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

REMARK: For simplicity, assume f to be convex

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

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

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

REMARK: For simplicity, assume f to be convex

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

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

  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

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

From the subgradient algorithm...

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

REMARK: For simplicity, assume f to be convex

- Subgradient algorithm [Shor, 1979]

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

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

REMARK: For simplicity, assume f to be convex

- Subgradient algorithm [Shor, 1979]

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

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

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

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

... to the origins of the proximity operator!

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

REMARK: For simplicity, assume f to be convex

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

- $\widehat{\mathbf{x}} = \mathrm{prox}_\psi(\mathbf{x})$
- $\mathrm{prox}_\psi\colon \mathbb{R}^N\to\mathbb{R}^N \rightsquigarrow$ proximity operator

☛ Proximal point algorithm

$$
\begin{aligned}
(\forall k\in\mathbb{N}) \qquad & \mathbf{x}_k-\mathbf{x}_{k+1}\in\gamma_k\partial\mathsf{f}(\mathbf{x}_{k+1})\\
\Leftrightarrow \quad & \mathbf{x}_{k+1} = \mathrm{prox}_{\gamma_k\mathsf{f}}(\mathbf{x}_k)
\end{aligned}
$$

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

$$\mathrm{prox}_{\mathsf{f}}(\overline{\mathbf{x}}) = \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{argmin}}\ \mathsf{f}(\mathbf{x}) + \frac{1}{2}\|\mathbf{x} - \overline{\mathbf{x}}\|^2$$

Characterisation of the proximity operator

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

$$\overline{\mathbf{p}} = \mathrm{prox}_{\mathsf{f}}(\overline{\mathbf{x}}) \quad \Leftrightarrow \quad \overline{\mathbf{x}} - \overline{\mathbf{p}} \in \partial \mathsf{f}(\overline{\mathbf{p}})$$

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 $\overline{\mathbf{x}} \in \mathbb{R}^N$. The projection of $\overline{\mathbf{x}}$ onto $\mathcal{C}$ is given by

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

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 $\overline{\mathbf{x}} \in \mathbb{R}^N$. The projection of $\overline{\mathbf{x}}$ onto $\mathcal{C}$ is given by

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

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

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

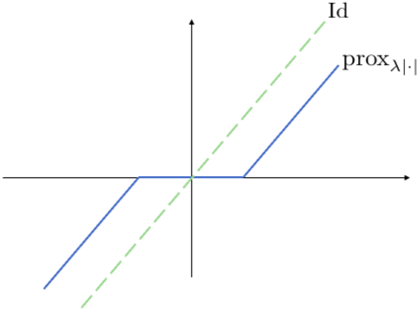
  $\overline{\mathbf{x}} = (\overline{\mathsf{x}}^{(n)})_{1\leqslant n\leqslant N} \in \mathbb{R}^N$. Then we have $\operatorname{prox}_{\mathsf{f}}(\overline{\mathbf{x}}) = \Big(\operatorname{prox}_{\psi_n}(\overline{\mathsf{x}}^{(n)})\Big)_{1\leqslant n\leqslant N}$

Proximity operator examples: ℓ_1 norm

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

$$\operatorname{prox}_{\mathsf{f}}(\overline{\mathbf{x}}) = \left(\operatorname{prox}_{\lambda|\cdot|}(\overline{x}^{(n)})\right)_{1 \leqslant n \leqslant N}$$

where, for every $n \in \{1, \ldots, N\}$, $\operatorname{prox}_{\lambda|\cdot|}(\overline{\mathsf{x}}^{(n)}) = \operatorname{sign}(\overline{\mathsf{x}}^{(n)}) \max\{|\overline{\mathsf{x}}^{(n)}| - \lambda, 0\}$

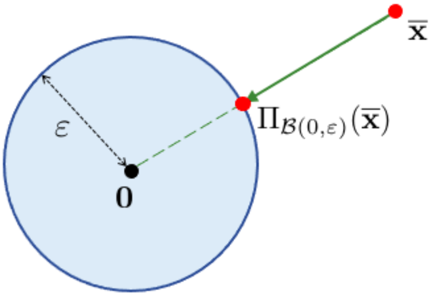


REMARK: Also known as *soft-thresholding* operator

Proximity operator examples: ℓ_2 ball

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

$$\mathrm{prox}_{\mathsf{f}}(\overline{\mathbf{x}}) = \Pi_{\mathcal{B}(\mathbf{0},\varepsilon)}(\overline{\mathbf{x}}) = \min\{\varepsilon, \|\overline{\mathbf{x}}\|\} \frac{\overline{\mathbf{x}}}{\|\overline{\mathbf{x}}\|}$$



Proximity operator: Denoising interpretation

Gaussian denoising problem

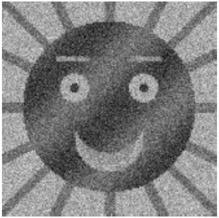
$$\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$$

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

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

Original x

Noisy 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

$$\text{Find } \widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \Big\{ \mathsf{f}(\mathbf{x}) = \mathsf{h}(\mathbf{x}) + \mathsf{g}(\mathbf{x}) \Big\}$$

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

Example: $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \mathbf{w}$

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

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

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

"Semi-smooth" minimisation problem

Minimisation problem

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

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

Remark: Usually $\operatorname{prox}_{\mathsf{f}}$ 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 $\mathbf{x}_0 \in \operatorname{dom} g$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2\beta^{-1}[$
For $k = 0, 1, \ldots$
$\lfloor \; \mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}\left(\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)\right)$

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

[Combettes & Wajs, 2005]

FB algorithm: Limits

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

Let $\mathbf{x}_0 \in \operatorname{dom} g$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2\beta^{-1}[$
For $k = 0, 1, \ldots$
$\lfloor \; \mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}\big(\mathbf{x}_k - \gamma_k \nabla h(\mathbf{x}_k)\big)$

- Convergence may be slow in practice...
  - ☛ Use Nesterov acceleration (*inertia*)
  - ☛ Use second order information (*preconditioning*)

- What if $\operatorname{prox}_{\gamma_k g}$ 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)** $(\mathbf{x}_{k'})_{k' \leqslant k}$ to build the next iterate $\mathbf{x}_{k+1}$.

WHY? Use memory to go faster!

For FB we have:

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

Introducing inertia would lead to

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

QUESTION: How to choose $\widetilde{\mathsf{T}}_k$?

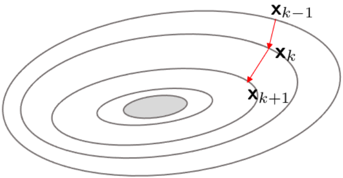
REMARK: In general $\widetilde{\mathsf{T}}_k$ only depends on $(\mathbf{x}_k, \mathbf{x}_{k-1})$ to avoid memory issues

Particular case: Gradient Descent algorithm

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

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 $k \in \mathbb{N}$, $h(\mathbf{x}_{k+1}) \leqslant h(\mathbf{x}_k)$
- This is the Gradient Descent (GD) algorithm

Particular case: Inertia for GD algorithm

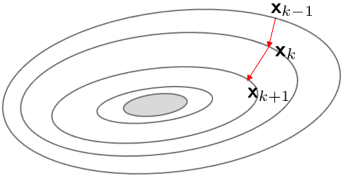
Let $g \equiv 0$. In this case $\mathrm{prox}_g = \mathrm{Id}$.

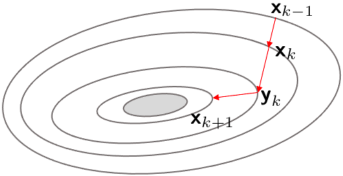
The *path* taken by the iterates $(\mathbf{x}_k)_{k\in\mathbb{N}}$ is determined by the opposite of the gradient direction:

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

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

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





Particular case: Inertia for GD algorithm

Let $\mathrm{g} \equiv 0$. In this case $\mathrm{prox}_{\mathrm{g}} = \mathrm{Id}$.

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

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

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

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

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

Particular case: Inertia for GD algorithm - example

Minimisation problem: Find $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \mathrm{h}_1(\mathbf{x}) + \mathrm{h}_2(\mathbf{x})$

- h_1 is an ℓ_2-norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathrm{h}_1(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x} - \mathbf{z}\|^2$ with $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ and $\mathbf{z} \in \mathbb{R}^M$ are realisations of i.i.d. Gaussian random variables.
- h_2 is the Huber function: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathrm{h}_2(\mathbf{x}) = \sum_{n=1}^N \varphi_n(\mathrm{x}^{(n)})$ with,

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

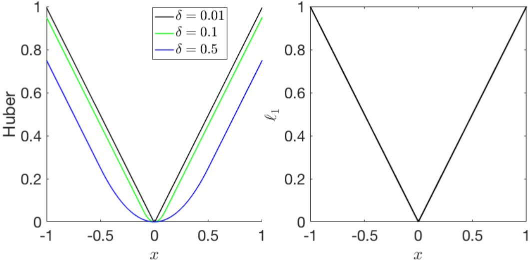
Particular case: Inertia for GD algorithm - example

Minimisation problem: Find $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \mathrm{h}_1(\mathbf{x}) + \mathrm{h}_2(\mathbf{x})$

- h_1 is an ℓ_2-norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathrm{h}_1(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x} - \mathbf{z}\|^2$ with $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ and $\mathbf{z} \in \mathbb{R}^M$ are realisations of i.i.d. Gaussian random variables.
- h_2 is the Huber function: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathrm{h}_2(\mathbf{x}) = \sum_{n=1}^N \varphi_n(\mathrm{x}^{(n)})$ with,

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

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



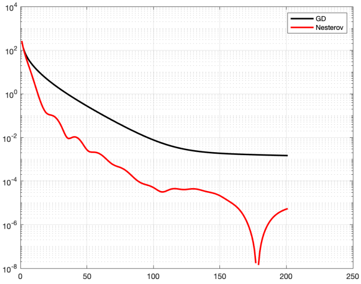
Particular case: Inertia for GD algorithm - example

Minimisation problem: Find $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\, \mathrm{h}_1(\mathbf{x}) + \mathrm{h}_2(\mathbf{x})$

- h_1 is an ℓ_2-norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathrm{h}_1(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x} - \mathbf{z}\|^2$ with $\boldsymbol{\Phi} \in \mathbb{R}^{M\times N}$ 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 $\beta = \|\boldsymbol{\Phi}\|^2 + \frac{1}{\delta}$

☛ $N = 1000$ and $M = 500$



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

Fast Iterative Shrinkage-Thresholding Algorithm (FISTA)

Let $\mathbf{x}_0 \in \operatorname{dom} \mathsf{g}$ and $\mathbf{y}_0 = \mathbf{x}_0$.Let $\theta_0 = 1$.
For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
\text{Let } \gamma_k \in]0, 1/\beta] \\
\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}\Big(\mathbf{y}_k - \gamma_k \nabla \mathsf{h}(\mathbf{y}_k)\Big) \\
\theta_{k+1} = \frac{1+\sqrt{1+4\theta_k^2}}{2} \\
\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k)
\end{array}
\right.
$$

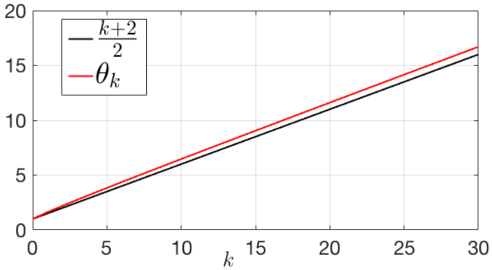
- Adopt the inertia (momentum) strategy proposed by Nesterov

Behaviour of FISTA' momentum parameter

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

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

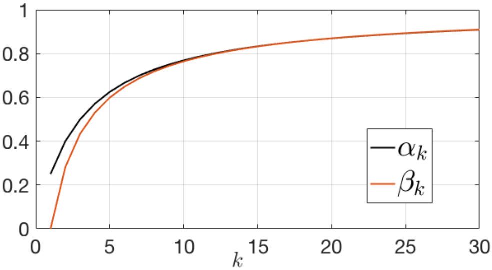
For every $k \in \mathbb{N}$, we have $\theta_k \geqslant \dfrac{k+2}{2}$



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 $(\forall k \geqslant 1) \quad \theta_{k+1} = \frac{1+\sqrt{1+4\theta_k^2}}{2}$

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



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

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

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

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

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

ISSUES:

- No convergence guaranties 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: $\mathbf{z} = \boldsymbol{\Phi}\overline{\mathbf{x}} + \mathbf{w}$

- $\mathbf{w}$ is a realisation on a zero-mean white Gaussian noise
- $\boldsymbol{\Phi} = \mathbf{RD}$ 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 $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ F(\mathbf{x}) + R(\mathbf{x})$

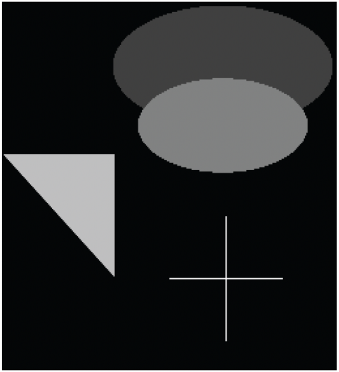
- Data-fidelity: F is an ℓ_2-norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad F(\mathbf{x}) = \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x} - \mathbf{z}\|^2$

  $\rightsquigarrow$ Lipschitz-differentiable

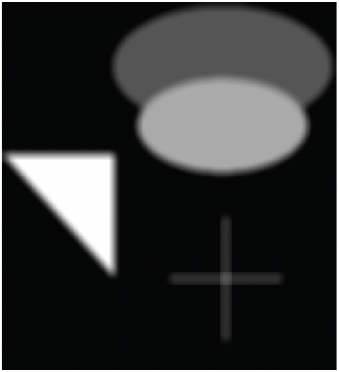
- Regularisation: R is an ℓ_1-norm: $(\forall \mathbf{x} \in \mathbb{R}^N) \quad R(\mathbf{x}) = \eta\|\mathbf{x}\|_1$

  $\rightsquigarrow$ prox_R is the soft-thresholding operator

Example: Image deconvolution [Beck and Teboulle, 2009]



$\overline{\mathbf{x}}$

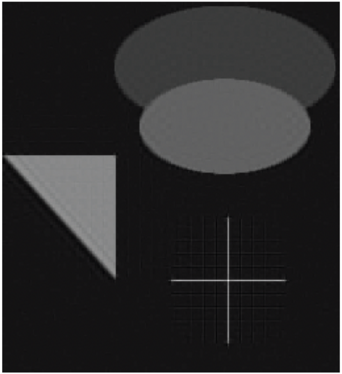


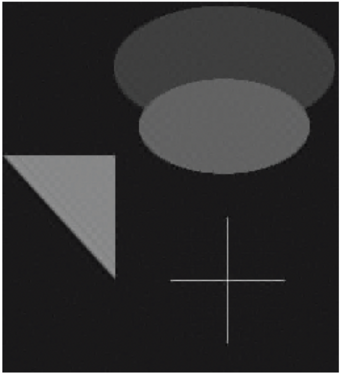
$\mathbf{z} = \Phi\overline{\mathbf{x}} + \mathbf{w}$

Example: Image deconvolution [Beck and Teboulle, 2009]

FB: $\mathbf{x}_k$ with $k = 100$ $\mathbf{x}_k$ with $k = 200$

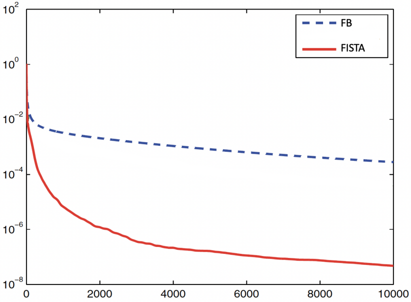
Example: Image deconvolution [Beck and Teboulle, 2009]





FISTA: $\mathbf{x}_k$ with $k = 100$ $\mathbf{x}_k$ with $k = 200$

Example: Image deconvolution [Beck and Teboulle, 2009]



$$f(\mathbf{x}_k) - f(\mathbf{x}^\star)$$

Convergent FISTA

"general" FISTA [Beck and Teboulle, 2009]

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

Let $\mathbf{x}_0 \in \operatorname{dom} \mathsf{g}$ and $\mathbf{y}_0 = \mathbf{x}_0$. Let $\theta_0 = 1$.
For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 1/\beta]$
- $\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}\left(\mathbf{y}_k - \gamma_k \nabla \mathsf{h}(\mathbf{y}_k)\right)$
- Choose θ_{k+1} using rule (Cond$_\theta$)
- $\mathbf{y}_{k+1} = \mathbf{x}_{k+1} + \frac{\theta_k - 1}{\theta_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k)$

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

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

"general" FISTA [Beck and Teboulle, 2009]

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

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

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

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

Hence, there exists $\upsilon \in]0,+\infty[$ such that $(\forall k \in \mathbb{N}) \quad \mathsf{f}(\mathbf{x}_k) - \mathsf{f}(\mathbf{x}^\star) \leqslant \frac{\upsilon}{(k+1)^2}$ where υ 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$_\theta$).

REMARK: *Best* choice to obtain the smaller constant υ will be when (Cond$_\theta$) is **saturated**, i.e. when the inequality becomes an equality.

"general" FISTA [Beck and Teboulle, 2009]

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

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

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

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$_\theta$)

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

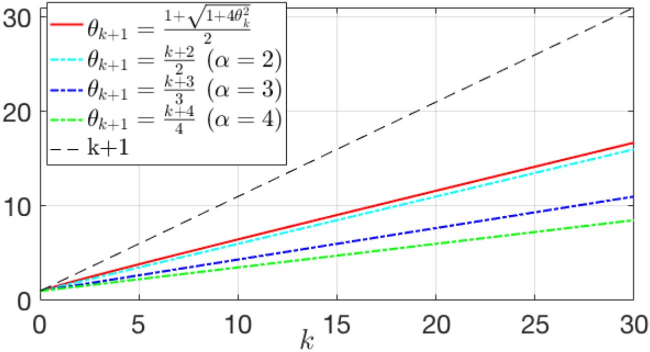
In other words, this is the tightest condition to obtain the best convergence rate.

QUESTION: Are they other (possibly *better*) choices?

Convergent FISTA [Chambolle and Dossal, 2015]

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

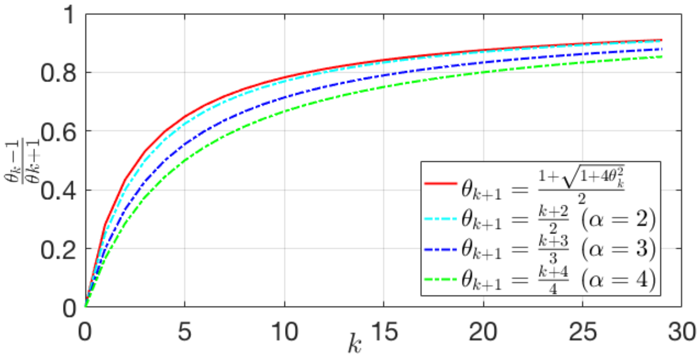
Let $\alpha \geqslant 2$ and let $(\forall k \in \mathbb{N}) \quad \theta_k = \frac{k+\alpha}{\alpha}$.



Convergent FISTA [Chambolle and Dossal, 2015]

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

Let $\alpha \geqslant 2$ 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 $\begin{cases}\theta_0 = 1 \\ (\forall k \geqslant 1) \quad \theta_k^2 - \theta_k \leqslant \theta_{k-1}^2\end{cases}$ (Cond_θ)

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

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

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

CONSEQUENCE: $(\exists \upsilon > 0)(\forall k \in \mathbb{N}) \quad \mathrm{f}(\mathbf{x}_k) - \mathrm{f}(\mathbf{x}^\star) \leqslant \frac{\upsilon}{(k+1)^2}$

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.

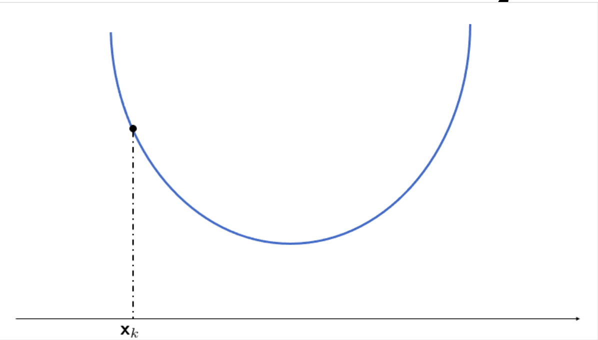
Proof: A complete proof is provided in [Chambolle and Dossal, 2015].

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

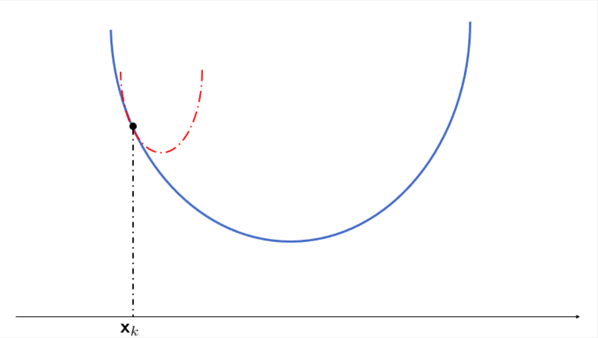
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



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

REMARK: Link between FB iterations and Majorisation-Minimisation methods:
Define $\mathbf{x}_{k+1}$ such that

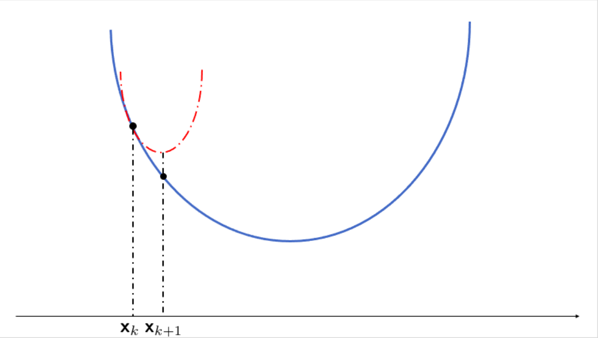
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{\beta}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



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

REMARK: Link between FB iterations and Majorisation-Minimisation methods:
Define $\mathbf{x}_{k+1}$ such that

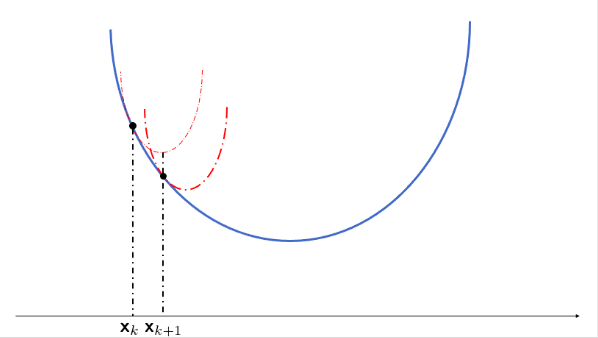
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



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

Remark: Link between FB iterations and Majorisation-Minimisation methods:
Define $\mathbf{x}_{k+1}$ such that

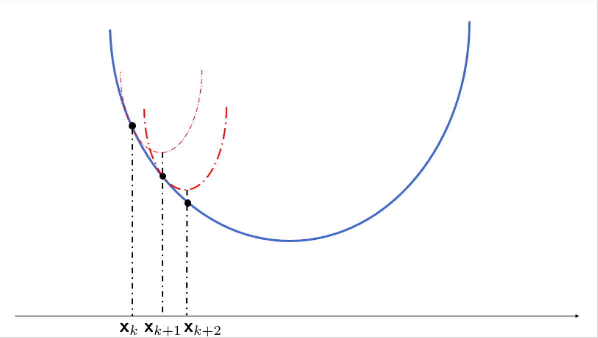
$$\mathrm{f}(\mathbf{x}_{k+1}) \leqslant \mathrm{q}(\mathbf{x}_{k+1}, \mathbf{x}_k) = \mathrm{h}(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla \mathrm{h}(\mathbf{x}_k) \rangle + \frac{\beta}{2} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \mathrm{g}(\mathbf{x}_{k+1})$$



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

REMARK: Link between FB iterations and Majorisation-Minimisation methods:
Define $\mathbf{x}_{k+1}$ such that

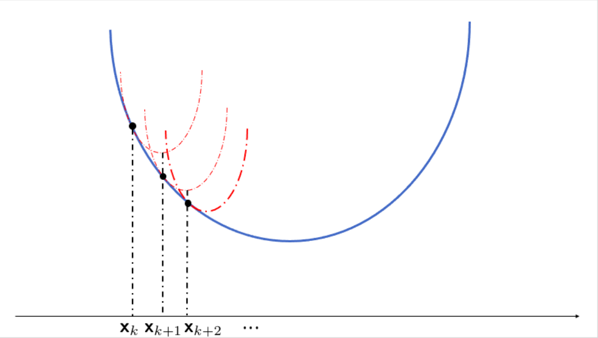
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{\beta}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



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

REMARK: Link between FB iterations and Majorisation-Minimisation methods:
Define $\mathbf{x}_{k+1}$ such that

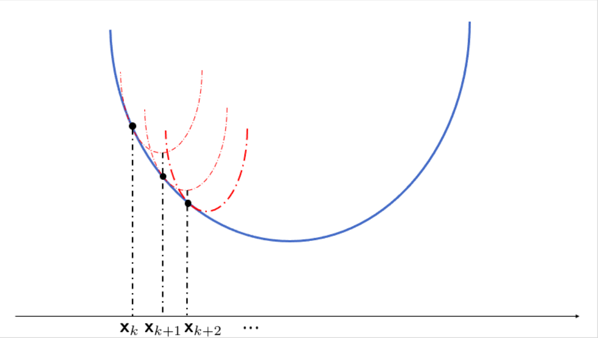
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{\beta}{2}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



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

REMARK: Link between FB iterations and Majorisation-Minimisation methods: Define $\mathbf{x}_{k+1}$ such that

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



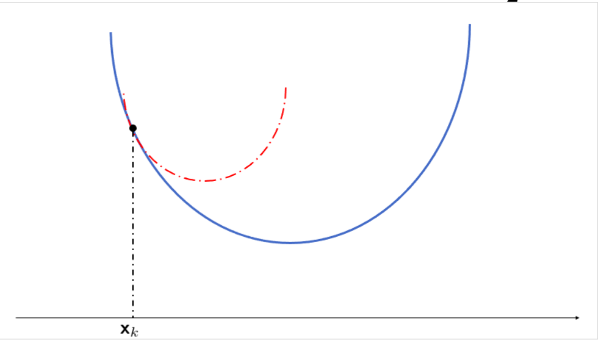
IDEA: Do not use the Lipschitz constant β to determine the step-size γ, but verify that local quadratic majorisation of f is satisfied.

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

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

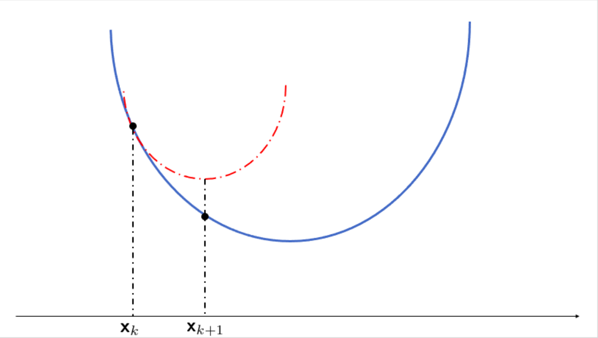


Case when β is unknown: Backtracking (2/2)

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

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

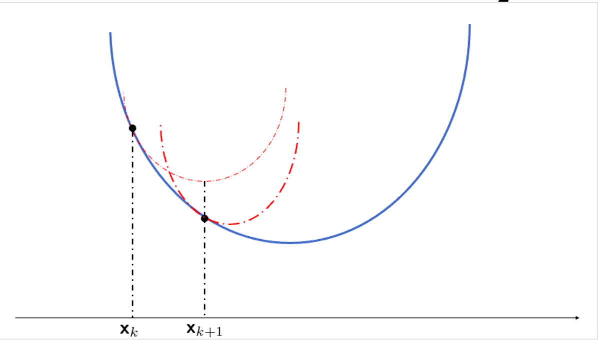


Case when β is unknown: Backtracking (2/2)

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

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

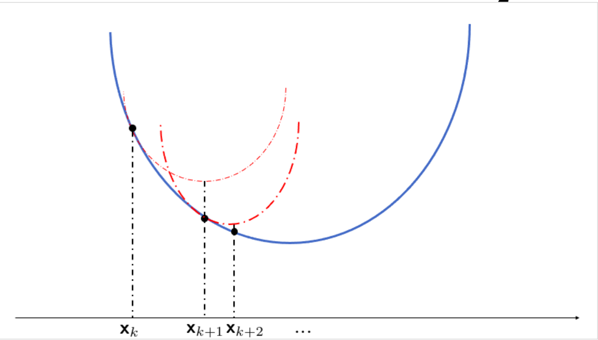


Case when β is unknown: Backtracking (2/2)

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

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



FISTA with Backtracking

Let $\overline{\beta} > 0$, and $\mathbf{x} \in \mathbb{R}^N$. Define $\mathsf{p}_{\overline{\beta}}(\mathbf{x}) = \mathrm{prox}_{\overline{\beta}^{-1}\mathsf{g}}\left(\mathbf{x} - \overline{\beta}^{-1}\nabla \mathsf{h}(\mathbf{x})\right)$
and $(\forall \mathbf{x}^+ \in \mathbb{R}^N) \quad \mathsf{q}_{\overline{\beta}}(\mathbf{x}^+, \mathbf{x}) = \mathsf{h}(\mathbf{x}) + \langle \mathbf{x}^+ - \mathbf{x} \mid \nabla \mathsf{h}(\mathrm{x})\rangle + \frac{\overline{\beta}}{2}\|\mathbf{x}^+ - \mathbf{x}\|^2 + \mathsf{g}(\mathbf{x}^+)$

FISTA with backtracking

Let $\mathbf{x}_0 \in \mathrm{dom}\,\mathsf{g}$ and $\mathbf{y}_0 = \mathbf{x}_0$.Let $\theta_0 = 1$.Let $\beta_0 > 0$ and $\eta > 1$.
For $k = 0, 1, \ldots$

Find the smallest $i_k \in \mathbb{N}$ such that $\begin{cases} \mathsf{f}(\mathsf{p}_{\overline{\beta}}(\mathbf{y}_k) \leqslant \mathsf{q}_{\overline{\beta}}\big(\mathsf{p}_{\overline{\beta}}(\mathbf{y}_k), \mathbf{y}_k\big) \\ \overline{\beta} = \eta^{i_k}\beta_k \end{cases}$

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

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

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

FISTA with Backtracking: How it works

Let $\overline{\beta} > 0$, and $\mathbf{x} \in \mathbb{R}^N$. Define $\mathsf{p}_{\overline{\beta}}(\mathbf{x}) = \mathrm{prox}_{\overline{\beta}^{-1}\mathsf{g}}\left(\mathbf{x} - \overline{\beta}^{-1}\nabla \mathsf{h}(\mathbf{x})\right)$
and $(\forall \mathbf{x}^+ \in \mathbb{R}^N) \quad \mathsf{q}_{\overline{\beta}}(\mathbf{x}^+, \mathbf{x}) = \mathsf{h}(\mathbf{x}) + \langle \mathbf{x}^+ - \mathbf{x} \mid \nabla \mathsf{h}(\mathsf{x})\rangle + \frac{\overline{\beta}}{2}\|\mathbf{x}^+ - \mathbf{x}\|^2 + \mathsf{g}(\mathbf{x}^+)$

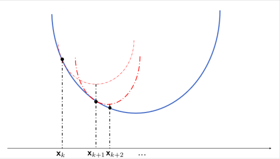
At each iteration $k \in \mathbb{N}$, start with $i_k = 0$ and:

- [Step 1]

  Fix $\overline{\beta} = \eta^{i_k}\beta_k$ and compute $\mathsf{p}_{\overline{\beta}}(\mathbf{y}_k) = \mathrm{prox}_{\overline{\beta}^{-1}\mathsf{g}}(\mathbf{y}_k - \overline{\beta}^{-1}\nabla \mathsf{h}(\mathbf{y}_k))$

- [Condition to move to Step 2]

  If $\mathsf{f}(\mathsf{p}_{\overline{\beta}}(\mathbf{y}_k)) \leqslant \mathsf{q}_{\overline{\beta}}\left(\mathsf{p}_{\overline{\beta}}(\mathbf{y}_k), \mathbf{y}_k\right)$, then go to [Step 2].
  Otherwise go back to [Step 1] with $i_k = i_k + 1$.



- [Step 2]

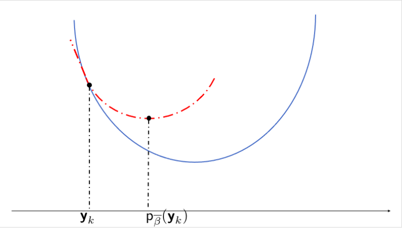
  Update $\beta_{k+1} = \eta^{i_k}\beta_k$, $\mathbf{x}_{k+1} = \mathsf{p}_{\beta_{k+1}}(\mathbf{y}_k)$.

  Compute FISTA inertia step.

FISTA with Backtracking: Remarks

- The approximations $q_{\overline{\beta}}(\cdot, \mathbf{y}_k)$ 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 $p_{\overline{\beta}}(\mathbf{y}_k)$:
$$f\big(p_{\overline{\beta}}(\mathbf{y}_k)\big) \leqslant q_{\overline{\beta}}\big(p_{\overline{\beta}}(\mathbf{y}_k), \mathbf{y}_k\big)$$
  $\rightsquigarrow$ 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 $\mathsf{f} = \mathsf{h} + \mathsf{g}$

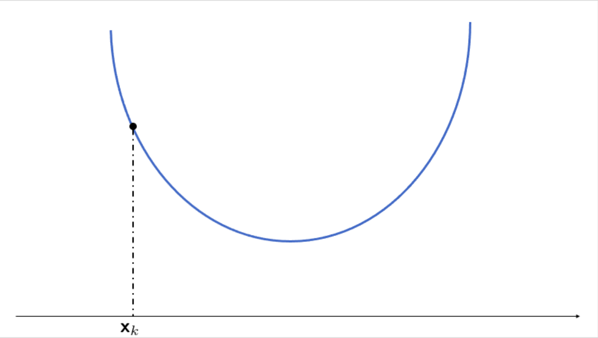
$$
\begin{aligned}
\mathbf{x}_{k+1} &= \operatorname{prox}_{\gamma_k \mathsf{g}}\left(\mathbf{x}_k - \gamma_k \nabla \mathsf{h}(\mathbf{x}_k)\right) \\
&= \underset{\mathbf{x}}{\operatorname{argmin}}\ \mathsf{g}(\mathbf{x}) + \tfrac{1}{2\gamma_k}\|\mathbf{x} - \mathbf{x}_k + \gamma_k \nabla \mathsf{h}(\mathbf{x}_k)\|^2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

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

From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

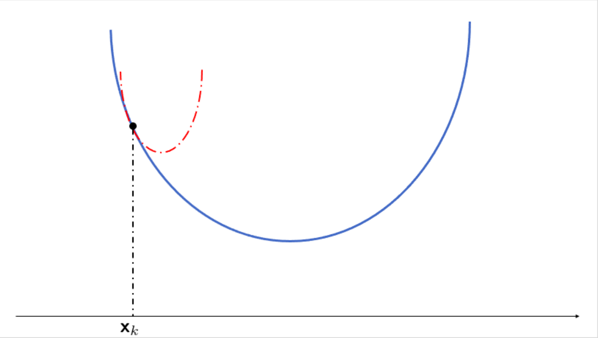
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{1}{2\gamma_k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



From FB ...

Remark: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

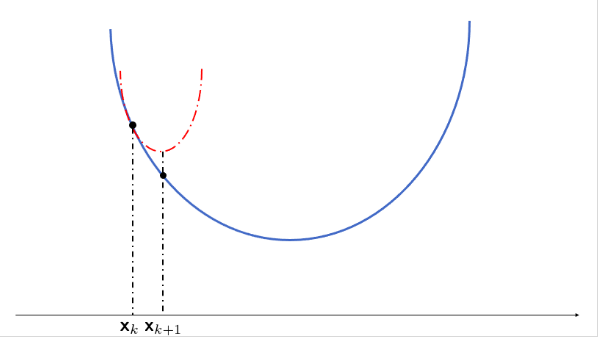
$$\mathrm{f}(\mathbf{x}_{k+1}) \leqslant \mathrm{q}(\mathbf{x}_{k+1}, \mathbf{x}_k) = \mathrm{h}(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla \mathrm{h}(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + \mathrm{g}(\mathbf{x}_{k+1})$$



From FB ...

Remark: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

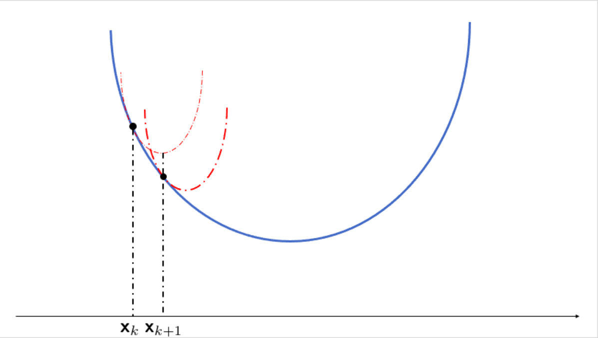
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{1}{2\gamma_k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

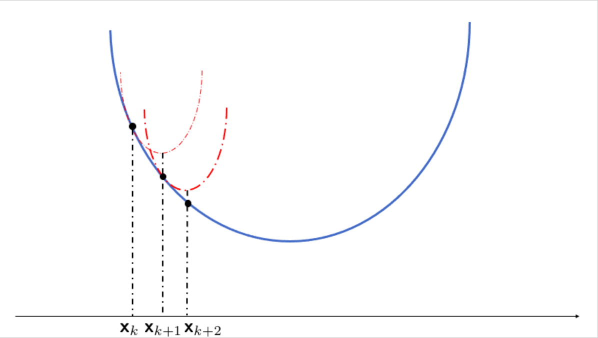
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{1}{2\gamma_k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



From FB ...

Remark: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

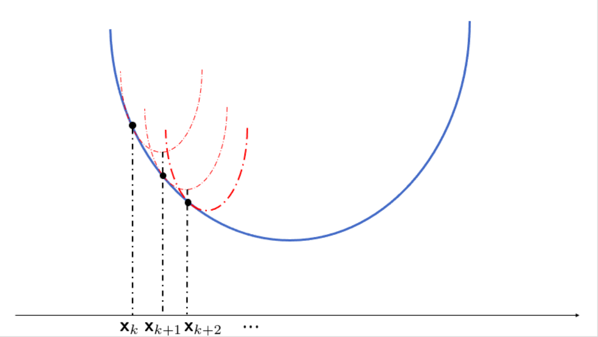
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k)\rangle + \frac{1}{2\gamma_k}\|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



From FB ...

Remark: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

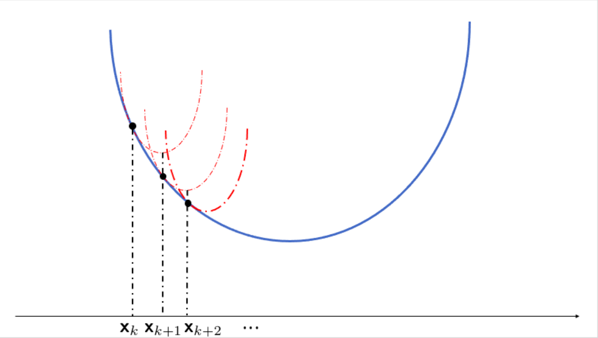
$$f(\mathbf{x}_{k+1}) \leqslant q(\mathbf{x}_{k+1}, \mathbf{x}_k) = h(\mathbf{x}_k) + \langle \mathbf{x}_{k+1} - \mathbf{x}_k \mid \nabla h(\mathbf{x}_k) \rangle + \frac{1}{2\gamma_k} \|\mathbf{x}_{k+1} - \mathbf{x}_k\|^2 + g(\mathbf{x}_{k+1})$$



From FB ...

REMARK: FB iterations as Majorisation-Minimisation scheme:
Define $\mathbf{x}_{k+1}$ such that

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



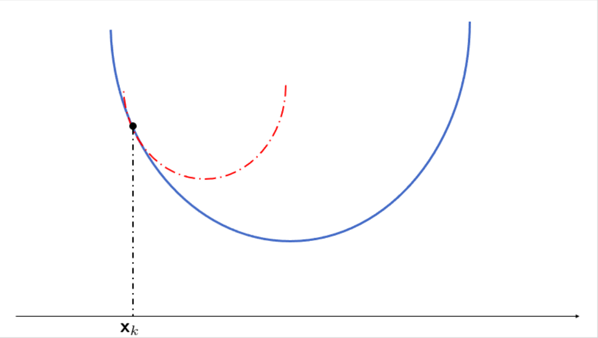
IDEA: Do not use the Lipschitz constant β to determine the step-size γ, but verify that local quadratic majorisation of f is satisfied.

... 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

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



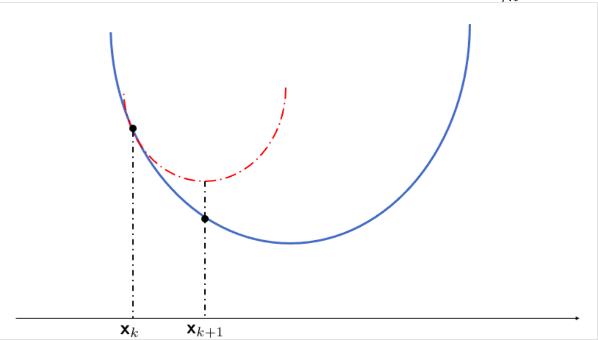
- $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \|\mathbf{x}\|^2_{\mathbf{A}_k} = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x}\rangle$

... 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

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



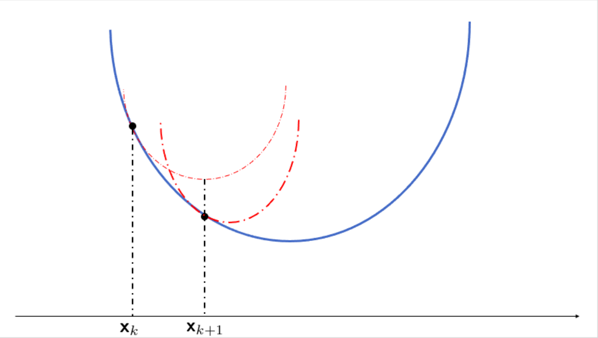
- $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \|\mathbf{x}\|^2_{\mathbf{A}_k} = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x}\rangle$

... 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

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



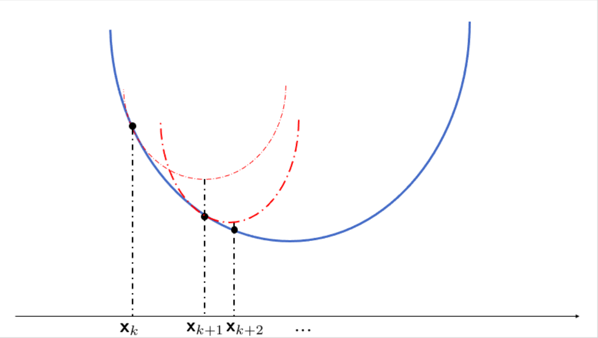
- $(\forall \mathbf{x} \in \mathbb{R}^N) \quad \|\mathbf{x}\|^2_{\mathbf{A}_k} = \langle \mathbf{x} \mid \mathbf{A}_k \mathbf{x}\rangle$

... 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

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



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

Variable Metric FB

OBJECTIVE: Find a minimiser of $\mathsf{f} = \mathsf{h} + \mathsf{g}$

Let $\mathbf{x}_0 \in \operatorname{dom} \mathsf{g}$, and $(\forall k \in \mathbb{N}) \quad \gamma_k \in]0, 2[$
For $k = 0, 1, \ldots$
$\quad \mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}^{\mathbf{A}_k} \left(\mathbf{x}_k - \gamma_k \mathbf{A}_k^{-1} \nabla \mathsf{h}(\mathbf{x}_k) \right)$

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

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

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:* $\mathbf{z} = |\boldsymbol{\Phi}\overline{\mathbf{x}}| + \mathbf{w}$

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

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

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

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

FB & Non-convex problems

OBJECTIVE: Find a minimiser of $\mathsf{f} = \mathsf{h} + \mathsf{g}$

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

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

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

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

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

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

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

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

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

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:* $g(\mathbf{x}) = \sum_{n=1}^{N} \log(|x^{(n)}| + \varepsilon)$
- *Cauchy penalisation:* $g(\mathbf{x}) = \sum_{n=1}^{N} \log(|x^{(n)}|^2 + \varepsilon)$
- 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 $\mathbf{x}^\star$ of f (i.e., $\mathbf{0} \in \partial f(\mathbf{x}^\star)$)
  - 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

[Attouch, Bolte, Svaiter, 2011] [Chouzenoux, Pesquet, Repetti, 2014]

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

$$\big(\forall \mathbf{t} \in \partial \mathrm{f}(\mathbf{x})\big) \qquad \|\mathbf{t}\| \geqslant \kappa |\mathrm{f}(\mathbf{x}) - \xi|^{\theta},$$

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

- ⋆ 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: $\mathbf{z} = \mathbf{\Phi}\overline{\mathbf{x}} + \mathbf{w}$

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

MINIMISATION PROBLEM: Find $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\, F(\mathbf{x}) + R(\mathbf{x})$

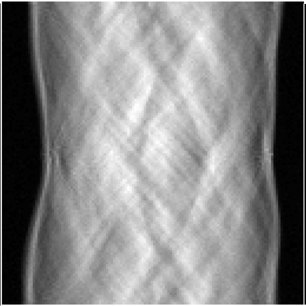
- Data-fidelity: $F(\mathbf{x}) = \sum_{m=1}^{M} h_m([\mathbf{\Phi}\mathbf{x}]^{(m)})$ with $h_m(\mathrm{u}) = \frac{1}{2}\frac{(\mathrm{u}-\mathrm{z}^{(m)})^2}{a^{(m)}\mathrm{u}+b^{(m)}} + \frac{1}{2}\log(a^{(m)}\mathrm{u}+b^{(m)})$

  $\rightsquigarrow$ Lipschitz-differentiable
- Regularisation: $R(\mathbf{x}) = \eta\|\mathbf{\Psi}\mathbf{x}\|_1 + \iota_{[0,+\infty[^2}(\mathbf{x})$ with $\eta > 0$, and $\mathbf{\Psi}$ redundant (undecimated) wavelet (DB8 over three resolution levels)

  $\rightsquigarrow$ prox_R computed with sub-iterations (dual FB algorithm)

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

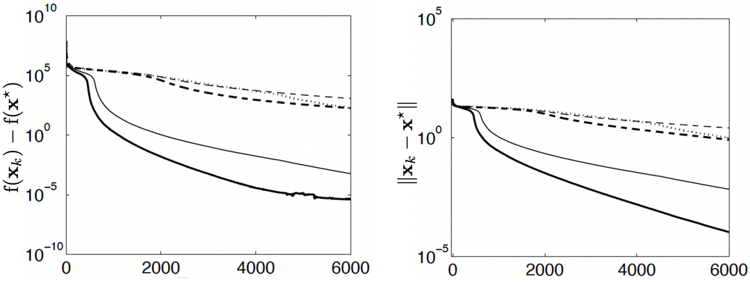
$\bar{\mathbf{x}}$



$\mathbf{z}$

$\mathbf{x}^{\star}$ (VMFB)

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 $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \mathsf{h}(\mathbf{x}) + \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x})$.

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

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

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

$$\underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{minimise}}\ \underbrace{\frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x}-\mathbf{z}\|^2}_{\mathsf{h}(\mathbf{x})} + \underbrace{\iota_{\mathcal{C}}(\mathbf{x})}_{\mathsf{f}(\mathbf{x})} + \underbrace{\eta\|\boldsymbol{\Psi}\mathbf{x}\|_1}_{\mathsf{g}(\mathbf{L}\mathbf{x})}$$

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

Minimization problem

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \mathsf{h}(\mathbf{x}) + \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x})$.

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

Use FB algorithm?

For $k = 0, 1, \ldots$

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

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

➤ Use primal-dual methods

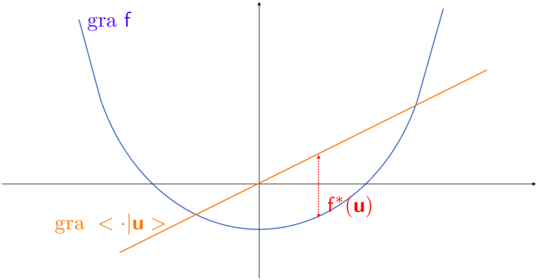
Fenchel-Legendre duality

Conjugate function

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

$$\mathsf{f}^*\colon \begin{array}{ccc} \mathbb{R}^N & \to & [-\infty,+\infty] \\ \mathbf{u} & \mapsto & \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - \mathsf{f}(\mathbf{x}) \end{array}$$

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

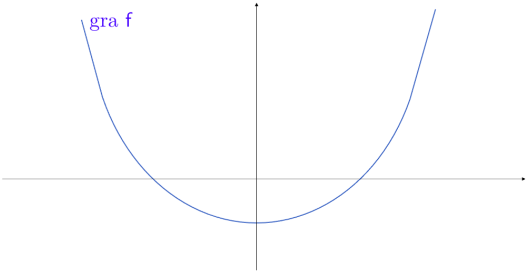


Conjugate function

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

$$f^*\colon \quad \begin{array}{rcl} \mathbb{R}^N & \to & [-\infty,+\infty] \\ \mathbf{u} & \mapsto & \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - f(\mathbf{x}) \end{array}$$

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

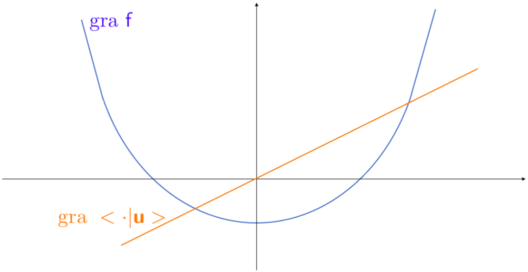


Conjugate function

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

$$\mathsf{f}^*\colon \quad \mathbb{R}^N \to [-\infty,+\infty], \quad \mathbf{u} \mapsto \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - \mathsf{f}(\mathbf{x})$$

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

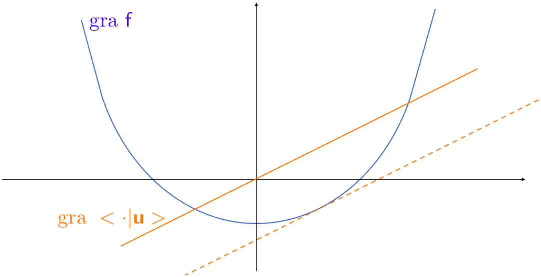


Conjugate function

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

$$f^*\colon \begin{array}{ccc} \mathbb{R}^N & \to & [-\infty,+\infty] \\ \mathbf{u} & \mapsto & \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - f(\mathbf{x}) \end{array}$$

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

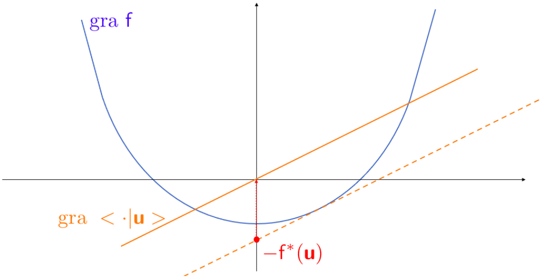


Conjugate function

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

$$f^*\colon \quad \mathbb{R}^N \to [-\infty,+\infty], \qquad \mathbf{u} \mapsto \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - f(\mathbf{x})$$

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



Conjugate function

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

$$\begin{array}{rccl} \mathrm{f}^*\colon & \mathbb{R}^N & \to & [-\infty,+\infty] \\ & \mathbf{u} & \mapsto & \sup_{\mathbf{x}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u}\rangle - \mathrm{f}(\mathbf{x}) \end{array}$$

REMARK:

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

  Proof: Assume that $(\exists \mathbf{u} \in \mathbb{R}^N)$ such that $\mathrm{f}^*(\mathbf{u}) = -\infty$.
  So, we have $\sup_{\mathbf{x}} \langle \mathbf{x} \mid \mathbf{u}\rangle - \mathrm{f}(\mathbf{x}) = -\infty$.
  This happens iff $(\forall \mathbf{x} \in \mathbb{R}^N)$ $\mathrm{f}(\mathbf{x}) = +\infty$, i.e. $\operatorname{dom} \mathrm{f} = \varnothing$. $\square$

Some properties

Fenchel-Young Inequality

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

$$\big(\forall (\mathbf{x}, \mathbf{u}) \in (\mathbb{R}^N)^2\big) \quad \mathrm{f}(\mathbf{x}) + \mathrm{f}^*(\mathbf{u}) \geqslant \langle \mathbf{x} \mid \mathbf{u} \rangle$$

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

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

$\square$

Some properties

Fenchel-Young Inequality

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

$$\big(\forall(\mathbf{x},\mathbf{u}) \in (\mathbb{R}^N)^2\big) \quad \mathrm{f}(\mathbf{x}) + \mathrm{f}^*(\mathbf{u}) \geqslant \langle \mathbf{x} \mid \mathbf{u} \rangle$$

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

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

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

In particular, $\sup_{\mathbf{u}\in\mathbb{R}^N} \langle \mathbf{x} \mid \mathbf{u} \rangle - \mathrm{f}^*(\mathbf{u}) \leqslant \mathrm{f}(\mathbf{x})$

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

Some properties

Fenchel-Young Inequality

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

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

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

Proof: We have $f^{***} = (f^*)^{**} \leqslant f^*$.
In addition, since $f^{**} \leqslant f$, using second property we have $(f^{**})^* \geqslant f^*$.
Hence the equality. $\square$

Some properties

Fenchel-Young Inequality

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

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

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

Moreau-Fenchel Theorem

Let $f\colon \mathbb{R}^N \to]-\infty,+\infty]$ 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

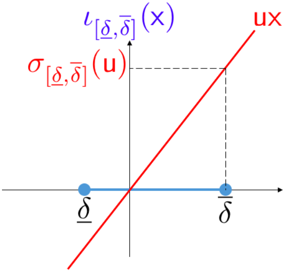
$$(\forall \mathbf{u} \in \mathbb{R}^N) \qquad \sigma_{\mathcal{C}}(\mathbf{u}) = \sup_{\mathbf{x}\in\mathcal{C}} \langle \mathbf{x} \mid \mathbf{u} \rangle$$

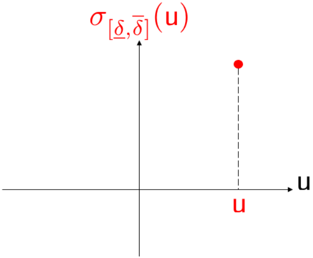
Example: Support function

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

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

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



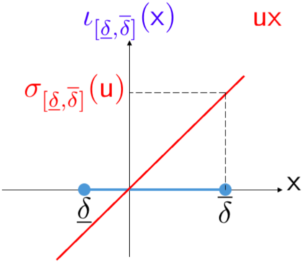


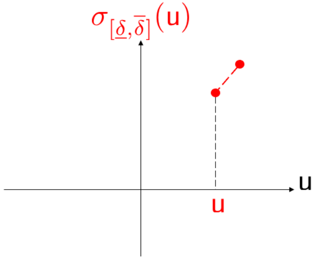
Example: Support function

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

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

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



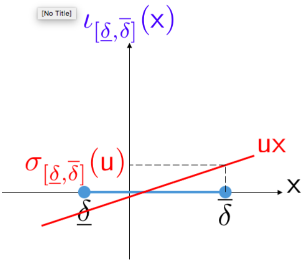


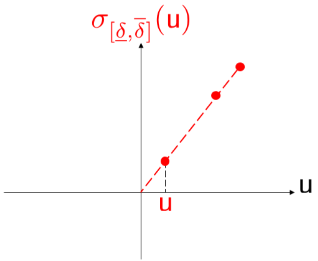
Example: Support function

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

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

Example: $\mathcal{C} = [\underline{\delta}, \overline{\delta}]$



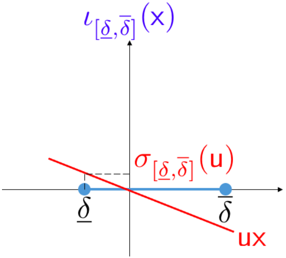


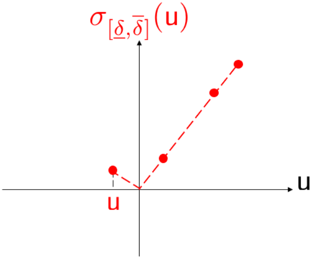
Example: Support function

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

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

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



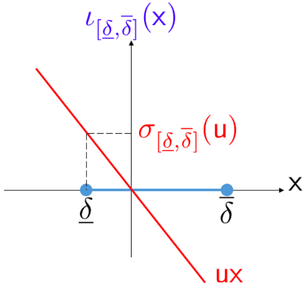


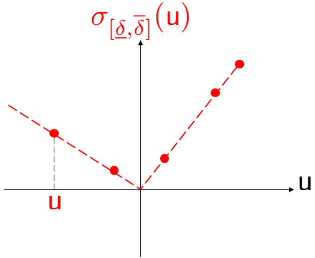
Example: Support function

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

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

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





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

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

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 $-\infty \leqslant \underline{\delta} < \overline{\delta} \leqslant +\infty$, and $\mathcal{C} = [\underline{\delta}, \overline{\delta}]$. Then

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

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

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

More generally, for $\mathrm{f} = \delta\|\cdot\|_1$, we have $\mathrm{f}^* = \iota_{\mathcal{B}_\infty(\mathbf{0},\delta)}$

Example: norms

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

Proof: We consider the case $\mathsf{u} \geqslant 0$ *(the case* $\mathsf{u} \leqslant 0$ *can be handled analogously).*

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

The function $\varphi\colon \mathsf{x} \in [0,+\infty[\mapsto \mathsf{ux} - \frac{1}{p}\mathsf{x}^p$ being concave, we can obtain its maximum using the Fermat's rule:
$$\partial\varphi(\mathsf{x}) = 0 \quad \Leftrightarrow \quad \mathsf{u} - \mathsf{x}^{p-1} = 0 \quad \Leftrightarrow \quad \mathsf{x} = \mathsf{u}^{\frac{1}{p-1}}$$

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

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

Example: norms

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

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

REMARKS:

- Let $\mathsf{f} = \|\cdot\|_p^p$. The conjugate function of f is $\mathsf{f}^* = \|\cdot\|_q^q$ 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 $\mathrm{f}\colon \mathbb{R}^N \to]-\infty], \mathrm{g}\colon \mathbb{R}^M \to]-\infty, +\infty]$ and $\mathbf{L} \in \mathbb{R}^{M\times N}$.
The primal problem associated with the composite function $\mathrm{f} + \mathrm{g} \circ \mathbf{L}$ is

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

and its dual problem is

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

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

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

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

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

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

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

Hence $f(\mathbf{x}) + g(\mathbf{L}\mathbf{x}) + f^*(-\mathbf{L}^*\mathbf{u}) + g^*(\mathbf{u}) \geqslant \langle \mathbf{x} \mid -\mathbf{L}^*\mathbf{u}\rangle + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u}\rangle = 0.$ $\square$

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

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

Strong duality

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

REMARKS:

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

Fenchel-Rockafellar duality: weak/strong duality

Weak duality

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

Strong duality

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

REMARKS:

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

  How to compute $\mu^* = \mathrm{f}^*(-\mathbf{L}^*\mathbf{u}^\star) + \mathrm{g}^*(\mathbf{u}^\star)$*?*

  ☛ We have $\begin{cases} \mathrm{f}^*(-\mathbf{L}^*\mathbf{u}^\star) = \langle -\mathbf{L}^*\mathbf{u}^* \mid \mathbf{x}^\star\rangle - \mathrm{f}(\mathbf{x}^\star) \\ \mathrm{g}^*(\mathbf{u}^\star) = \langle \mathbf{u}^* \mid \mathbf{L}\mathbf{x}^\star\rangle - \mathrm{g}(\mathbf{L}\mathbf{x}^\star) \end{cases}$

Dual algorithms

Use duality to approximate proximity operators

Primal problem: Find $\mathbf{x}^\star = \operatorname{prox}_{\mathsf{f}+\mathsf{g}\circ\mathbf{L}}(\mathbf{y}) = \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \mathsf{f}(\mathbf{x})+\frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2+\mathsf{g}(\mathbf{L}\mathbf{x})$

Dual problem: Find $\mathbf{u}^\star = \underset{\mathbf{u}\in\mathbb{R}^M}{\operatorname{Argmin}}\ (\mathsf{f}+\frac{1}{2}\|\cdot-\mathbf{y}\|^2)^*(-\mathbf{L}^*\mathbf{u})+\mathsf{g}^*(\mathbf{u})$

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

Proof: Recall that the Fenchel-Rockafellar duality gives:

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

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

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

Use duality to approximate proximity operators

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

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

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

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

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

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

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

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

Dual FB algorithm

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

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

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

Choose $\mathbf{u}_0 \in \mathbb{R}^M$, and, for every $k \in \mathbb{N}$, choose $\gamma_k \in]0, 2/\|\mathbf{L}\|^2[$.
For $k = 0, 1, \ldots$

$$\left\lfloor \begin{array}{l} \mathbf{x}_k = \mathrm{prox}_{\mathsf{f}}\left(\mathbf{y} - \mathbf{L}^*\mathbf{u}_k\right) \\ \mathbf{u}_{k+1} = \mathrm{prox}_{\gamma_k \mathsf{g}^*}\left(\mathbf{u}_k + \gamma_k \mathbf{L}\mathbf{x}_k\right) \end{array} \right.$$

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 $\mathbf{x}^* = \mathrm{prox}_{\mathsf{f}}(\mathbf{y} - \mathbf{L}^*\mathbf{u}^*)$.

[Combettes, Dung, Vu, 2011]

Moreau proximal decomposition theorem

Question: How to compute $\operatorname{prox}_{\sigma \mathrm{g}^*}$?

Moreau decomposition theorem:

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

Examples:

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

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

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

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

Dual FB algorithm: Example to denoising problem

Gaussian denoising problem: $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$
where $\mathbf{w} \in \mathbb{R}^N$ is a realisation of a white Gaussian noise with variance $\sigma > 0$

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

- **Example of regularisation:** $g(\mathbf{L}\mathbf{x}) = \|\mathbf{x}\|_{\mathrm{TV}}$

Anisotropic total variation

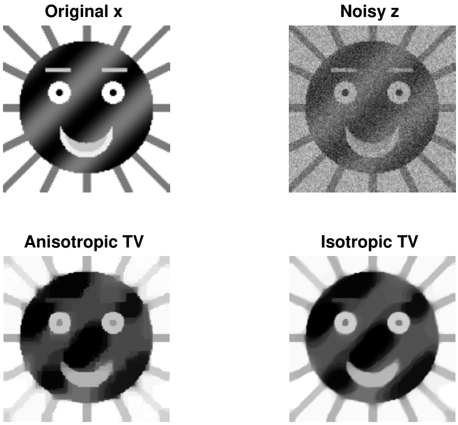
- Sparse transform: $\mathbf{L} = [\mathbf{L}_1^\top, \mathbf{L}_2^\top]^\top \in \mathbb{R}^{2N\times N}$
- $\|\mathbf{x}\|_{\mathrm{TV}} = \|\mathbf{L}\mathbf{x}\|_1 = \|\mathbf{L}_1\mathbf{x}\|_1 + \|\mathbf{L}_2\mathbf{x}\|_1$

Isotropic total variation

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

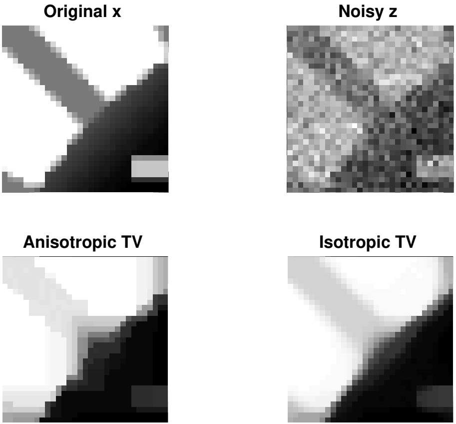
Dual FB algorithm: Example to denoising problem

GAUSSIAN DENOISING PROBLEM : $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$
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: $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$
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 $\mathbf{x}^\star = \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \frac{1}{2}\|\mathbf{x}-\mathbf{y}\|^2 + \sum_{i=1}^{I} \mathrm{g}_i(\mathbf{L}_i\mathbf{x})$

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

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

For $k = 0, 1, \ldots$

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

$\quad$ for $i = 1, \ldots, I$

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

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

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

Primal-Dual algorithms

Problem formulation

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

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

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

Proof: Recall that the Fenchel-Rockafellar duality gives:

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

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

To obtain the dual problem, use $\mathsf{F} = \mathsf{h} + \mathsf{f}$ and $\mathsf{G} = \mathsf{g}$. $\square$

Problem formulation

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

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

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

Proof: Recall that the Fenchel-Rockafellar duality gives:

(P) $\underset{\mathbf{x}}{\text{minimise}}\ \mathsf{F}(\mathbf{x}) + \mathsf{G}(\mathbf{L}\mathbf{x})$

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

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

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

Lagrangian and Karush-Kuhn-Tucker conditions

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

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

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

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

Lagrangian and Karush-Kuhn-Tucker conditions

PRIMAL PROBLEM: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \mathsf{h}(\mathbf{x}) + \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x})$

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

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

$$\text{Find } (\mathbf{x}^\star, \mathbf{u}^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \underset{\mathbf{u}\in\mathbb{R}^M}{\max}\ \mathsf{h}(\mathbf{x}) + \mathsf{f}(\mathbf{x}) - \mathsf{g}^*(\mathbf{u}) + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u}\rangle$$

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

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

$\rightsquigarrow$ Look at the Lagrangian saddle point problem and derive optimal conditions for $\mathbf{x}^\star$, and for $\mathbf{u}^\star$ *alternatively*

$\rightsquigarrow$ These are called KKT conditions

Lagrangian and Karush-Kuhn-Tucker conditions

Primal problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \mathrm{h}(\mathbf{x}) + \mathrm{f}(\mathbf{x}) + \mathrm{g}(\mathbf{L}\mathbf{x})$

Dual problem: Find $\mathbf{u}^\star \in \underset{\mathbf{u}\in\mathbb{R}^M}{\operatorname{Argmin}}\ (\mathrm{h}+\mathrm{f})^*(-\mathbf{L}^*\mathbf{u}) + \mathrm{g}^*(\mathbf{u})$

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

$$\text{Find } (\mathbf{x}^\star, \mathbf{u}^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\ \underset{\mathbf{u}\in\mathbb{R}^M}{\max}\ \mathrm{h}(\mathbf{x}) + \mathrm{f}(\mathbf{x}) - \mathrm{g}^*(\mathbf{u}) + \langle \mathbf{L}\mathbf{x} \mid \mathbf{u}\rangle$$

Karush-Kuhn-Tucker conditions

Assume that $\operatorname{dom} \mathrm{g} \cap \mathbf{L}(\operatorname{dom} \mathrm{f}) \neq \varnothing$.
$(\mathbf{x}^\star, \mathbf{u}^\star) \in \mathbb{R}^N \times \mathbb{R}^M$ is a solution to the Primal-Dual problem if and only if

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

From KKT to fixed-point equations...

KKT:

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

From KKT to fixed-point equations...

KKT:

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

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

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

From KKT to fixed-point equations...

KKT:

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

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

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

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

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

From KKT to fixed-point equations...

KKT:

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

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

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

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

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

From KKT to fixed-point equations...

KKT:

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

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

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

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

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

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

From KKT to fixed-point equations...

KKT:

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

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

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

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

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

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

From KKT to fixed-point equations...

KKT:

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

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

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

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

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

$$\Leftrightarrow \begin{cases} \mathbf{x}^\star = \mathrm{prox}_{\tau \mathrm{f}}\Big(\mathbf{x}^\star - \tau\big(\mathbf{L}^* \mathbf{u}^\star + \nabla \mathrm{h}(\mathbf{x}^\star)\big)\Big) \\ \mathbf{u}^\star = \mathrm{prox}_{\sigma \mathrm{g}^*}\Big(\mathbf{u}^\star + \sigma \mathbf{L}(2\mathbf{x}^\star - \mathbf{x}^\star)\Big) \end{cases} \qquad \rightsquigarrow \textit{Fixed-point equations}$$

Fixed-point algorithm

From the fixed-point equations:

$$\begin{cases} \mathbf{x}^\star = \operatorname{prox}_{\tau \mathsf{f}}\Big(\mathbf{x}^\star - \tau\big(\mathbf{L}^*\mathbf{u}^\star + \nabla \mathsf{h}(\mathbf{x}^\star)\big)\Big) \\ \mathbf{u}^\star = \operatorname{prox}_{\sigma \mathsf{g}^*}\Big(\mathbf{u}^\star + \sigma \mathbf{L}(2\mathbf{x}^\star - \mathbf{x}^\star)\Big) \end{cases}$$

We can derive a fixed-point algorithm:

For $k = 0, 1, \ldots$

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

REMARKS:

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

$$\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma \mathsf{g}}(\mathbf{x}_k - \gamma \nabla \mathsf{h}(\mathbf{x}_k)) \quad \text{“}\underset{k\to\infty}{\longrightarrow}\text{”} \quad \mathbf{x}^\star = \operatorname{prox}_{\gamma \mathsf{g}}(\mathbf{x}^\star - \gamma \nabla \mathsf{h}(\mathbf{x}^\star))$$

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 $k = 0, 1, \ldots$

$$\mathbf{x}_{k+1} = \underbrace{\mathrm{prox}_{\tau \mathrm{f}}}_{\text{backward step}} \Big(\underbrace{\mathbf{x}_k - \tau\big(\mathbf{L}^* \mathbf{u}_k + \nabla \mathrm{h}(\mathbf{x}_k)\big)}_{\text{forward step}}\Big) \quad \rightsquigarrow \quad \text{“primal” update}$$

$$\mathbf{u}_{k+1} = \underbrace{\mathrm{prox}_{\sigma \mathrm{g}^*}}_{\text{backward step}} \Big(\underbrace{\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{forward step}}\Big) \quad \rightsquigarrow \quad \text{“dual” update}$$

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 $k = 0, 1, \dots$

$$
\left\lfloor
\begin{array}{ll}
\mathbf{x}_{k+1} = \underbrace{\mathrm{prox}_{\tau \mathsf{f}}}_{\text{backward step}} \Big(\underbrace{\mathbf{x}_k - \tau \big(\mathbf{L}^* \mathbf{u}_k + \nabla \mathsf{h}(\mathbf{x}_k) \big)}_{\text{forward step}} \Big) & \rightsquigarrow \text{“primal” update} \\
\mathbf{u}_{k+1} = \underbrace{\mathrm{prox}_{\sigma \mathsf{g}^*}}_{\text{backward step}} \Big(\underbrace{\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)}_{\text{forward step}} \Big) & \rightsquigarrow \text{“dual” update}
\end{array}
\right.
$$

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

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

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

Second approach: Condat-Vu algorithm as FB

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

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

Primal update

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

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

Second approach: Condat-Vu algorithm as FB

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

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

Primal update

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

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

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

Second approach: Condat-Vu algorithm as FB

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

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

Primal update

FB update: $\mathbf{x}_{k+1} = \mathrm{prox}_{\tau \mathsf{f}}\Big(\mathbf{x}_k - \tau\big(\nabla \mathsf{h}(\mathbf{x}_k) + \mathbf{L}^*\mathbf{u}_k\big)\Big)$

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

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

PRIMAL UPDATE

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

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

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

Primal update

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

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**!

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

Second approach: Condat-Vu algorithm as FB

Find $(\mathbf{x}^\star, \mathbf{u}^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}} \ \max_{\mathbf{u}\in\mathbb{R}^M} \mathcal{L}(\mathbf{x}, \mathbf{u})$

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

Primal update

FB update: $\mathbf{x}_{k+1} = \mathrm{prox}_{\tau \mathsf{f}} \Big(\mathbf{x}_k - \tau\big(\nabla \mathsf{h}(\mathbf{x}_k) + \mathbf{L}^*\mathbf{u}_k\big)\Big)$

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**!

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

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

Step-size and convergence of Condat-Vu algorithm

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

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

where $\mathsf{h}\colon \mathbb{R}^N \to \mathbb{R}$ is convex and β-Lipschitz differentiable, $\mathsf{f} \in \Gamma_0(\mathbb{R}^N)$, $\mathsf{g} \in \Gamma_0(\mathbb{R}^M)$ 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 $k = 0, 1, \dots$

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

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.

[Vu, 2013][Condat, 2013]

Particular cases

Chambolle-Pock algorithm: $h \equiv 0$ [Chambolle & Pock, 2011]

Problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\, f(\mathbf{x}) + g(\mathbf{L}\mathbf{x})$

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

For $k = 0, 1, \ldots$

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

Particular cases

Chambolle-Pock algorithm: $\mathsf{h} \equiv 0$ [Chambolle & Pock, 2011]

Problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x})$

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

For $k = 0, 1, \ldots$

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

Douglas-Rachford algorithm: $h \equiv 0$, $\mathbf{L} = \mathrm{Id}$ and $\tau = 1/\sigma$

Problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{x})$

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

For $k = 0, 1, \ldots$

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

Particular cases

Chambolle-Pock algorithm: $h \equiv 0$ [Chambolle & Pock, 2011]

Problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{L}\mathbf{x})$

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

For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
\mathbf{x}_{k+1} = \operatorname{prox}_{\tau \mathsf{f}}\left(\mathbf{x}_k - \tau \mathbf{L}^* \mathbf{u}_k\right) \\
\mathbf{u}_{k+1} = \operatorname{prox}_{\sigma \mathsf{g}^*}\left(\mathbf{u}_k + \sigma \mathbf{L}(2\mathbf{x}_{k+1} - \mathbf{x}_k)\right)
\end{array}
\right.
$$

Douglas-Rachford algorithm: $h \equiv 0$, $\mathbf{L} = \mathrm{Id}$ and $\tau = 1/\sigma$

Problem: Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \mathsf{f}(\mathbf{x}) + \mathsf{g}(\mathbf{x})$

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

For $k = 0, 1, \ldots$

$$
\left\lfloor
\begin{array}{l}
\mathbf{x}_{k+1} = \operatorname{prox}_{\sigma^{-1} \mathsf{f}}\left(\mathbf{s}_k\right) \\
\mathbf{s}_{k+1} = \mathbf{s}_k - \mathbf{x}_{k+1} - \operatorname{prox}_{\sigma^{-1} \mathsf{g}}\left(2\mathbf{x}_{k+1} - \mathbf{s}_k\right)
\end{array}
\right.
$$

Application: ℓ_1 in a redundant dictionary + constraints

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \frac{1}{2}\|\mathbf{\Phi x} - \mathbf{z}\|^2 + \eta\|\mathbf{\Psi x}\|_1 + \iota_{[\mathrm{x}_{\min},\mathrm{x}_{\max}]^N}(\mathbf{x})$
where $\mathbf{\Phi} \in \mathbb{R}^{M\times N}$, $\mathbf{z} \in \mathbb{R}^M$, $\eta > 0$, $\mathbf{\Psi} \in \mathbb{R}^{L\times N}$, $(\mathrm{x}_{\min},\mathrm{x}_{\max}) \in \mathbb{R}^2$.

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

For $k = 0, 1, \dots$

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

Application: ℓ_1-ball regularisation

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \frac{1}{2}\|\boldsymbol{\Phi}\mathbf{x}-\mathbf{z}\|^2 + \iota_{\mathcal{B}_1(\mathbf{0},\eta)}(\boldsymbol{\Psi}\mathbf{x}) + \iota_{[\mathrm{x}_{\min},\mathrm{x}_{\max}]^N}(\mathbf{x})$
where $\boldsymbol{\Phi}\in\mathbb{R}^{M\times N}$, $\mathbf{z}\in\mathbb{R}^M$, $\eta>0$, $\boldsymbol{\Psi}\in\mathbb{R}^{L\times N}$, $(\mathrm{x}_{\min},\mathrm{x}_{\max})\in\mathbb{R}^2$, 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 $\frac{1}{\tau}-\sigma\|\boldsymbol{\Psi}\|^2 > \frac{\|\boldsymbol{\Phi}\|^2}{2}$.
For $k=0,1,\ldots$

$$
\left\lfloor
\begin{array}{l}
\mathbf{x}_{k+1} = \Pi_{[\mathrm{x}_{\min},\mathrm{x}_{\max}]^N}\Big(\mathbf{x}_k - \tau\big(\boldsymbol{\Phi}^*(\boldsymbol{\Phi}\mathbf{x}_k-\mathbf{z}) + \boldsymbol{\Psi}^*\mathbf{u}_k\big)\Big)\\
\widetilde{\mathbf{u}}_k = \mathbf{u}_k + \sigma\boldsymbol{\Psi}(2\mathbf{x}_{k+1}-\mathbf{x}_k)\\
\mathbf{u}_{k+1} = \widetilde{\mathbf{u}}_k - \sigma\Pi_{\mathcal{B}_1(\mathbf{0},\eta)}\Big(\sigma^{-1}\widetilde{\mathbf{u}}_k\Big)
\end{array}
\right.
$$

Example: 3D mesh denoising

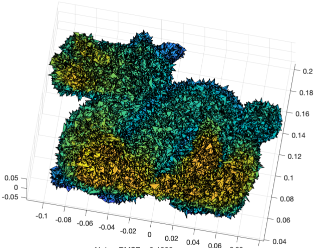
INVERSE PROBLEM: $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$

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

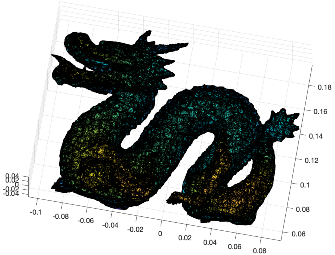
MINIMISATION PROBLEM: Find $\widehat{\mathbf{x}} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} F(\mathbf{x}) + R(\mathbf{x})$

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

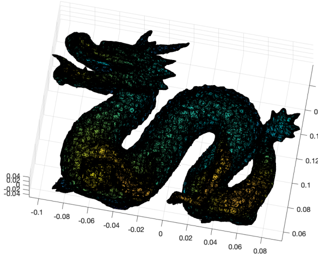
Example: 3D mesh denoising



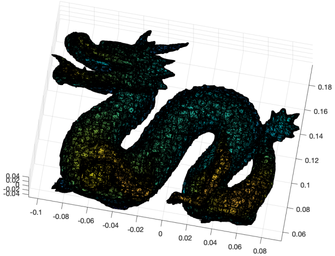
Noisy, RMSE = 3.4388



Primal dual, RMSE = 0.48048

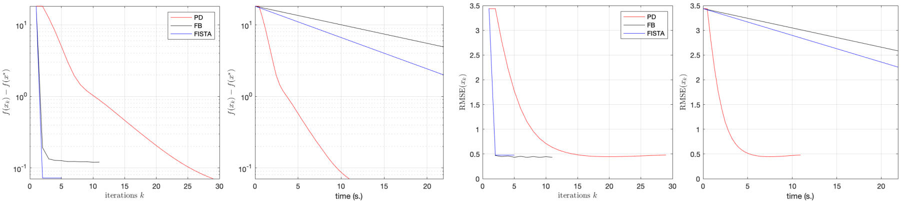


forward-backward, RMSE = 0.43502



FISTA, RMSE = 0.47765

Example: 3D mesh denoising



Generalisation of the Condat-Vu algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^\star \in \underset{\mathbf{x}}{\mathrm{Argmin}}\ \mathsf{h}(\mathbf{x}) + \mathsf{f}(\mathbf{x}) + \sum_{i=1}^{I} \mathsf{g}_i(\mathbf{L}_i\mathbf{x})$

DUAL PROBLEM: Find $(\mathbf{u}_i^\star)_{1\leqslant i\leqslant I} \in \underset{(\mathbf{u}_1,\ldots,\mathbf{u}_I)}{\mathrm{Argmin}}\ (\mathsf{h}+\mathsf{f})^*(-\sum_{i=1}^{I}\mathbf{L}_i^*\mathbf{u}_i) + \sum_{i=1}^{I}\mathsf{g}_i^*(\mathbf{u}_i)$

where $\mathsf{h}\colon \mathbb{R}^N \to \mathbb{R}$ is convex and β-Lipschitz differentiable, $\mathsf{f} \in \Gamma_0(\mathbb{R}^N)$, for every $i \in \{1,\ldots,I\}$, $\mathsf{g}_i \in \Gamma_0(\mathbb{R}^{M_i})$ 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 $k = 0, 1, \ldots$

$$\mathbf{x}_{k+1} = \mathrm{prox}_{\tau\mathsf{f}}\Big(\mathbf{x}_k - \tau\big(\nabla\mathsf{h}(\mathbf{x}_k) + \textstyle\sum_{i=1}^{I}\mathbf{L}_i^*\mathbf{u}_{i,k}\big)\Big)$$

for $i = 1,\ldots,I$

$$\mathbf{u}_{i,k+1} = \mathrm{prox}_{\sigma\mathsf{g}_i^*}\big(\mathbf{u}_{i,k} + \sigma\mathbf{L}_i(2\mathbf{x}_{k+1} - \mathbf{x}_k)\big)$$

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

Generalisation of the Condat-Vu algorithm

PRIMAL PROBLEM: Find $\mathbf{x}^\star \in \underset{\mathbf{x}}{\operatorname{Argmin}}\; \mathrm{h}(\mathbf{x}) + \mathrm{f}(\mathbf{x}) + \sum_{i=1}^{I} \mathrm{g}_i(\mathbf{L}_i\mathbf{x})$

DUAL PROBLEM: Find $(\mathbf{u}_i^\star)_{1\leqslant i\leqslant I} \in \underset{(\mathbf{u}_1,\ldots,\mathbf{u}_I)}{\operatorname{Argmin}}\; (\mathrm{h}+\mathrm{f})^*(-\sum_{i=1}^{I} \mathbf{L}_i^*\mathbf{u}_i) + \sum_{i=1}^{I} \mathrm{g}_i^*(\mathbf{u}_i)$

where $\mathrm{h}\colon \mathbb{R}^N \to \mathbb{R}$ is convex and β-Lipschitz differentiable, $\mathrm{f} \in \Gamma_0(\mathbb{R}^N)$, for every $i \in \{1,\ldots,I\}$, $\mathrm{g}_i \in \Gamma_0(\mathbb{R}^{M_i})$ 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 $k = 0, 1, \ldots$

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

for $i = 1, \ldots, I$

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

If $\mathrm{h} \equiv 0$, 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: $\mathbf{z} = \mathbf{\Phi}\overline{\mathbf{x}} + \mathbf{w}$

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

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

- Data-fidelity: $F(\mathbf{x}) = \iota_{\mathcal{B}_2(\mathbf{z},\varepsilon)}(\mathbf{\Phi x})$

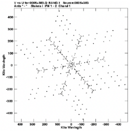
  $\rightsquigarrow$ Splitting in the dual domain (proximity operator)

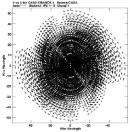
- Regularisation: $R(\mathbf{x}) = \eta \|\mathbf{\Psi x}\|_1 + \iota_{[0,+\infty[^N}(\mathbf{x})$
  with $\eta > 0$, and $\mathbf{\Psi}$ concatenation of first 8 Db wavelets and Dirac basis (average sparsity)

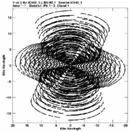
  $\rightsquigarrow$ Splitting to handle all terms separately (with proximity operators)

Example: Radio-interferometric imaging

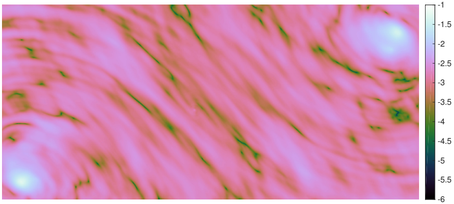
Image courtesy of NRAO/AUI



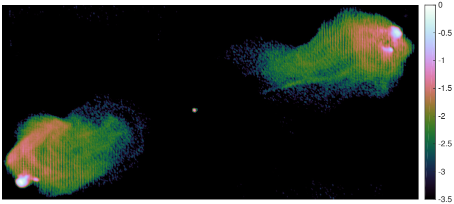




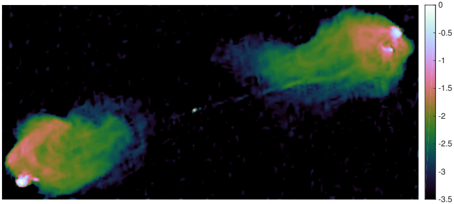
Example: Radio-interferometric imaging



Dirty image $\mathbf{\Phi}^* \mathbf{z}$



Standard method (CLEAN)



Primal-dual