* \mathbf{u}^*)$

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

Let $f: \mathbb{R}^N \rightarrow]-\infty]$ and $g: \mathbb{R}^M \rightarrow]-\infty, +\infty]$ be **proper** functions, and let $L \in \mathbb{R}^{M \times N}$. We have $\mu \geq -\mu^*$.

Strong duality

Let $f \in \Gamma_0(\mathbb{R}^N)$ and $g \in \Gamma_0(\mathbb{R}^M)$, and let $L \in \mathbb{R}^{M \times N}$.
If $0 \in \text{int}(\text{dom } g - L(\text{dom } f))$, then $\mu = -\mu^*$.

REMARKS:

- Let $x^* \in \mathbb{R}^N$ be a solution to the primal problem, and $u^* \in \mathbb{R}^M$ be a solution to the dual problem.

How to compute $\mu^* = f^*(-L^*u^*) + g^*(u^*)$?

>We have
$$\begin{cases} f^*(-L^*u^*) = \langle -L^*u^* | x^* \rangle - f(x^*) \\ g^*(u^*) = \langle u^* | Lx^* \rangle - g(Lx^*) \end{cases}$$

Dual algorithms

Use duality to approximate proximity operators

PRIMAL PROBLEM: Find $\mathbf{x}^* = \text{prox}_{f+g \circ \mathbf{L}}(\mathbf{y}) = \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmin}} f(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{Lx})$

DUAL PROBLEM: Find $\mathbf{u}^* = \underset{\mathbf{u} \in \mathbb{R}^M}{\text{Argmin}} (f + \frac{1}{2} \|\cdot - \mathbf{y}\|^2)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

- $\mathbf{y} \in \mathbb{R}^N$, $f \in \Gamma_0(\mathbb{R}^N)$, $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

Proof: Recall that the Fenchel-Rockafellar duality gives:

$$(P) \underset{\mathbf{x}}{\text{minimise}} \quad F(\mathbf{x}) + G(\mathbf{Lx})$$

$$(D) \underset{\mathbf{u}}{\text{minimise}} \quad F^*(-\mathbf{L}^* \mathbf{u}) + G^*(\mathbf{u})$$

To obtain the dual problem, use $F = f + \frac{1}{2} \|\cdot - \mathbf{y}\|^2$ and $G = g$. □

Use duality to approximate proximity operators

PRIMAL PROBLEM: Find $\mathbf{x}^* = \text{prox}_{f+g \circ \mathbf{L}}(\mathbf{y}) = \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmin}} f(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + g(\mathbf{Lx})$

DUAL PROBLEM: Find $\mathbf{u}^* = \underset{\mathbf{u} \in \mathbb{R}^M}{\text{Argmin}} (f + \frac{1}{2} \|\cdot - \mathbf{y}\|^2)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

- $\mathbf{y} \in \mathbb{R}^N$, $f \in \Gamma_0(\mathbb{R}^N)$, $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

REMARK: [(Lem. 2.5) Combettes et al, 2010]

Let $\varphi = f + \frac{1}{2} \|\cdot - \mathbf{y}\|^2$. Then $\varphi^* = \widetilde{f}^*(\cdot + \mathbf{y}) - \frac{1}{2} \|\mathbf{z}\|^2$

Where \widetilde{f}^* is the **Moreau enveloppe** of f^* : $\widetilde{f}^*(\mathbf{v}) = \min_{\mathbf{z}} f^*(\mathbf{z}) + \frac{1}{2} \|\mathbf{z} - \mathbf{v}\|^2$

DUAL PROBLEM: Find $\mathbf{u}^* \in \underset{\mathbf{u} \in \mathbb{R}^M}{\text{Argmin}} \widetilde{f}^*(\mathbf{y} - \mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

- \widetilde{f}^* is differentiable and $\nabla \widetilde{f}^* = \text{prox}_f = \text{Id} - \text{prox}_{f^*}$ [Moreau, 1965]
- Use FB on the dual problem!

Dual FB algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^* = \text{prox}_{f+g \circ L}(\mathbf{y}) = \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmin}} f(\mathbf{x}) + \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + g(L\mathbf{x})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \underset{\mathbf{u} \in \mathbb{R}^M}{\text{Argmin}} \tilde{f}^*(\mathbf{y} - L^*\mathbf{u}) + g^*(\mathbf{u})$

where $\mathbf{y} \in \mathbb{R}^N$, $f \in \Gamma_0(\mathbb{R}^N)$, $g \in \Gamma_0(\mathbb{R}^M)$ and $L \in \mathbb{R}^{M \times N}$

Choose $\mathbf{u}_0 \in \mathbb{R}^M$, and, for every $k \in \mathbb{N}$, choose $\gamma_k \in]0, 2/\|L\|^2[$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_k = \text{prox}_f(\mathbf{y} - L^*\mathbf{u}_k) \\ \mathbf{u}_{k+1} = \text{prox}_{\gamma_k g^*}(\mathbf{u}_k + \gamma_k L\mathbf{x}_k) \end{cases}$$

The sequence $(\mathbf{u}_k)_{k \in \mathbb{N}}$ converges to a solution to the dual problem \mathbf{u}^* .

The sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a solution to the primal problem $\mathbf{x}^* = \text{prox}_f(\mathbf{z} - L^*\mathbf{u}^*)$.

Moreau proximal decomposition theorem

QUESTION: How to compute $\text{prox}_{\sigma g^*}$?

Moreau decomposition theorem:

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \text{prox}_{\gamma g^*}(\mathbf{x}) = \mathbf{x} - \sigma \text{prox}_{\sigma^{-1}g}(\sigma^{-1}\mathbf{x})$$

EXAMPLES:

- If $g = \eta \|\cdot\|_1$, where $\eta > 0$, then

$$\text{prox}_{\gamma g^*}(\mathbf{x}) = \mathbf{x} - \sigma \text{prox}_{\sigma^{-1}\eta \|\cdot\|_1}(\sigma^{-1}\mathbf{x})$$

- If $g = \iota_{\mathcal{C}}$, where \mathcal{C} is a convex, closed, non-empty subset of \mathbb{R}^N , then

$$\text{prox}_{\gamma g^*}(\mathbf{x}) = \mathbf{x} - \sigma \Pi_{\mathcal{C}}(\sigma^{-1}\mathbf{x})$$

Dual FB algorithm: Example to denoising problem

GAUSSIAN DENOISING PROBLEM: $\mathbf{z} = \bar{\mathbf{x}} + \mathbf{w}$

where $\mathbf{w} \in \mathbb{R}^N$ is a realisation of a white Gaussian noise with variance $\sigma > 0$

MAP ESTIMATE: Find $\mathbf{x}^* = \operatorname{argmin}_{\mathbf{x} \in \mathbb{R}^N} \underbrace{\frac{1}{2} \|\mathbf{x} - \mathbf{z}\|^2}_{\text{data-fidelity}} + \underbrace{g(\mathbf{Lx})}_{\text{regularisation}} = \operatorname{prox}_{g \circ \mathbf{L}}(\mathbf{z})$

- EXAMPLE OF REGULARISATION: $g(\mathbf{Lx}) = \|\mathbf{x}\|_{\text{TV}}$

ANISOTROPIC TOTAL VARIATION

- Sparse transform: $\mathbf{L} = [\mathbf{L}_1^\top, \mathbf{L}_2^\top]^\top \in \mathbb{R}^{2N \times N}$
- $\|\mathbf{x}\|_{\text{TV}} = \|\mathbf{Lx}\|_1 = \|\mathbf{L}_1 \mathbf{x}\|_1 + \|\mathbf{L}_2 \mathbf{x}\|_1$

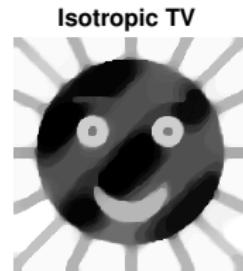
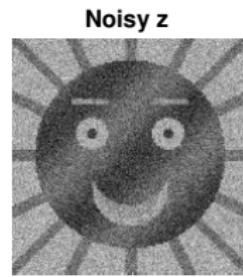
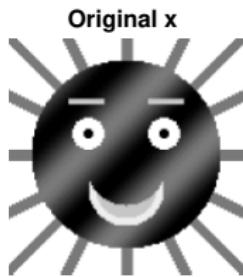
ISOTROPIC TOTAL VARIATION

- Sparse transform: $\mathbf{L} = [\mathbf{L}_1^\top, \mathbf{L}_2^\top]^\top \in \mathbb{R}^{2N \times N}$
- $\|\mathbf{x}\|_{\text{TV}} = \|\mathbf{Lx}\|_{1,2} = (\sqrt{|[\mathbf{L}_1 \mathbf{x}]_n| + |[\mathbf{L}_2 \mathbf{x}]_2|})_{1 \leq n \leq N}$

Dual FB algorithm: Example to denoising problem

GAUSSIAN DENOISING PROBLEM : $\mathbf{z} = \bar{\mathbf{x}} + \mathbf{w}$

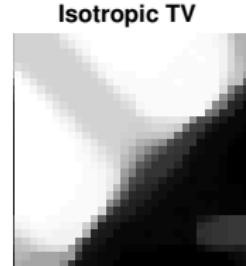
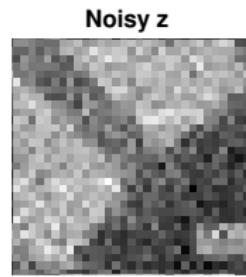
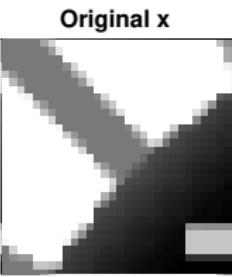
where $\mathbf{w} \in \mathbb{R}^N$ is a realisation of a white Gaussian noise with variance $\sigma > 0$



Dual FB algorithm: Example to denoising problem

GAUSSIAN DENOISING PROBLEM : $\mathbf{z} = \bar{\mathbf{x}} + \mathbf{w}$

where $\mathbf{w} \in \mathbb{R}^N$ is a realisation of a white Gaussian noise with variance $\sigma > 0$



Dual FB algorithm: Generalisation

PRIMAL PROBLEM: Find $\mathbf{x}^* = \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{y}\|^2 + \sum_{i=1}^I g_i(\mathbf{L}_i \mathbf{x})$

where $\mathbf{y} \in \mathbb{R}^N$, and, for every $i \in \{1, \dots, I\}$, $g_i \in \Gamma_0(\mathbb{R}_i^M)$, $\mathbf{L}_i \in \mathbb{R}^{M_i \times N}$, and $\omega_i \in]0, 1]$ such that $\sum_i \omega_i = 1$

Choose $(\mathbf{u}_{i,0}) \in \mathbb{R}^{M_1} \times \dots \times \mathbb{R}^{M_I}$, and, for every $k \in \mathbb{N}$, choose $\gamma_k \in]0, 2/(\max_i \|\mathbf{L}_i\|^2)[$.

For $k = 0, 1, \dots$

$$\mathbf{x}_k = \mathbf{y} - \sum_i \omega_i \mathbf{L}_i^* \mathbf{u}_{i,k}$$

for $i = 1, \dots, I$

$$\mathbf{u}_{k+1} = \operatorname{prox}_{\gamma_k g_i^*} (\mathbf{u}_{i,k} + \gamma_k \mathbf{L}_i \mathbf{x}_k)$$

The sequence $(\mathbf{u}_{1,k}, \dots, \mathbf{u}_{I,k})_{k \in \mathbb{N}}$ converges to a solution to the dual problem $(\mathbf{u}_i^*)_{1 \leq i \leq I}$.

The sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a solution to the primal problem $\mathbf{x}^* = \operatorname{prox}_f(\mathbf{z} - \sum_i \omega_i \mathbf{L}_i^* \mathbf{u}_i^*)$.

Primal-Dual algorithms

Problem formulation

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{L}\mathbf{x})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \operatorname{Argmin}_{\mathbf{u} \in \mathbb{R}^M} (h + f)^*(-\mathbf{L}^*\mathbf{u}) + g^*(\mathbf{u})$

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable
- $f \in \Gamma_0(\mathbb{R}^N)$
- $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

Proof: Recall that the Fenchel-Rockafellar duality gives:

$$(P) \quad \underset{\mathbf{x}}{\operatorname{minimise}} \quad F(\mathbf{x}) + G(\mathbf{L}\mathbf{x})$$

$$(D) \quad \underset{\mathbf{u}}{\operatorname{minimise}} \quad F^*(-\mathbf{L}^*\mathbf{u}) + G^*(\mathbf{u})$$

To obtain the dual problem, use $F = h + f$ and $G = g$. □

Problem formulation

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{L}\mathbf{x})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \operatorname{Argmin}_{\mathbf{u} \in \mathbb{R}^M} (h + f)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable
- $f \in \Gamma_0(\mathbb{R}^N)$
- $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$

Proof: Recall that the Fenchel-Rockafellar duality gives:

$$(P) \underset{\mathbf{x}}{\operatorname{minimise}} \quad F(\mathbf{x}) + G(\mathbf{L}\mathbf{x})$$

$$(D) \underset{\mathbf{u}}{\operatorname{minimise}} \quad F^*(-\mathbf{L}^* \mathbf{u}) + G^*(\mathbf{u})$$

To obtain the dual problem, use $F = h + f$ and $G = g$. □

REMARK: $(\forall \mathbf{v} \in \mathbb{R}^N) \quad (h + f)^*(\mathbf{v}) = \min_{\mathbf{x}' \in \mathbb{R}^N} h^*(\mathbf{v} - \mathbf{x}') + f^*(\mathbf{x}')$

Lagrangian and Karush-Kuhn-Tucker conditions

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{Lx})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \operatorname{Argmin}_{\mathbf{u} \in \mathbb{R}^M} (h + f)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

Lagrangian formulation Another formulation of the Primal-Dual problem is to combine them into the search of a **saddle point of the Lagrangian**:

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} | \mathbf{u} \rangle$

Lagrangian and Karush-Kuhn-Tucker conditions

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{Lx})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \operatorname{Argmin}_{\mathbf{u} \in \mathbb{R}^M} (h + f)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

Lagrangian formulation Another formulation of the Primal-Dual problem is to combine them into the search of a **saddle point of the Lagrangian**:

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} | \mathbf{u} \rangle$

REMARK: Recall that, for $\psi \in \Gamma_0(\mathbb{R}^N)$, $\mathbf{x}^* \in \operatorname{Argmin} \psi \Leftrightarrow \mathbf{0} \in \partial\psi(\mathbf{x}^*)$

Do we have similar conditions for the primal-dual problem?

- ~~~ Look at the Lagrangian saddle point problem and derive optimal conditions for \mathbf{x}^* , and for \mathbf{u}^* *alternatively*
- ~~~ These are called **KKT conditions**

Lagrangian and Karush-Kuhn-Tucker conditions

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{Lx})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \operatorname{Argmin}_{\mathbf{u} \in \mathbb{R}^M} (h + f)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

Lagrangian formulation Another formulation of the Primal-Dual problem is to combine them into the search of a **saddle point of the Lagrangian**:

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} | \mathbf{u} \rangle$

Karush-Kuhn-Tucker conditions

Assume that $\operatorname{dom} g \cap \mathbf{L}(\operatorname{dom} f) \neq \emptyset$.

$(\mathbf{x}^*, \mathbf{u}^*) \in \mathbb{R}^N \times \mathbb{R}^M$ is a solution to the Primal-Dual problem if and only if

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

From KKT to fixed-point equations...

KKT:

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

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L}\mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L}\mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

Since $\mathbf{x}^* - \mathbf{x}^* = \mathbf{0}$, and $\mathbf{u}^* - \mathbf{u}^* = \mathbf{0}$, the last equations are equivalent to

$$\begin{cases} \mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) - \mathbf{x}^* \in \tau \partial f(\mathbf{x}^*) \\ \mathbf{u}^* + \sigma \mathbf{L}\mathbf{x}^* - \mathbf{u}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L} \mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

Since $\mathbf{x}^* - \mathbf{x}^* = \mathbf{0}$, and $\mathbf{u}^* - \mathbf{u}^* = \mathbf{0}$, the last equations are equivalent to

$$\begin{cases} \mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) - \mathbf{x}^* \in \tau \partial f(\mathbf{x}^*) \\ \mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*) - \mathbf{u}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L} \mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

Since $\mathbf{x}^* - \mathbf{x}^* = \mathbf{0}$, and $\mathbf{u}^* - \mathbf{u}^* = \mathbf{0}$, the last equations are equivalent to

$$\begin{cases} \underbrace{\mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*))}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}} \in \tau \partial f(\underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\tau f} \\ \underbrace{\mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*) - \mathbf{u}^*}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}} \in \sigma \partial g^*(\underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\sigma g^*} \end{cases}$$

Prox characterisation: $\bar{\mathbf{x}} - \bar{\mathbf{p}} \in \gamma \partial \psi(\bar{\mathbf{p}}) \Leftrightarrow \bar{\mathbf{p}} = \text{prox}_{\gamma \psi}(\bar{\mathbf{x}})$

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L} \mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

Since $\mathbf{x}^* - \mathbf{x}^* = \mathbf{0}$, and $\mathbf{u}^* - \mathbf{u}^* = \mathbf{0}$, the last equations are equivalent to

$$\begin{cases} \underbrace{\mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*))}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}} \in \tau \partial f(\underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\tau f} \\ \underbrace{\mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*)}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}} \in \sigma \partial g^*(\underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\sigma g^*} \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{x}^* = \text{prox}_{\tau f} \left(\mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \right) \\ \mathbf{u}^* = \text{prox}_{\sigma g^*} \left(\mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*) \right) \end{cases}$$

From KKT to fixed-point equations...

KKT:

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

Multiply by $\tau > 0$ the first equation and $\sigma > 0$ the second equation:

$$\begin{cases} -\tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \in \tau \partial f(\mathbf{x}^*) \\ \sigma \mathbf{L} \mathbf{x}^* \in \sigma \partial g^*(\mathbf{u}^*) \end{cases}$$

Since $\mathbf{x}^* - \mathbf{x}^* = \mathbf{0}$, and $\mathbf{u}^* - \mathbf{u}^* = \mathbf{0}$, the last equations are equivalent to

$$\begin{cases} \underbrace{\mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*))}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}} \in \tau \partial f(\underbrace{\mathbf{x}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\tau f} \\ \underbrace{\mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*)}_{\bar{\mathbf{x}}} - \underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}} \in \sigma \partial g^*(\underbrace{\mathbf{u}^*}_{\bar{\mathbf{p}}}) \rightsquigarrow \text{prox}_{\sigma g^*} \end{cases}$$

$$\Leftrightarrow \begin{cases} \mathbf{x}^* = \text{prox}_{\tau f} \left(\mathbf{x}^* - \tau(\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \right) \\ \mathbf{u}^* = \text{prox}_{\sigma g^*} \left(\mathbf{u}^* + \sigma \mathbf{L}(2\mathbf{x}^* - \mathbf{x}^*) \right) \end{cases} \rightsquigarrow \text{Fixed-point equations}$$

Fixed-point algorithm

From the fixed-point equations:

$$\begin{cases} \mathbf{x}^* = \text{prox}_{\tau f} \left(\mathbf{x}^* - \tau (\mathbf{L}^* \mathbf{u}^* + \nabla h(\mathbf{x}^*)) \right) \\ \mathbf{u}^* = \text{prox}_{\sigma g^*} \left(\mathbf{u}^* + \sigma \mathbf{L} (2\mathbf{x}^* - \mathbf{x}^*) \right) \end{cases}$$

We can derive a fixed-point algorithm:

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \text{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\mathbf{L}^* \mathbf{u}_k + \nabla h(\mathbf{x}_k)) \right) \\ \mathbf{u}_{k+1} = \text{prox}_{\sigma g^*} \left(\mathbf{u}_k + \sigma \mathbf{L} (2\mathbf{x}_{k+1} - \mathbf{x}_k) \right) \end{cases}$$

REMARKS:

- This algorithm is known as the Condat-Vu algorithm
- The FB algorithm is also a fixed-point algorithm:

$$\mathbf{x}_{k+1} = \text{prox}_{\gamma g} (\mathbf{x}_k - \gamma \nabla h(\mathbf{x}_k)) \quad " \xrightarrow[k \rightarrow \infty]{} " \quad \mathbf{x}^* = \text{prox}_{\gamma g} (\mathbf{x}^* - \gamma \nabla h(\mathbf{x}^*))$$

Condat-Vu algorithm as a forward-backward algorithm

We can see that the Condat-Vu algorithm has a very similar structure as the FB algorithm:

For $k = 0, 1, \dots$

$$\left[\begin{array}{l} \mathbf{x}_{k+1} = \underbrace{\text{prox}_{\tau f}}_{\text{backward step}} \left(\underbrace{\mathbf{x}_k - \tau(\mathbf{L}^* \mathbf{u}_k + \nabla h(\mathbf{x}_k))}_{\text{forward step}} \right) \rightsquigarrow \text{"primal" update} \\ \mathbf{u}_{k+1} = \underbrace{\text{prox}_{\sigma g^*}}_{\text{backward step}} \left(\underbrace{\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{forward step}} \right) \rightsquigarrow \text{"dual" update} \end{array} \right]$$

Condat-Vu algorithm as a forward-backward algorithm

We can see that the Condat-Vu algorithm has a very similar structure as the FB algorithm:

For $k = 0, 1, \dots$

$$\left[\begin{array}{l} \mathbf{x}_{k+1} = \underbrace{\text{prox}_{\tau f}}_{\text{backward step}} \left(\underbrace{\mathbf{x}_k - \tau(\mathbf{L}^* \mathbf{u}_k + \nabla h(\mathbf{x}_k))}_{\text{forward step}} \right) \rightsquigarrow \text{"primal" update} \\ \mathbf{u}_{k+1} = \underbrace{\text{prox}_{\sigma g^*}}_{\text{backward step}} \left(\underbrace{\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{forward step}} \right) \rightsquigarrow \text{"dual" update} \end{array} \right]$$

In fact, it is derived from the FB algorithm, applied to the primal-dual problem (or equivalently to the Lagrangian saddle point problem)...

Lagrangian saddle point problem

$$\text{Find } (\mathbf{x}^*, \mathbf{u}^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\text{Argmin}} \underset{\mathbf{u} \in \mathbb{R}^M}{\max} \mathcal{L}(\mathbf{x}, \mathbf{u})$$

$$\text{where } (\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u} \rangle$$

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

Let $k \in \mathbb{N}$. Look at $\mathcal{L}(\cdot, \mathbf{u}_k)$ for $\mathbf{u}_k \in \mathbb{R}^M$ fixed:

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}_k) = \underbrace{h(\mathbf{x})}_{\text{smooth}} + \underbrace{f(\mathbf{x})}_{\text{non-smooth}} - \underbrace{g^*(\mathbf{u}_k)}_{\text{constant}} + \underbrace{\langle \mathbf{Lx} \mid \mathbf{u}_k \rangle}_{\text{smooth}}$$

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

Let $k \in \mathbb{N}$. Look at $\mathcal{L}(\cdot, \mathbf{u}_k)$ for $\mathbf{u}_k \in \mathbb{R}^M$ **fixed**:

$$(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}_k) = \underbrace{h(\mathbf{x})}_{\text{smooth}} + \underbrace{f(\mathbf{x})}_{\text{non-smooth}} - \underbrace{g^*(\mathbf{u}_k)}_{\text{constant}} + \underbrace{\langle \mathbf{Lx} \mid \mathbf{u}_k \rangle}_{\text{smooth}}$$

FB update: $\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right)$

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

FB update: $\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right)$

DUAL UPDATE

Let $k \in \mathbb{N}$. To compute \mathbf{u}_{k+1} we want to use the new update \mathbf{x}_{k+1} , **and** keep information from \mathbf{x}_k : Look at $-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \cdot)$ for $(\mathbf{x}_{k+1}, \mathbf{x}_k) \in \mathbb{R}^N \times \mathbb{R}^N$ **fixed!**

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{Lx} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

FB update: $\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right)$

DUAL UPDATE

Let $k \in \mathbb{N}$. To compute \mathbf{u}_{k+1} we want to use the new update \mathbf{x}_{k+1} , **and** keep information from \mathbf{x}_k : Look at $-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \cdot)$ for $(\mathbf{x}_{k+1}, \mathbf{x}_k) \in \mathbb{R}^N \times \mathbb{R}^N$ **fixed!**

REMARK: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathcal{L}(2\mathbf{x} - \mathbf{x}, \mathbf{u}) = \mathcal{L}(\mathbf{x}, \mathbf{u})$

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

FB update: $\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right)$

DUAL UPDATE

Let $k \in \mathbb{N}$. To compute \mathbf{u}_{k+1} we want to use the new update \mathbf{x}_{k+1} , **and** keep information from \mathbf{x}_k : Look at $-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \cdot)$ for $(\mathbf{x}_{k+1}, \mathbf{x}_k) \in \mathbb{R}^N \times \mathbb{R}^N$ **fixed!**

$$-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{u}) = \underbrace{-h(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{constant}} - \underbrace{f(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{constant}} + \underbrace{g^*(\mathbf{u})}_{\text{non-smooth}} - \underbrace{\langle \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k) \mid \mathbf{u} \rangle}_{\text{smooth}}$$

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^*, \mathbf{u}^*) \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \max_{\mathbf{u} \in \mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

where $(\forall \mathbf{x} \in \mathbb{R}^N)(\forall \mathbf{u} \in \mathbb{R}^M) \quad \mathcal{L}(\mathbf{x}, \mathbf{u}) = h(\mathbf{x}) + f(\mathbf{x}) - g^*(\mathbf{u}) + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u} \rangle$

PRIMAL UPDATE

FB update: $\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right)$

DUAL UPDATE

Let $k \in \mathbb{N}$. To compute \mathbf{u}_{k+1} we want to use the new update \mathbf{x}_{k+1} , **and** keep information from \mathbf{x}_k : Look at $-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \cdot)$ for $(\mathbf{x}_{k+1}, \mathbf{x}_k) \in \mathbb{R}^N \times \mathbb{R}^N$ **fixed!**

$$-\mathcal{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k, \mathbf{u}) = \underbrace{-h(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{constant}} - \underbrace{-f(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{constant}} + \underbrace{-g^*(\mathbf{u})}_{\text{non-smooth}} - \underbrace{\langle \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k) \mid \mathbf{u} \rangle}_{\text{smooth}}$$

FB update: $\mathbf{u}_{k+1} = \operatorname{prox}_{\sigma g^*} \left(\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k) \right)$

Step-size and convergence of Condat-Vu algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \text{Argmin}_{\mathbf{x} \in \mathbb{R}^N} h(\mathbf{x}) + f(\mathbf{x}) + g(\mathbf{L}\mathbf{x})$

DUAL PROBLEM: Find $\mathbf{u}^* \in \text{Argmin}_{\mathbf{u} \in \mathbb{R}^M}^{x \in \mathbb{R}^N} (h + f)^*(-\mathbf{L}^* \mathbf{u}) + g^*(\mathbf{u})$

where $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable, $f \in \Gamma_0(\mathbb{R}^N)$,
 $g \in \Gamma_0(\mathbb{R}^M)$ and $\mathbf{L} \in \mathbb{R}^{M \times N}$.

Choose $\tau > 0$ and $\sigma > 0$ such that $\frac{1}{\tau} - \sigma \|\mathbf{L}\|^2 > \frac{\beta}{2}$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \text{prox}_{\tau f} \left(\mathbf{x}_k - \tau (\nabla h(\mathbf{x}_k) + \mathbf{L}^* \mathbf{u}_k) \right) \\ \mathbf{u}_{k+1} = \text{prox}_{\sigma g^*} \left(\mathbf{u}_k + \sigma \mathbf{L} (2\mathbf{x}_{k+1} - \mathbf{x}_k) \right) \end{cases}$$

The sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a solution to the primal problem.

The sequence $(\mathbf{u}_k)_{k \in \mathbb{N}}$ converges to a solution to the dual problem.

Particular cases

CHAMBOLLE-POCK ALGORITHM: $h \equiv 0$ [Chambolle & Pock, 2011]

PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{Lx})$

Choose $\tau > 0$ and $\sigma > 0$ such that $\sigma\tau\|\mathbf{L}\|^2 < 1$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} (\mathbf{x}_k - \tau \mathbf{L}^* \mathbf{u}_k) \\ \mathbf{u}_{k+1} = \operatorname{prox}_{\sigma g^*} (\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)) \end{cases}$$

Particular cases

CHAMBOLLE-POCK ALGORITHM: $h \equiv 0$ [Chambolle & Pock, 2011]

PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{Lx})$

Choose $\tau > 0$ and $\sigma > 0$ such that $\sigma\tau\|\mathbf{L}\|^2 < 1$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} (\mathbf{x}_k - \tau \mathbf{L}^* \mathbf{u}_k) \\ \mathbf{u}_{k+1} = \operatorname{prox}_{\sigma g^*} (\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)) \end{cases}$$

DOUGLAS-RACHFORD ALGORITHM: $h \equiv 0$, $\mathbf{L} = \operatorname{Id}$ and $\tau = 1/\sigma$

PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{x})$

Choose $\sigma \in]0, 1[$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\sigma^{-1} f} (\mathbf{x}_k - \sigma^{-1} \mathbf{u}_k) \\ \mathbf{u}_{k+1} = \operatorname{prox}_{\sigma g^*} (\mathbf{u}_k + \sigma(2\mathbf{x}_{k+1} - \mathbf{x}_k)) \end{cases}$$

Particular cases

CHAMBOLLE-POCK ALGORITHM: $\mathbf{h} \equiv 0$ [Chambolle & Pock, 2011]

PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{Lx})$

Choose $\tau > 0$ and $\sigma > 0$ such that $\sigma\tau\|\mathbf{L}\|^2 < 1$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\tau f} (\mathbf{x}_k - \tau \mathbf{L}^* \mathbf{u}_k) \\ \mathbf{u}_{k+1} = \operatorname{prox}_{\sigma g^*} (\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)) \end{cases}$$

DOUGLAS-RACHFORD ALGORITHM: $\mathbf{h} \equiv 0$, $\mathbf{L} = \operatorname{Id}$ and $\tau = 1/\sigma$

PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{x})$

Choose $\sigma \in]0, 1[$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \operatorname{prox}_{\sigma^{-1} f} (\mathbf{s}_k) \\ \mathbf{s}_{k+1} = \mathbf{s}_k - \mathbf{x}_{k+1} - \operatorname{prox}_{\sigma^{-1} g} (2\mathbf{x}_{k+1} - \mathbf{s}_k) \end{cases}$$

Application: ℓ_1 in a redundant dictionary + constraints

$$\text{Find } \mathbf{x}^* \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2 + \eta \|\Psi \mathbf{x}\|_1 + \iota_{[\mathbf{x}_{\min}, \mathbf{x}_{\max}]^N}(\mathbf{x})$$

where $\Phi \in \mathbb{R}^{M \times N}$, $\mathbf{z} \in \mathbb{R}^M$, $\eta > 0$, $\Psi \in \mathbb{R}^{L \times N}$, $(\mathbf{x}_{\min}, \mathbf{x}_{\max}) \in \mathbb{R}^2$.

Choose $\tau > 0$ and $\sigma > 0$ such that $\frac{1}{\tau} - \sigma \|\Psi\|^2 > \frac{\|\Phi\|^2}{2}$.

For $k = 0, 1, \dots$

$$\left[\begin{array}{l} \mathbf{x}_{k+1} = \Pi_{[\mathbf{x}_{\min}, \mathbf{x}_{\max}]^N} \left(\mathbf{x}_k - \tau (\Phi^* (\Phi \mathbf{x}_k - \mathbf{z}) + \Psi^* \mathbf{u}_k) \right) \\ \tilde{\mathbf{u}}_k = \mathbf{u}_k + \sigma \Psi (2\mathbf{x}_{k+1} - \mathbf{x}_k) \\ \mathbf{u}_{k+1} = \tilde{\mathbf{u}}_k - \sigma \operatorname{prox}_{\sigma^{-1} \eta \|\cdot\|_1} \left(\sigma^{-1} \tilde{\mathbf{u}}_k \right) \end{array} \right]$$

Application: ℓ_1 -ball regularisation

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2 + \iota_{\mathcal{B}_1(\mathbf{0}, \eta)}(\Psi \mathbf{x}) + \iota_{[\mathbf{x}_{\min}, \mathbf{x}_{\max}]^N}(\mathbf{x})$

where $\Phi \in \mathbb{R}^{M \times N}$, $\mathbf{z} \in \mathbb{R}^M$, $\eta > 0$, $\Psi \in \mathbb{R}^{L \times N}$, $(\mathbf{x}_{\min}, \mathbf{x}_{\max}) \in \mathbb{R}^2$, and $\mathcal{B}_1(\mathbf{0}, \eta)$ denotes the ℓ_1 -ball centred in $\mathbf{0}$ with radius η .

Choose $\tau > 0$ and $\sigma > 0$ such that $\frac{1}{\tau} - \sigma \|\Psi\|^2 > \frac{\|\Phi\|^2}{2}$.

For $k = 0, 1, \dots$

$$\begin{cases} \mathbf{x}_{k+1} = \Pi_{[\mathbf{x}_{\min}, \mathbf{x}_{\max}]^N} \left(\mathbf{x}_k - \tau (\Phi^*(\Phi \mathbf{x}_k - \mathbf{z}) + \Psi^* \mathbf{u}_k) \right) \\ \tilde{\mathbf{u}}_k = \mathbf{u}_k + \sigma \Psi (2\mathbf{x}_{k+1} - \mathbf{x}_k) \\ \mathbf{u}_{k+1} = \tilde{\mathbf{u}}_k - \sigma \Pi_{\mathcal{B}_1(\mathbf{0}, \eta)} \left(\sigma^{-1} \tilde{\mathbf{u}}_k \right) \end{cases}$$

Example: 3D mesh denoising

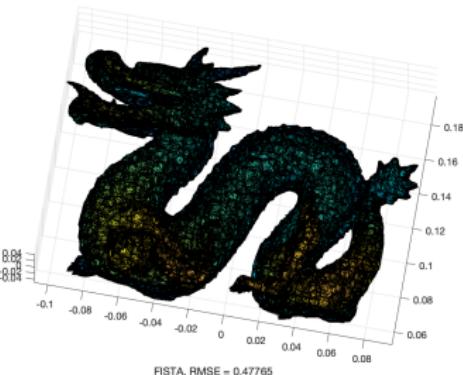
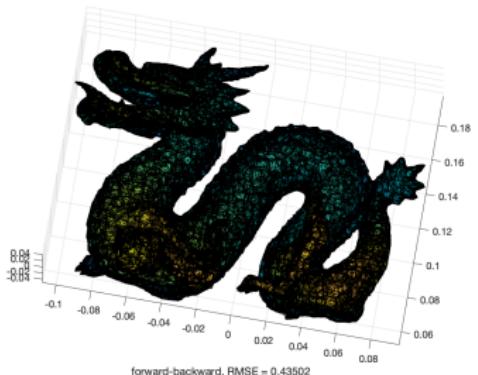
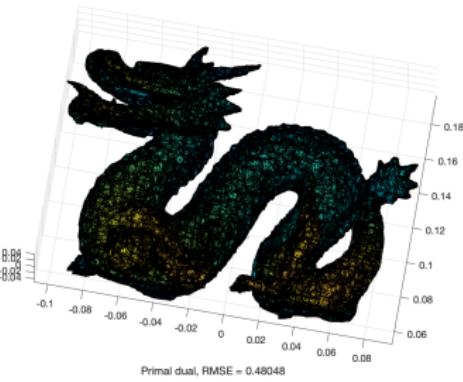
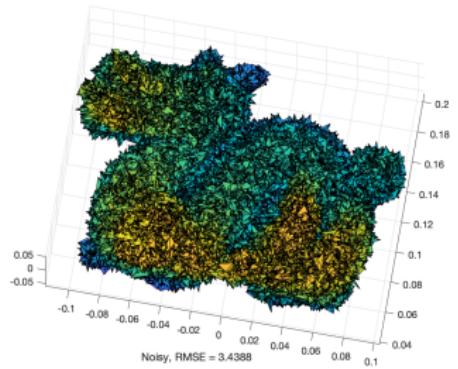
INVERSE PROBLEM: $\mathbf{z} = \bar{\mathbf{x}} + \mathbf{w}$

- 437,645 nodes, 1,309,193 edges
- \mathbf{w} realisation of white Gaussian noise

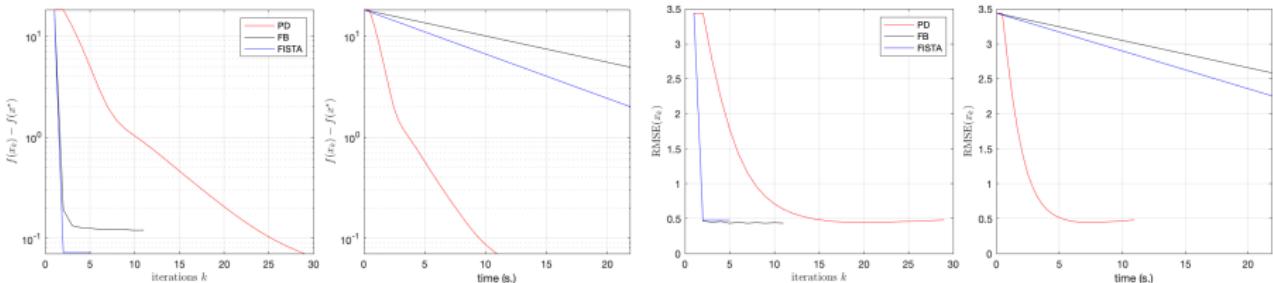
MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} F(\mathbf{x}) + R(\mathbf{x})$

- Data-fidelity: $F(\mathbf{x}) = \frac{1}{2} \|\mathbf{z} - \mathbf{x}\|^2$
~~~ Lipschitz-differentiable
- Regularisation:  $R(\mathbf{x}) = \eta \|\mathbf{x}\|_{\text{TV}} + \iota_{[0,1]^N}(\mathbf{x})$   
with  $\eta > 0$ , and  $\|\cdot\|_{\text{TV}}$  isotropic TV on graphs  
~~~ Splitting to handle TV and constraint separately (with proximity operators)

Example: 3D mesh denoising



Example: 3D mesh denoising



Generalisation of the Condat-Vu algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x}} h(\mathbf{x}) + f(\mathbf{x}) + \sum_{i=1}^I g_i(\mathbf{L}_i \mathbf{x})$

DUAL PROBLEM: Find $(\mathbf{u}_i^*)_{1 \leq i \leq I} \in \operatorname{Argmin}_{(\mathbf{u}_1, \dots, \mathbf{u}_I)} (h + f)^*(-\sum_{i=1}^I \mathbf{L}_i^* \mathbf{u}_i) + \sum_{i=1}^I g_i^*(\mathbf{u}_i)$

where $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable, $f \in \Gamma_0(\mathbb{R}^N)$,
for every $i \in \{1, \dots, I\}$, $g_i \in \Gamma_0(\mathbb{R}^{M_i})$ and $\mathbf{L}_i \in \mathbb{R}^{M_i \times N}$.

Choose $\tau > 0$ and $\sigma > 0$ such that $\frac{1}{\tau} - \sigma(\sum_{i=1}^I \|\mathbf{L}_i\|^2) > \frac{\beta}{2}$.

For $k = 0, 1, \dots$

$$\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f}\left(\mathbf{x}_k - \tau(\nabla h(\mathbf{x}_k) + \sum_{i=1}^I \mathbf{L}_i^* \mathbf{u}_{i,k})\right)$$

for $i = 1, \dots, I$

$$\mathbf{u}_{i,k+1} = \operatorname{prox}_{\sigma g_i^*}\left(\mathbf{u}_{i,k} + \sigma \mathbf{L}_i(2\mathbf{x}_{k+1} - \mathbf{x}_k)\right)$$

The sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges to a solution to the primal problem.

The sequence $(\mathbf{u}_{1,k}, \dots, \mathbf{u}_{I,k})_{k \in \mathbb{N}}$ converges to a solution to the dual problem.

Generalisation of the Condat-Vu algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x}} h(\mathbf{x}) + f(\mathbf{x}) + \sum_{i=1}^I g_i(\mathbf{L}_i \mathbf{x})$

DUAL PROBLEM: Find $(\mathbf{u}_i^*)_{1 \leq i \leq I} \in \operatorname{Argmin}_{(\mathbf{u}_1, \dots, \mathbf{u}_I)} (h + f)^*(-\sum_{i=1}^I \mathbf{L}_i^* \mathbf{u}_i) + \sum_{i=1}^I g_i^*(\mathbf{u}_i)$

where $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex and β -Lipschitz differentiable, $f \in \Gamma_0(\mathbb{R}^N)$,
for every $i \in \{1, \dots, I\}$, $g_i \in \Gamma_0(\mathbb{R}^{M_i})$ and $\mathbf{L}_i \in \mathbb{R}^{M_i \times N}$.

Choose $\tau > 0$ and $\sigma > 0$ such that $\frac{1}{\tau} - \sigma(\sum_{i=1}^I \|\mathbf{L}_i\|^2) > \frac{\beta}{2}$.

For $k = 0, 1, \dots$

$$\mathbf{x}_{k+1} = \operatorname{prox}_{\tau f}\left(\mathbf{x}_k - \tau(\nabla h(\mathbf{x}_k) + \sum_{i=1}^I \mathbf{L}_i^* \mathbf{u}_{i,k})\right)$$

for $i = 1, \dots, I$

$$\mathbf{u}_{i,k+1} = \operatorname{prox}_{\sigma g_i^*}\left(\mathbf{u}_{i,k} + \sigma \mathbf{L}_i(2\mathbf{x}_{k+1} - \mathbf{x}_k)\right)$$

If $h \equiv 0$, then condition on the step-sizes simplifies as $\sigma\tau(\sum_{i=1}^I \|\mathbf{L}_i\|^2) < 1$

Example: Radio-interferometric imaging

INVERSE PROBLEM: $\mathbf{z} = \Phi \bar{\mathbf{x}} + \mathbf{w}$

- $\Phi = \mathbf{G}\mathbf{F}\mathbf{Z}$ with $\mathbf{Z} \in \mathbb{R}^{\bar{N} \times N}$ zero-padding, $\mathbf{F} \in \mathbb{C}^{\bar{N} \times \bar{N}}$, and $\mathbf{G} \in \mathbb{C}^{M \times \bar{N}}$ (de)-gridding matrix
- \mathbf{w} realisation of white Gaussian noise

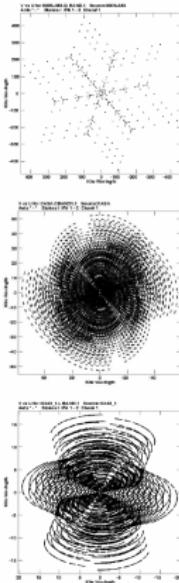
MINIMISATION PROBLEM: Find $\hat{\mathbf{x}} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} F(\mathbf{x}) + R(\mathbf{x})$

- Data-fidelity: $F(\mathbf{x}) = \iota_{B_2(\mathbf{z}, \varepsilon)}(\Phi \mathbf{x})$
 - ~~~ Splitting in the dual domain (proximity operator)
- Regularisation: $R(\mathbf{x}) = \eta \|\Psi \mathbf{x}\|_1 + \iota_{[0, +\infty]^N}(\mathbf{x})$
 - with $\eta > 0$, and Ψ concatenation of first 8 Db wavelets and Dirac basis (average sparsity)
 - ~~~ Splitting to handle all terms separately (with proximity operators)

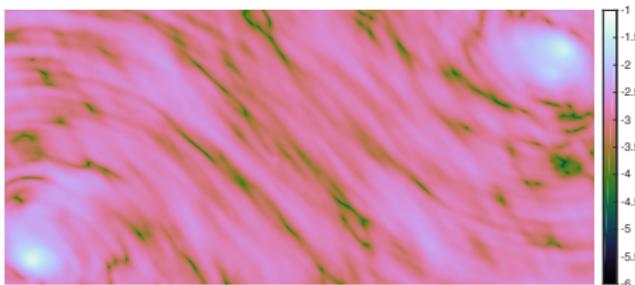
Example: Radio-interferometric imaging



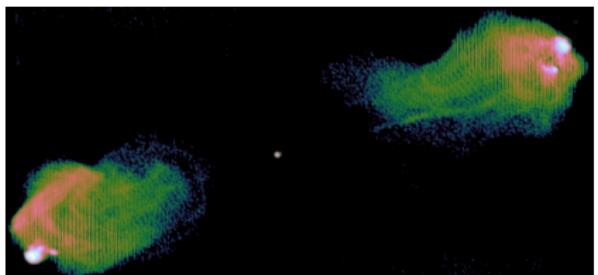
Image courtesy of NRAO/AUI



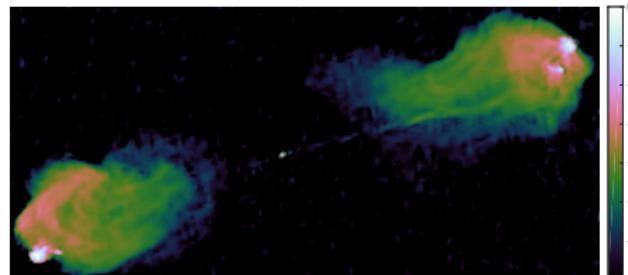
Example: Radio-interferometric imaging



Dirty image $\Phi^* z$



Standard method (CLEAN)



Primal-dual