

OPTIMISATION METHODS FOR COMPUTATIONAL IMAGING

Chapter 5 - Optimisation algorithms
Block-coordinate approaches

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

Journées SMAI-MODE 2022 – Limoges

Introduction

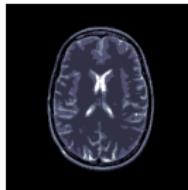
Motivation for BC approaches: non convex problems

BLIND DECONVOLUTION:

- **Non-linear inverse problem:** $\mathbf{z} = \overline{\Phi}\overline{\mathbf{x}} + \mathbf{w}$
 - ~ $\overline{\Phi}$ and $\overline{\mathbf{x}}$ unknown
- **Non-convex minimisation problem:** $\underset{\mathbf{x}, \Phi}{\text{minimise}} \frac{1}{2} \|\Phi\mathbf{x} - \mathbf{z}\|^2 + R_1(\mathbf{x}) + R_2(\Phi)$
 - ~ R_1 regularisation for \mathbf{x}
 - ~ R_2 regularisation for Φ
 - ~Even if R_1 and R_2 are convex, the full objective function is not
- **Approach:** Use *alternating minimisation* algorithms
 - ~See \mathbf{x} and Φ as “blocks”

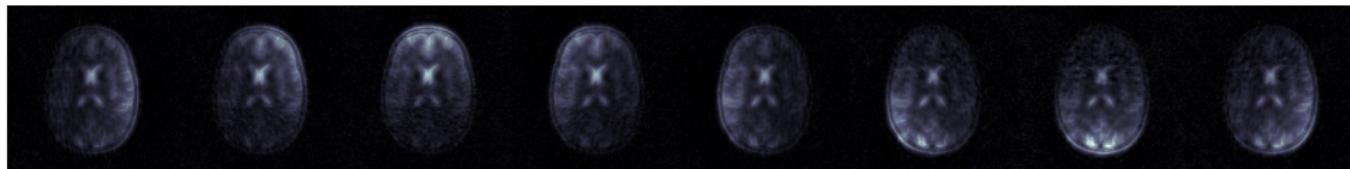
Motivation for BC approaches: non convex problems

BLIND DECONVOLUTION: Example in parallel MRI



acquisition ↓

↑ reconstruction?

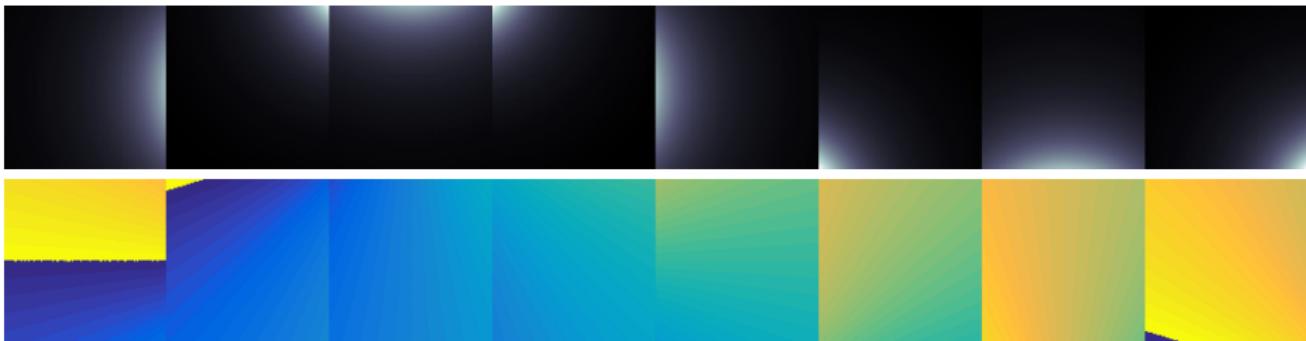


Motivation for BC approaches: non convex problems

BLIND DECONVOLUTION: Example in parallel MRI

$$(\forall c \in \{1, \dots, C\}) \quad \mathbf{z}_c = \mathbf{M}\mathbf{F}(\bar{\mathbf{s}}_c \cdot \bar{\mathbf{x}})$$

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown original image.
- $\bar{\mathbf{s}}_c \in \mathbb{C}^N \rightsquigarrow$ unknown sensitivity map related to coil $c \in \{1, \dots, C\}$.

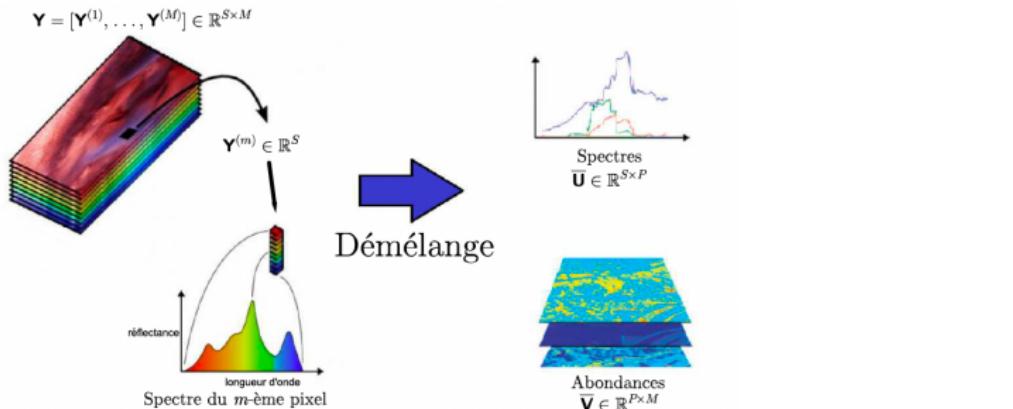


Example of simulated sensitivity maps (magnitude and phase)

Motivation for BC approaches: non convex problems

OTHER NON-CONVEX PROBLEMS:

- Dictionary learning: minimise $H(\mathbf{x}) + R_{\mathbf{x}, \Psi}(\mathbf{x})$
 - ~~~ \mathbf{x} unknown image
 - ~~~ Ψ unknown sparsifying operator
- Non-negative Matrix Factorisation (NMF): minimise $H_{\mathbf{U}, \mathbf{V}}(\mathbf{UV}) + R_1(\mathbf{U}) + R_2(\mathbf{V})$



Motivation for block-coordinate approaches

LARGE DIMENSIONS: **High dimensional images**

- **Inverse problem:** $\mathbf{z} = \Phi \bar{\mathbf{x}} + \mathbf{w}$

- ~~ $\bar{\mathbf{x}} \in \mathbb{R}^N$ unknown with N very large ($\Phi \in \mathbb{R}^{M \times N}$ fat matrix)

- ~~ Split $\bar{\mathbf{x}} = (\bar{\mathbf{x}}^{(b)})_{1 \leq b \leq B}$ into blocks

- **Minimisation problem:** minimise $\frac{1}{2} \|\Phi \mathbf{x} - \mathbf{z}\|^2 + R(\mathbf{x})$

- ~~ $R(\mathbf{x}) = \sum_{1 \leq b \leq B} R_b(\mathbf{x}^{(b)})$

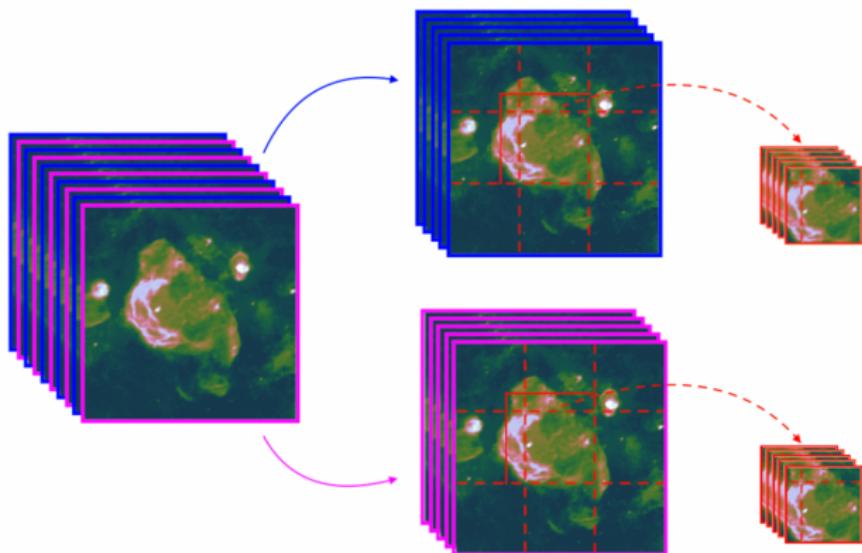
- **Approach:** Use *block splitting* algorithms

- ~~ Only handle a subset of $(\mathbf{x}^{(b)})_{1 \leq b \leq B}$ at each iteration

- ~~ OR parallelise $(\mathbf{x}^{(b)})_{1 \leq b \leq B}$

Motivation for block-coordinate approaches

LARGE DIMENSIONS: **High dimensional images**



Hyperspectral radio-astronomical imaging (typically $N > 10^8$)

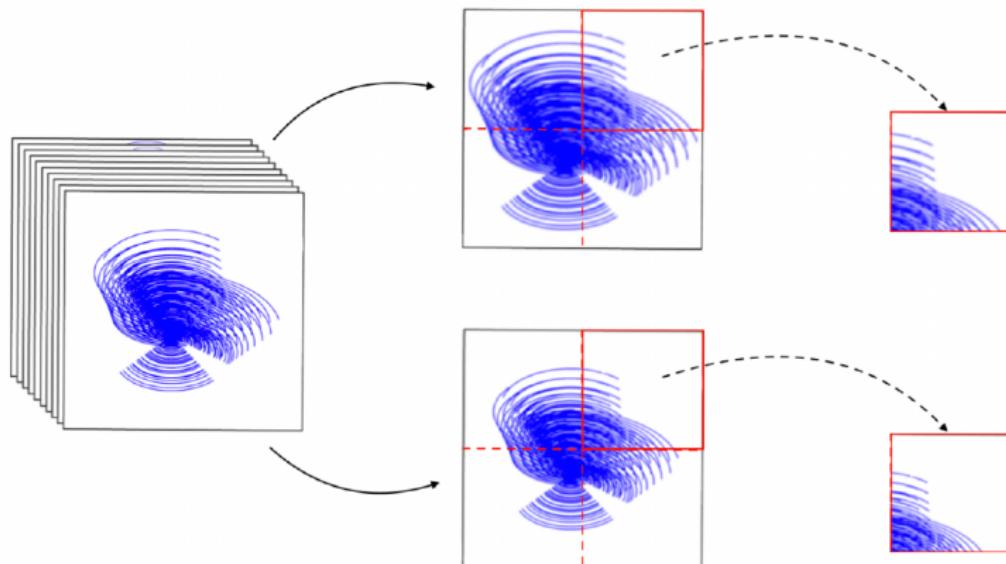
Motivation for block-coordinate approaches

LARGE DIMENSIONS: High dimensional data

- **Non-linear inverse problem:** $\mathbf{z} = \Phi \bar{\mathbf{x}} + \mathbf{w}$
 - ~~~ $\bar{\mathbf{x}} \in \mathbb{R}^N$ unknown
 - ~~~ $\Phi \in \mathbb{R}^{M \times N}$ with M very large (tall matrix)
 - ~~~ Split $\mathbf{z} = (\mathbf{z}^{(d)})_{1 \leq d \leq D}$ and $\Phi = (\Phi_d)_{1 \leq d \leq D}$
- **Minimisation problem:** $\underset{\mathbf{x}, \Phi}{\text{minimise}} \quad \frac{1}{2} \sum_{d=1}^D \|\Phi_d \mathbf{x} - \mathbf{z}^{(d)}\|^2 + R(\mathbf{x})$
- **Approach:** Use *splitting* algorithms
 - ~~~ Only handle a subset of $(\mathbf{z}^{(d)})_{1 \leq d \leq D}$ at each iteration
 - ~~~ OR parallelise $(\mathbf{z}^{(d)})_{1 \leq d \leq D}$

Motivation for block-coordinate approaches

LARGE DIMENSIONS: High dimensional data



Hyperspectral radio-astronomical imaging (Fourier measurements, typically $M > 10^8$)

Motivation for block-coordinate approaches

LARGE DIMENSIONS: High dimensional data

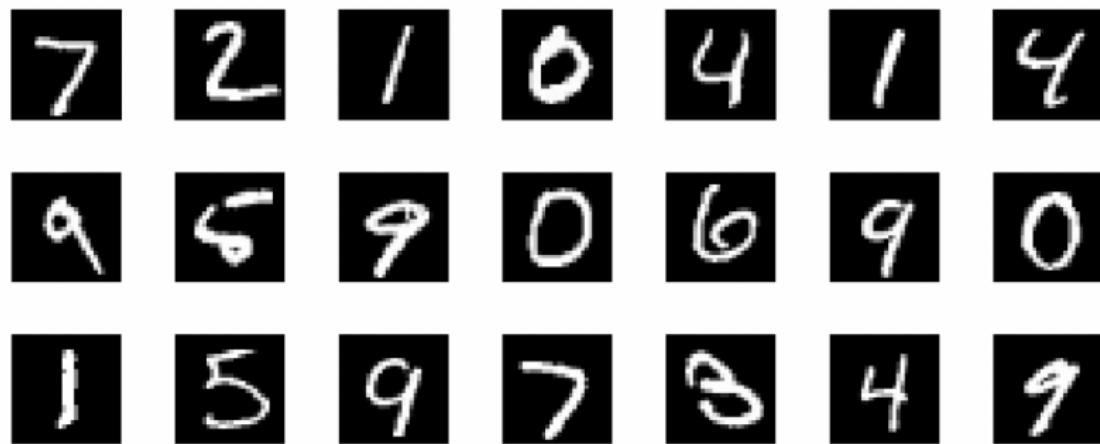
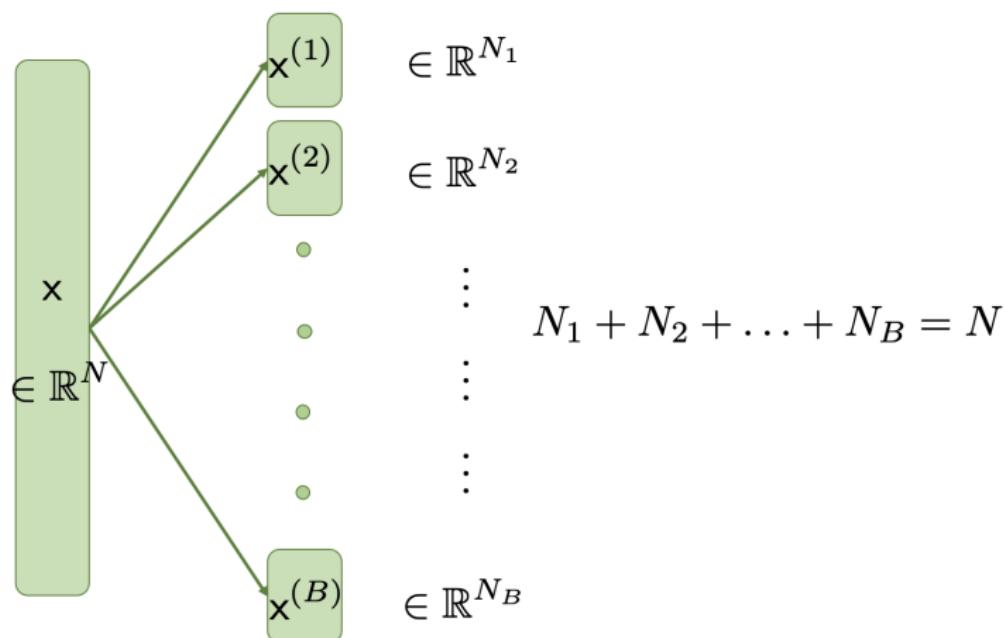


Figure 1: Sample of the MNIST dataset (<http://yann.lecun.com/exdb/mnist/>).

Machine learning (training dataset, typically $M > 10^8$)

Variable splitting



WHY? Split a large object into smaller objects

Operator splitting

$$\mathbf{L} \in \mathbb{R}^{L \times N}$$

$$= \begin{matrix} \mathbf{L}_{1,1} & \mathbf{L}_{1,2} & \cdots & \mathbf{L}_{1,B} \\ \mathbf{L}_{2,1} & \mathbf{L}_{2,2} & \cdots & \mathbf{L}_{2,B} \\ \vdots & & & \vdots \\ \vdots & & & \vdots \\ \mathbf{L}_{D,1} & \mathbf{L}_{D,2} & \cdots & \mathbf{L}_{D,B} \end{matrix} \left. \begin{array}{c} L_1 \\ L_2 \\ \vdots \\ \vdots \\ L_D \end{array} \right\} L$$

$\underbrace{\qquad\qquad\qquad}_{N_1} \underbrace{\qquad\qquad\qquad}_{N_2} \cdots \underbrace{\qquad\qquad\qquad}_{N_B} \qquad\qquad\qquad \underbrace{\qquad\qquad\qquad}_{N}$

WHY? Split a large amount of data

Questions

- Can we **split the object of interest** (e.g. an image) into blocks, and only handle a **subpart of blocks** per iteration?

i.e. $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(B)})$

- ↳ Reduce complexity per iteration
- ↳ Stochastic approaches

Questions

- Can we **split the object of interest** (e.g. an image) into blocks, and only handle a **subpart of blocks** per iteration?

i.e. $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(B)})$

- Reduce complexity per iteration
- Stochastic approaches

- Can we **split the data** (e.g. Fourier measurements) into blocks, and only handle a **subpart of associated functions** per iteration?

i.e. $\mathbf{L} = (\mathbf{L}_{d,b})_{1 \leq d \leq D, 1 \leq b \leq B}$

- Reduce complexity per iteration
- Stochastic approaches

Questions

- Can we **split the object of interest** (e.g. an image) into blocks, and only handle a **subpart of blocks** per iteration?

i.e. $\mathbf{x} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(B)})$

- Reduce complexity per iteration
- Stochastic approaches

- Can we **split the data** (e.g. Fourier measurements) into blocks, and only handle a **subpart of associated functions** per iteration?

i.e. $\mathbf{L} = (\mathbf{L}_{d,b})_{1 \leq d \leq D, 1 \leq b \leq B}$

- Reduce complexity per iteration
- Stochastic approaches

REMARK: Usually stochastic methods are *slow* but necessary to handle huge data (scalability)

Block-Coordinate Forward-Backward Algorithm

Nonconvex minimisation problem

Find $\mathbf{x}^* = ([\mathbf{x}^*]^{(b)})_{1 \leq b \leq B} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})\}$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is β -Lipschitz differentiable
- $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ additively separable

IDEA: At each iteration $k \in \mathbb{N}$, update only a subset of components
~ Gauss-Seidel methods

For $k = 0, 1, \dots$

for $b = 1, \dots, B$

$\mathbf{x}_{k+1}^{(b)} = \operatorname{argmin}_{\mathbf{u}} f(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \dots, \mathbf{x}_k^{(B)})$

Nonconvex minimisation problem

Find $\mathbf{x}^* = ([\mathbf{x}^*]^{(b)})_{1 \leq b \leq B} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})\}$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is β -Lipschitz differentiable
- $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ additively separable

IDEA: At each iteration $k \in \mathbb{N}$, update only a subset of components
~ Gauss-Seidel methods

For $k = 0, 1, \dots$

for $b = 1, \dots, B$

$\mathbf{x}_{k+1}^{(b)} = \operatorname{argmin}_{\mathbf{u}} f(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \dots, \mathbf{x}_k^{(B)})$

- Convergence? Usually necessitates strong conditions (e.g., strong convexity)

Nonconvex minimisation problem

Find $\mathbf{x}^* = ([\mathbf{x}^*]^{(b)})_{1 \leq b \leq B} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})\}$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is β -Lipschitz differentiable
- $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ additively separable

IDEA: At each iteration $k \in \mathbb{N}$, update only a subset of components
~ Gauss-Seidel methods

For $k = 0, 1, \dots$

for $b = 1, \dots, B$

$$\left| \begin{array}{l} \mathbf{x}_{k+1}^{(b)} = \operatorname{argmin}_{\mathbf{u}} f(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \dots, \mathbf{x}_k^{(B)}) + \frac{1}{2\gamma_k} \|\mathbf{u} - \mathbf{x}_k^{(b)}\|^2 \end{array} \right.$$

- Convergence? Usually necessitates strong conditions (e.g., strong convexity)
- Add an ℓ_2 term to obtain convergence...

Nonconvex minimisation problem

Find $\mathbf{x}^* = ([\mathbf{x}^*]^{(b)})_{1 \leq b \leq B} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})\}$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is β -Lipschitz differentiable
- $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ additively separable

IDEA: At each iteration $k \in \mathbb{N}$, update only a subset of components
~ Gauss-Seidel methods

For $k = 0, 1, \dots$

for $b = 1, \dots, B$

$$\left[\left[\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k f_b}(\mathbf{x}_k^{(b)}) \right] \right]$$

- Convergence? Usually necessitates strong conditions (e.g., strong convexity)
- Add an ℓ_2 term to obtain convergence... Leading to alternated proximal method
[Attouch, Bolte, Svaiter, 2011]

Nonconvex minimisation problem

Find $\mathbf{x}^* = ([\mathbf{x}^*]^{(b)})_{1 \leq b \leq B} \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} \{f(\mathbf{x}) := h(\mathbf{x}) + g(\mathbf{x})\}$.

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is β -Lipschitz differentiable
- $g: \mathbb{R}^N \rightarrow]-\infty, +\infty]$ additively separable

IDEA: At each iteration $k \in \mathbb{N}$, update only a subset of components
~ Gauss-Seidel methods

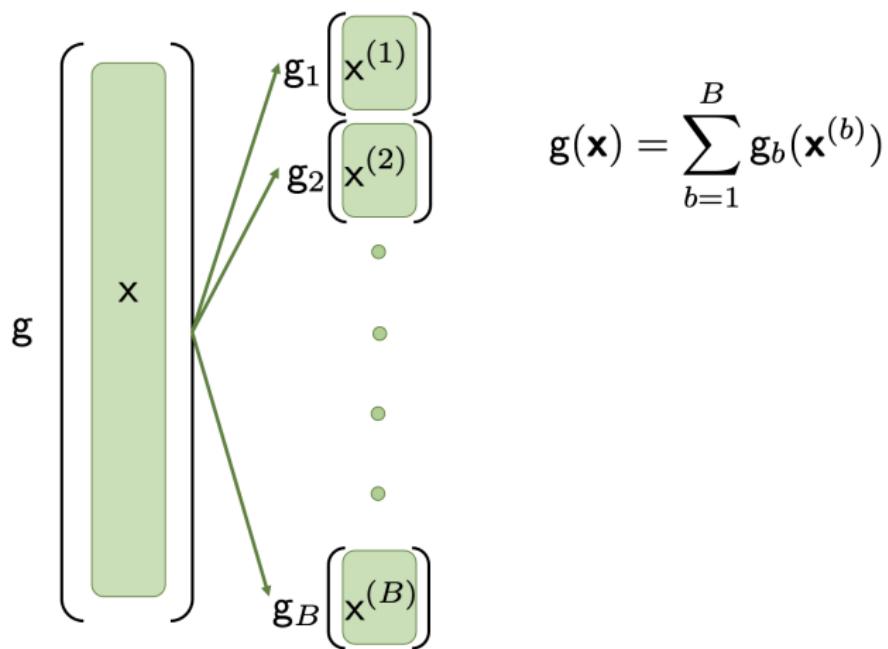
For $k = 0, 1, \dots$

for $b = 1, \dots, B$

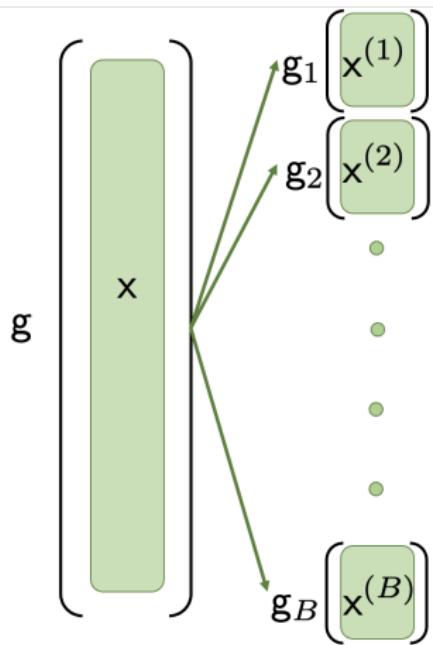
$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k f_b}(\mathbf{x}_k^{(b)})$

- Convergence? Usually necessitates strong conditions (e.g., strong convexity)
- Add an ℓ_2 term to obtain convergence... Leading to alternated proximal method
- **IDEA:** Use alternated FB iterations to fully exploit the splitting structure of f

Additively separable functions: g



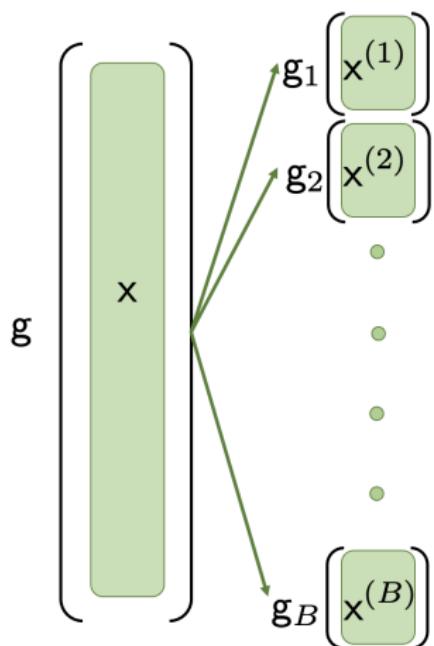
Additively separable functions: g



$$g(\mathbf{x}) = \sum_{b=1}^B g_b(\mathbf{x}^{(b)})$$

$(\forall b \in \{1, \dots, B\})$
 $g_b: \mathbb{R}^{N_b} \rightarrow]-\infty, +\infty]$ l.s.c.

Additively separable functions: g



$$g(\mathbf{x}) = \sum_{b=1}^B g_b(\mathbf{x}^{(b)})$$

$(\forall b \in \{1, \dots, B\})$
 $g_b: \mathbb{R}^{N_b} \rightarrow]-\infty, +\infty]$ l.s.c.

$$\text{prox}_g(\mathbf{x}) = \left(\text{prox}_{g_b}(\mathbf{x}^{(b)}) \right)_{1 \leq b \leq B}$$

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \text{dom } g$

For $k = 0, 1, \dots$

Let $b_k \in \{1, \dots, B\}$ and $\gamma_k \in]0, \beta_{b_k}^{-1}[$.

$$\mathbf{x}_{k+1}^{(b_k)} \in \text{prox}_{\gamma_k^{-1} g_{b_k}} \left(\mathbf{x}_k^{(b_k)} - \gamma_k \nabla_{b_k} h(\mathbf{x}_k) \right)$$

for $b \in \{1, \dots, B\} \setminus \{b_k\}$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

- $(\forall k \in \mathbb{N})$ update only one block $\mathbf{x}^{(b_k)}$
- $(\forall b \in \{1, \dots, B\})$ $\nabla_b h$ partial gradient of h , and β_b its Lipschitz constant
- If $g \in \Gamma_0(\mathbb{R}^N)$, $\gamma_k \in]0, 2\beta_{b_k}^{-1}[$

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \text{dom } g$

For $k = 0, 1, \dots$

Let $b_k \in \{1, \dots, B\}$, $\gamma_k \in]0, 1[$ and $\mathbf{A}_{b_k} \in \mathbb{R}^{N_{b_k} \times N_{b_k}}$.

$$\mathbf{x}_{k+1}^{(b_k)} \in \text{prox}_{\gamma_k^{-1} \mathbf{g}^{b_k}} \left(\mathbf{x}_k^{(b_k)} - \gamma_k \mathbf{A}_{b_k}^{-1} \nabla_{b_k} h(\mathbf{x}_k) \right)$$

for $b \in \{1, \dots, B\} \setminus \{b_k\}$

$$\left. \begin{aligned} \mathbf{x}_{k+1}^{(b)} &= \mathbf{x}_k^{(b)} \end{aligned} \right]$$

- $(\forall k \in \mathbb{N})$ update only one block $\mathbf{x}^{(b_k)}$
- $(\forall b \in \{1, \dots, B\})$ $\nabla_b h$ partial gradient of h , and β_b its Lipschitz constant
- If $g \in \Gamma_0(\mathbb{R}^N)$, $\gamma_k \in]0, 2\beta_{b_k}^{-1}[$
- Can be paired with variable metric strategy for acceleration

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \text{dom } g$

For $k = 0, 1, \dots$

Let $b_k \in \{1, \dots, B\}$, $\gamma_k \in]0, 1[$ and $\mathbf{A}_{b_k} \in \mathbb{R}^{N_{b_k} \times N_{b_k}}$.

$$\mathbf{x}_{k+1}^{(b_k)} \in \text{prox}_{\gamma_k^{-1} g^{b_k}} \left(\mathbf{x}_k^{(b_k)} - \gamma_k \mathbf{A}_{b_k}^{-1} \nabla_{b_k} h(\mathbf{x}_k) \right)$$

for $b \in \{1, \dots, B\} \setminus \{b_k\}$

$$\left. \mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)} \right.$$

- $(\forall k \in \mathbb{N})$ update only one block $\mathbf{x}^{(b_k)}$
- $(\forall b \in \{1, \dots, B\})$ $\nabla_b h$ partial gradient of h , and β_b its Lipschitz constant
- If $g \in \Gamma_0(\mathbb{R}^N)$, $\gamma_k \in]0, 2\beta_{b_k}^{-1}[$
- Can be paired with variable metric strategy for acceleration
- See [Bolte, Sabbach, Teboulle, 2013], [Frankel, Garrigos, Peypouquet, 2014], and [Chouzenoux, Pesquet, Repetti, 2015]

BCFB algorithm: cyclic/quasi-cyclic updating rules

- * Cyclic rule: update sequentially blocks $1, 2, \dots, B$.
- * Quasi-cyclic rule: there exists $K \geq B$ such that, for every $k \in \mathbb{N}$,
 $\{1, \dots, B\} \subset \{b_k, \dots, b_{k+K-1}\}$.

BCFB algorithm: cyclic/quasi-cyclic updating rules

- * Cyclic rule: update sequentially blocks $1, 2, \dots, B$.
- * Quasi-cyclic rule: there exists $K \geq B$ such that, for every $k \in \mathbb{N}$,
 $\{1, \dots, B\} \subset \{b_k, \dots, b_{k+K-1}\}$.

Example: $B = 3$ blocks denoted $\{1, 2, 3\}$

- $K = 3$:
 - cyclic updating order: $\{1, 2, 3, 1, 2, 3, \dots\}$
 - example of quasi-cyclic updating order: $\{1, 3, 2, 2, 1, 3, \dots\}$

BCFB algorithm: cyclic/quasi-cyclic updating rules

- * Cyclic rule: update sequentially blocks $1, 2, \dots, B$.
- * Quasi-cyclic rule: there exists $K \geq B$ such that, for every $k \in \mathbb{N}$,
 $\{1, \dots, B\} \subset \{b_k, \dots, b_{k+K-1}\}$.

Example: $B = 3$ blocks denoted $\{1, 2, 3\}$

- $K = 3$:
 - cyclic updating order: $\{1, 2, 3, 1, 2, 3, \dots\}$
 - example of quasi-cyclic updating order: $\{1, 3, 2, 2, 1, 3, \dots\}$
- $K = 4$: possibility to update some blocks more than once every K iteration
 - $\{1, 3, 2, 2, 2, 1, 3, \dots\}$

BC-FB algorithm: Conditions for convergence

ASSUMPTIONS:

- $(\forall b_k \in \{1, \dots, B\}) \mathbf{A}_{b_k} \in \mathbb{R}^{N_{b_k} \times N_{b_k}}$ SPD satisfies the **MM assumption**
at $x_k^{(b_k)}$ for the restriction of \mathbf{h} to the block b_k :
$$\mathbf{y} \in \mathbb{R}^{N_{b_k}} \mapsto \mathbf{h} \left(x_k^{(1)}, \dots, x_k^{(b_k-1)}, \mathbf{y}, x_k^{(b_k+1)}, \dots, x_k^{(J)} \right).$$
- $(\exists (\underline{\nu}, \bar{\nu}) \in]0, +\infty[^2) (\forall k \in \mathbb{N}) (\forall \mathbf{u} \in \mathbb{R}^{N_{b_k}}) \underline{\nu} \|\mathbf{u}\|^2 \leq \mathbf{u}^\top \mathbf{A}_{b_k} \mathbf{u} \leq \bar{\nu} \|\mathbf{u}\|^2$
- Blocks $(b_k)_{k \in \mathbb{N}}$ updated according to a **quasi-cyclic rule**
- f satisfies the **KL inequality**
- The **step-size** is chosen such that:
 - if g is **convex**: $(\forall k \in \mathbb{N}) \gamma_k \in]0, 2[$
 - if g is **non-convex**: $(\forall k \in \mathbb{N}) \gamma_k \in]0, 1[$

BC-FB algorithm: Conditions for convergence

ASSUMPTIONS:

- $(\forall b_k \in \{1, \dots, B\})$ \mathbf{A}_{b_k} SPD satisfies the **MM assumption** at $x_k^{(b_k)}$
- $(\exists (\underline{\nu}, \bar{\nu}) \in]0, +\infty[^2) (\forall k \in \mathbb{N}) (\forall \mathbf{u} \in \mathbb{R}^{N_{b_k}}) \underline{\nu} \|\mathbf{u}\|^2 \leq \mathbf{u}^\top \mathbf{A}_{b_k} \mathbf{u} \leq \bar{\nu} \|\mathbf{u}\|^2$
- Blocks $(b_k)_{k \in \mathbb{N}}$ updated according to a **quasi-cyclic rule**
- f satisfies the **KL inequality**
- The **step-size** is chosen such that:
 - if g is **convex**: $(\forall k \in \mathbb{N}) \gamma_k \in]0, 2[$
 - if g is **non-convex**: $(\forall k \in \mathbb{N}) \gamma_k \in]0, 1[$

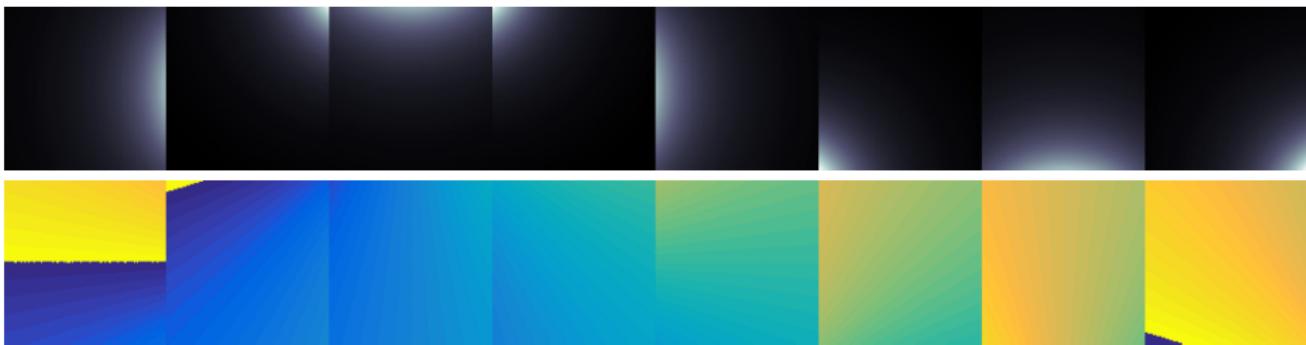
- $(x_k)_{k \in \mathbb{N}}$ converges to a critical point x^* of f .
- $(f(x_k))_{k \in \mathbb{N}}$ is a nonincreasing sequence converging to $f(x^*)$.

Example: Calibration and Imaging in MR

NON-LINEAR INVERSE PROBLEM: $(\forall c \in \{1, \dots, C\})$

$$\mathbf{z}_c = \mathbf{MF}(\bar{\mathbf{s}}_c \cdot \bar{\mathbf{x}}) + \mathbf{w}_c$$

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown original image
- $\bar{\mathbf{s}}_c \in \mathbb{C}^N \rightsquigarrow$ unknown sensitivity map related to coil $c \in \{1, \dots, C\}$



Example of simulated sensitivity maps (magnitude and phase)

Example: Calibration and Imaging in MR

NON-LINEAR INVERSE PROBLEM: $(\forall c \in \{1, \dots, C\})$

$$\mathbf{z}_c = \mathbf{MF}(\bar{\mathbf{s}}_c \cdot \bar{\mathbf{x}}) + \mathbf{w}_c$$

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown original image
- $\bar{\mathbf{s}}_c \in \mathbb{C}^N \rightsquigarrow$ unknown sensitivity map related to coil $c \in \{1, \dots, C\}$

NON-CONVEX MINIMISATION PROBLEM:

$$\underset{\mathbf{x}, \mathbf{s}_1, \dots, \mathbf{s}_C}{\text{minimise}} \quad \frac{1}{2} \sum_{c=1}^C \|\mathbf{MF}(\mathbf{s}_c \cdot \mathbf{x}) - \mathbf{z}_c\|^2 + R_{\mathbf{x}}(\mathbf{x}) + \sum_{c=1}^C R_{\mathbf{s}_c}(\mathbf{s}_c)$$

- $R_{\mathbf{x}}(\mathbf{x}) = \eta \|\Psi \mathbf{x}\|_1$ with Ψ db8 wavelet transform
- $R_{\mathbf{s}_c} = \mu_c \|\mathbf{s}_c\|_{TV}$
- Alternate between estimating $\bar{\mathbf{x}}$ and $(\bar{\mathbf{s}}_c)_{1 \leqslant c \leqslant C}$

Example: Calibration and Imaging in MR

NON-LINEAR INVERSE PROBLEM: $(\forall c \in \{1, \dots, C\})$

$$\mathbf{z}_c = \mathbf{MF}(\bar{\mathbf{s}}_c \cdot \bar{\mathbf{x}}) + \mathbf{w}_c$$

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown original image
- $\bar{\mathbf{s}}_c \in \mathbb{C}^N \rightsquigarrow$ unknown sensitivity map related to coil $c \in \{1, \dots, C\}$

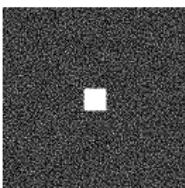
NON-CONVEX MINIMISATION PROBLEM:

$$\underset{\mathbf{x}, \mathbf{s}_1, \dots, \mathbf{s}_C}{\text{minimise}} \quad \frac{1}{2} \sum_{c=1}^C \|\mathbf{MF}(\mathbf{s}_c \cdot \mathbf{x}) - \mathbf{z}_c\|^2 + R_{\mathbf{x}}(\mathbf{x}) + \sum_{c=1}^C R_{\mathbf{s}_c}(\mathbf{s}_c)$$

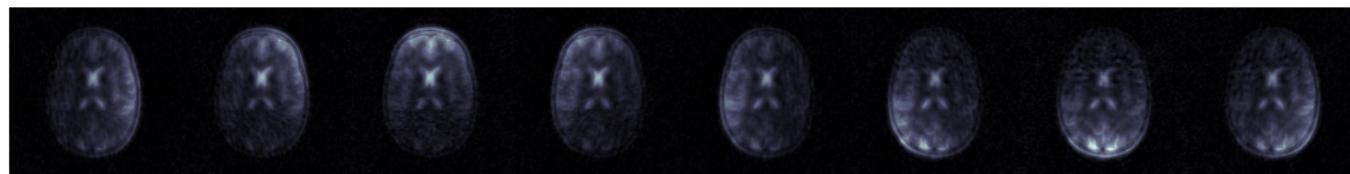
- $R_{\mathbf{x}}(\mathbf{x}) = \eta \|\Psi \mathbf{x}\|_1$ with Ψ db8 wavelet transform
- \mathbf{s}_c are smooth spatial functions, i.e., band limited
 - \rightsquigarrow Estimate (only) the non-zero central Fourier coefficients $\bar{\mathbf{d}}_c \in \mathbb{C}^S$ with $S \ll N$ (reduce degrees of freedom)
- Alternate between estimating $\bar{\mathbf{x}}$ and $(\bar{\mathbf{d}}_c)_{1 \leq c \leq C}$

Example: Calibration and Imaging in MR

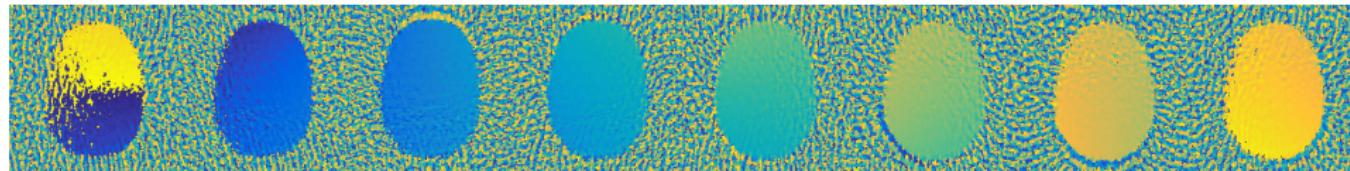
SIMULATED DATA: $N = 200 \times 200$, $C = 8$, $M = 8 \times 10683$



Example of mask with acceleration factor 4 ($8N/M = 3.74$)



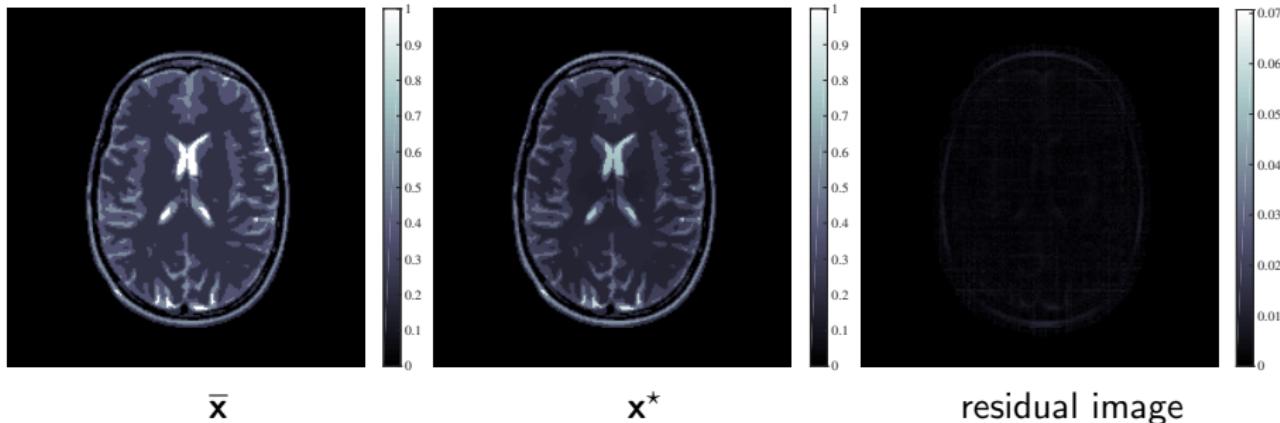
Magnitude physical coil images



Phase physical coil images

Example: Calibration and Imaging in MR

SIMULATED DATA: $N = 200 \times 200$, $C = 8$, $M = 8 \times 10683$



Example: Calibration and Imaging in radio-astronomy

NON-LINEAR INVERSE PROBLEM:

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

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown sky intensity image
 - $\bar{\mathbf{G}}_c \in \mathbb{C}^{M \times N} \rightsquigarrow$ contains the unknown antenna-based gains centred at the frequencies measured by the antenna pairs
 - \rightsquigarrow each row contains the convolution between the antenna-based gain **compact** Fourier kernels $\bar{\mathbf{u}}_\alpha$ and $\bar{\mathbf{u}}_\beta$ associated with antennas (α, β)
- [Repetti *et al.*, 2017]

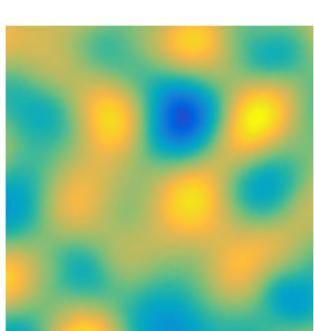
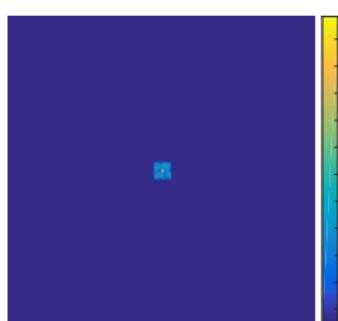
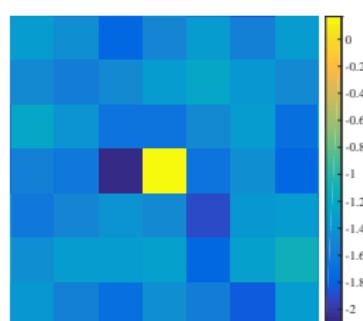


Image space



Fourier space



Fourier kernels

Example: Calibration and Imaging in radio-astronomy

NON-LINEAR INVERSE PROBLEM:

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

- $\bar{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown sky intensity image
- $\bar{\mathbf{G}}_c \in \mathbb{C}^{M \times N} \rightsquigarrow$ contains the unknown antenna-based gains centred at the frequencies measured by the antenna pairs
 - \rightsquigarrow each row contains the convolution between the antenna-based gain **compact** Fourier kernels $\bar{\mathbf{u}}_\alpha$ and $\bar{\mathbf{u}}_\beta$ associated with antennas (α, β)

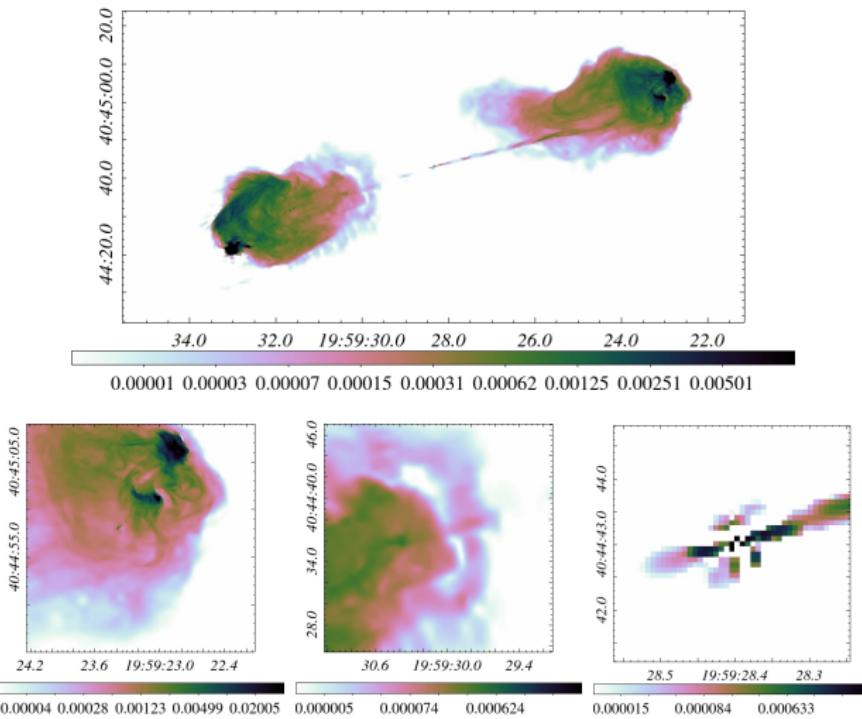
NON-CONVEX MINIMISATION PROBLEM:

$$\underset{\mathbf{x}, \mathbf{d}_1, \dots, \mathbf{u}_{n_a}}{\text{minimise}} \quad \frac{1}{2} \|\mathbf{G}\mathbf{F}(\mathbf{x}) - \mathbf{z}_c\|^2 + R_{\mathbf{x}}(\mathbf{x}) + \sum_{\alpha=1}^{n_a} R_\alpha(\mathbf{u}_\alpha)$$

- $R_{\mathbf{x}}(\mathbf{x}) = \eta \|\Psi \mathbf{x}\|_1$ with Ψ concatenation of first 8 Db wavelet transforms and Dirac basis
- R_α constraints on the gains
- Alternate between estimating $\bar{\mathbf{x}}$ and $(\bar{\mathbf{u}}_\alpha)_{1 \leqslant \alpha \leqslant n_a}$

Example: Calibration and Imaging in radio-astronomy

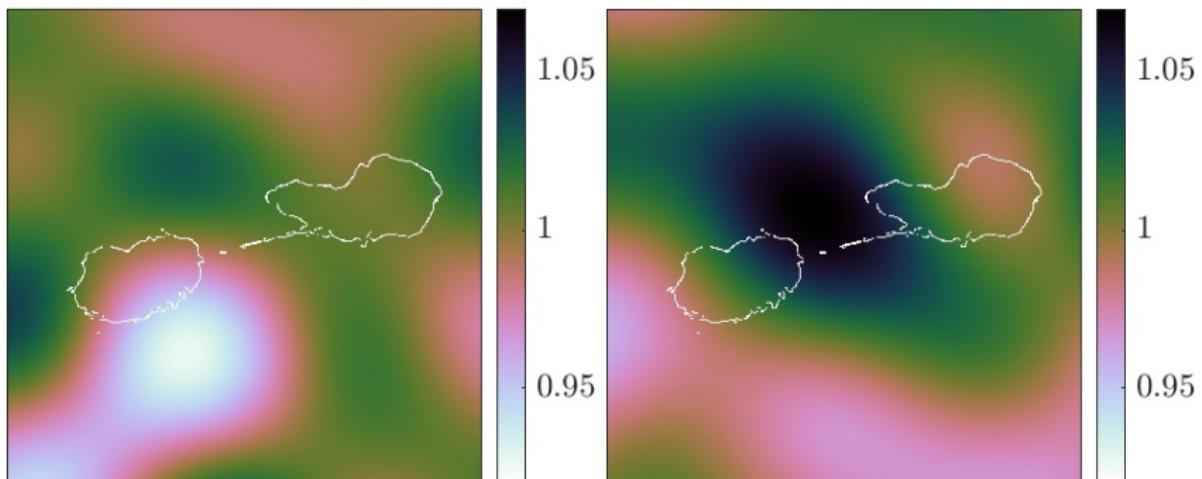
REAL DATA: Cygnus A - image estimate (source: Dabbech et al, 2021)



Example: Calibration and Imaging in radio-astronomy

REAL DATA: Cygnus A - gain estimates (image domain)

Config. C, $S = 7 \times 7$



Stochastic Block-Coordinate
Forward-Backward Algorithm

Minimisation problem involving a linear operator

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

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

👉 This problem can be solved using FB:

Let $\mathbf{x}_0 \in \operatorname{dom} g$

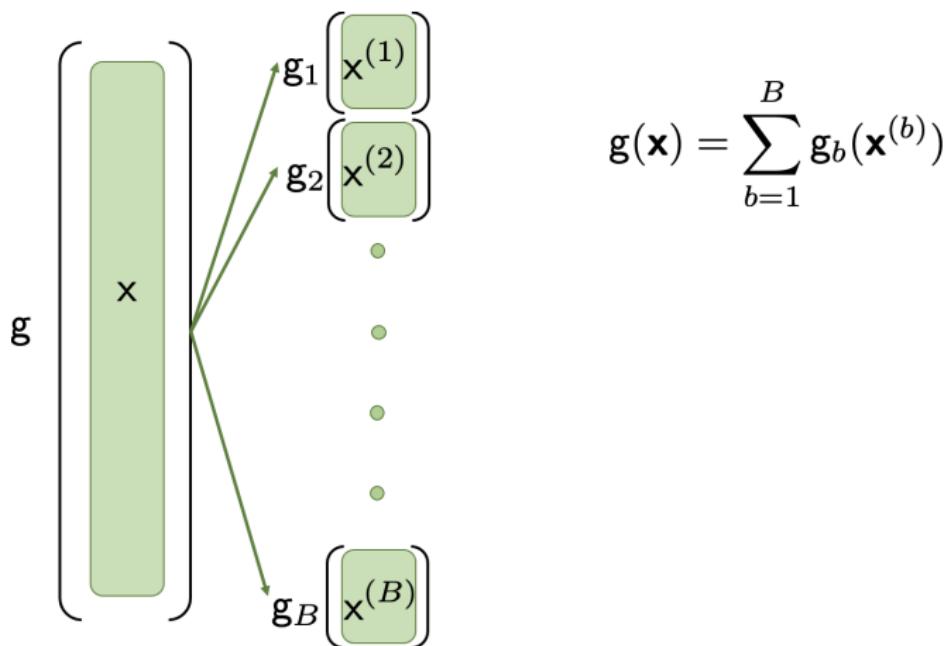
For $k = 0, 1, \dots$

 Let $\gamma_k \in]0, 2/(\beta \|\mathbf{L}\|^2)[$

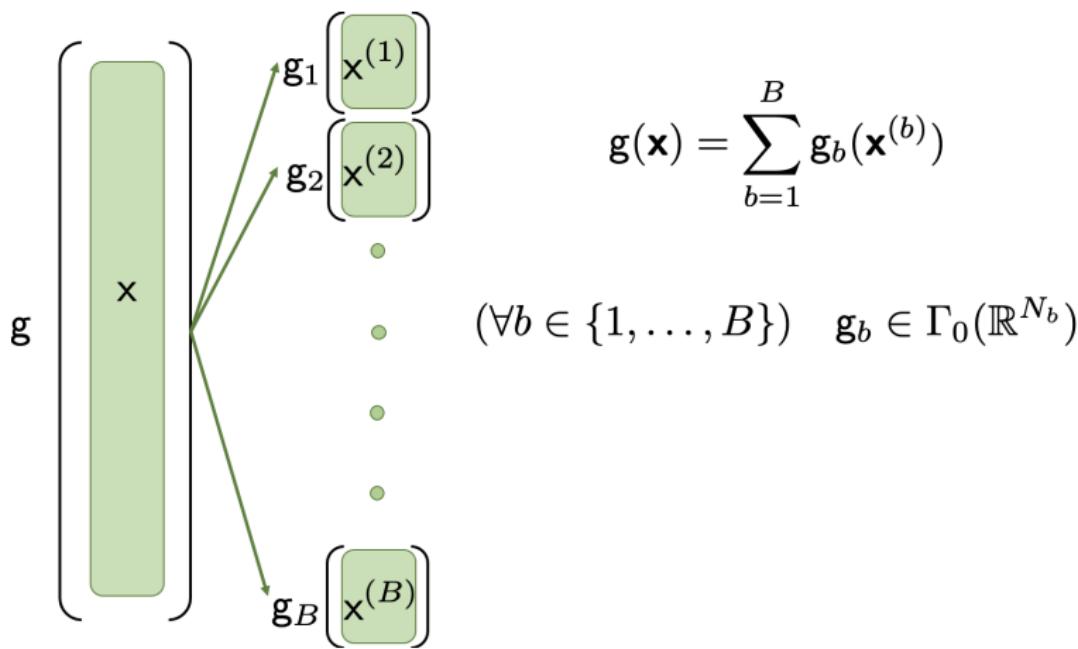
$\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g} \left(\mathbf{x}_k - \gamma_k \mathbf{L}^* \nabla h(\mathbf{L}\mathbf{x}_k) \right)$

ISSUE: Not completely **scalable**

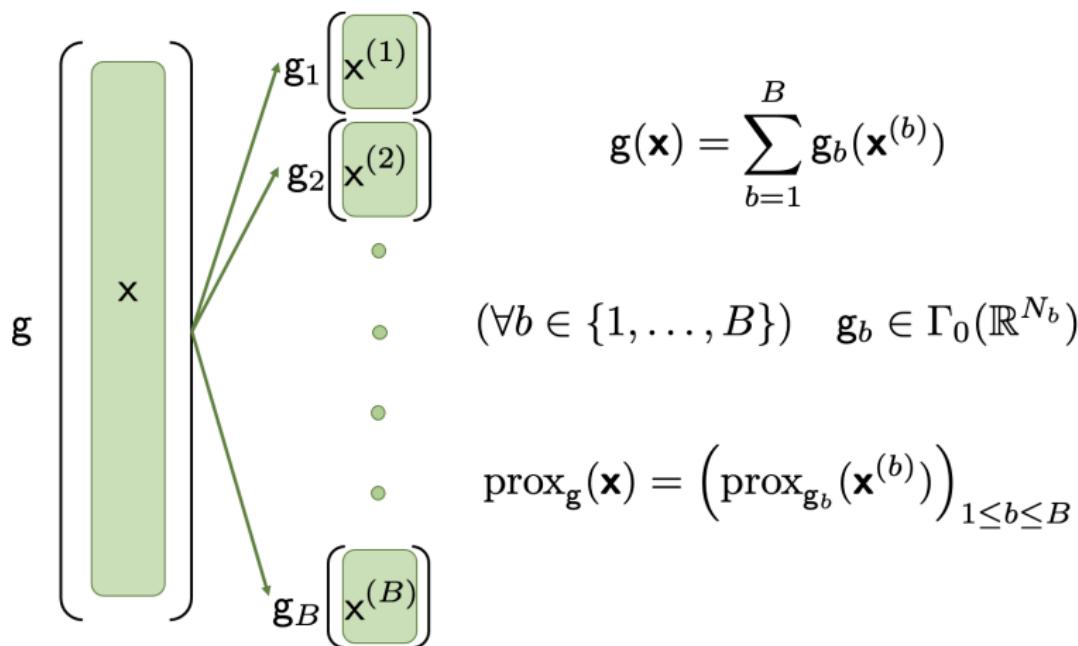
Additively separable functions: g



Additively separable functions: g



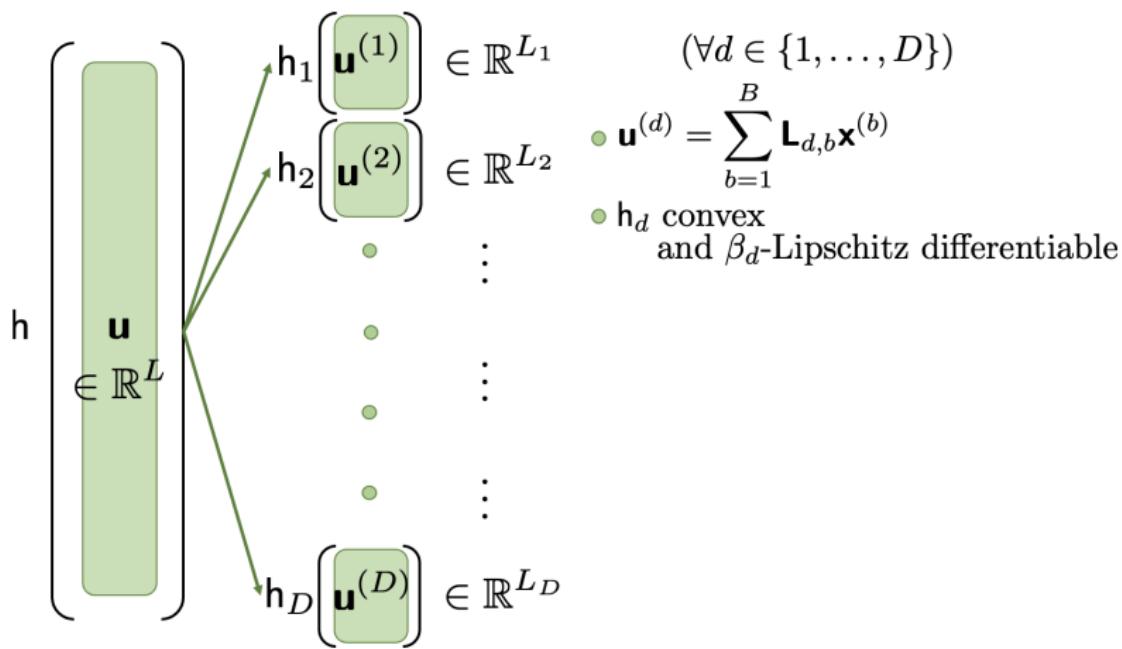
Additively separable functions: g



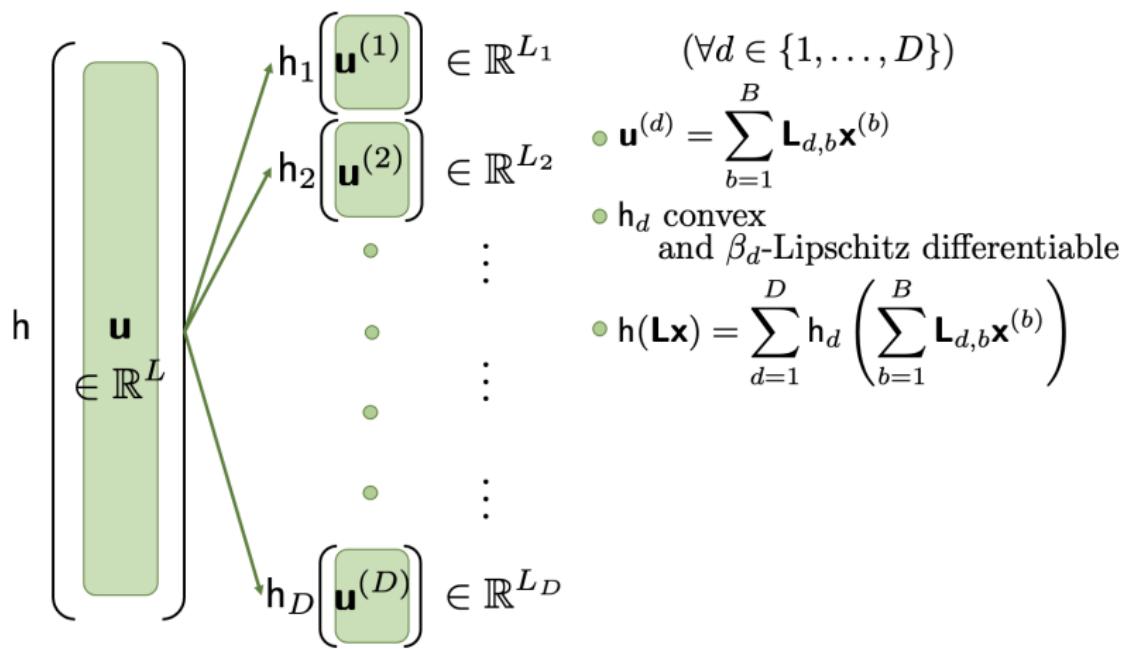
Additively separable functions: h

$$\begin{aligned} h \in \mathbb{R}^L & \quad \left(\begin{array}{c} \text{u} \\ \vdots \\ \text{u} \end{array} \right) \quad \left(\begin{array}{c} h_1 \left(\begin{array}{c} \text{u}^{(1)} \\ \vdots \\ \text{u}^{(D)} \end{array} \right) \in \mathbb{R}^{L_1} \\ h_2 \left(\begin{array}{c} \text{u}^{(2)} \\ \vdots \\ \text{u}^{(D)} \end{array} \right) \in \mathbb{R}^{L_2} \\ \vdots \\ h_D \left(\begin{array}{c} \text{u}^{(D)} \end{array} \right) \in \mathbb{R}^{L_D} \end{array} \right) \quad (\forall d \in \{1, \dots, D\}) \\ & \quad \bullet \quad \text{u}^{(d)} = \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \end{aligned}$$

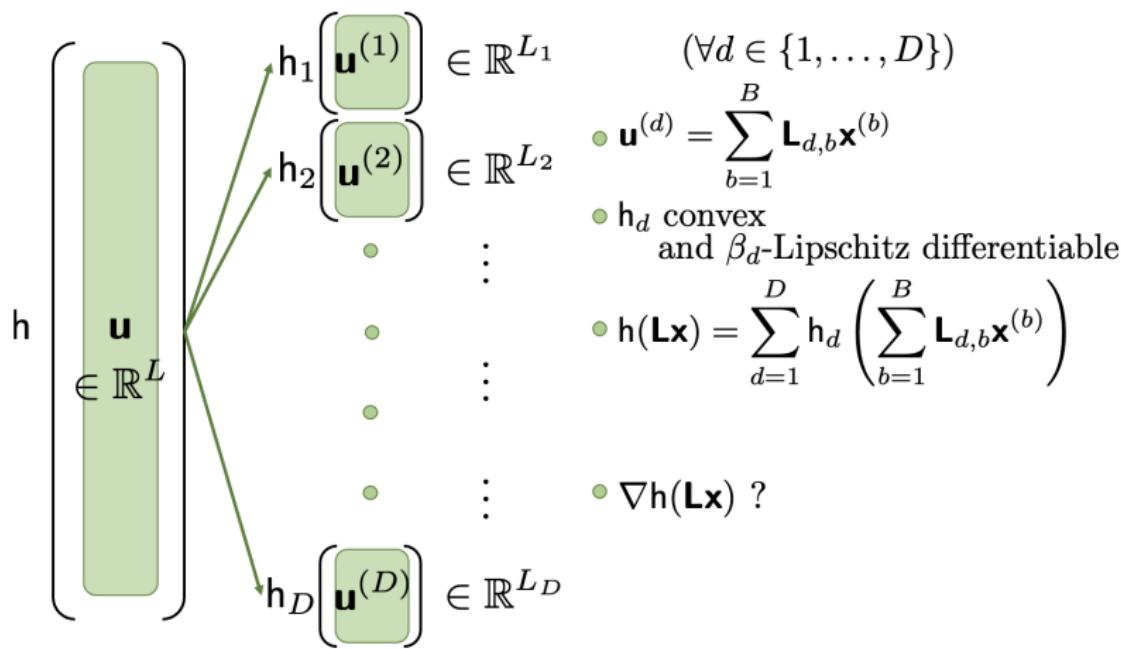
Additively separable functions: h



Additively separable functions: h



Additively separable functions: h



Additively separable functions: h (gradient)

$$\begin{aligned} (\forall \mathbf{x} \in \mathbb{R}^N) \quad h(\mathbf{Lx}) &= \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) \\ &= \sum_{d=1}^D h_d \left([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}] \mathbf{x} \right) \end{aligned}$$

GRADIENT?

$$\begin{aligned} (\forall \mathbf{x} \in \mathbb{R}^N) \quad h(\mathbf{Lx}) &= \sum_{d=1}^D [\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}]^* \nabla h_d ([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}] \mathbf{x}) \\ &= \sum_{d=1}^D \begin{bmatrix} \mathbf{L}_{d,1}^* \\ \vdots \\ \mathbf{L}_{d,B}^* \end{bmatrix} \nabla h_d ([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}] \mathbf{x}) \end{aligned}$$

Block-minimisation problem

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}).$$

where $(\forall d \in \{1, \dots, D\})(\forall b \in \{1, \dots, B\})$

- $h_d: \mathbb{R}^{L_d} \rightarrow \mathbb{R}$ is convex, proper and β_d -Lipschitz differentiable
- $\mathbf{L}_{d,b} \in \mathbb{R}^{L_d \times N_b}$
- $g_b \in \Gamma_0(\mathbb{R}^{N_b})$

OBJECTIVE:

Design a forward-backward algorithm that will only update a subset of the variable \mathbf{x} per iteration, with convergence guarantees.

- Block-coordinate approach

Stochastic BC-FB algorithm in a nutshell

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

 Let $\gamma_k \in]0, 2\tau^{-1}[$

 Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

 For $b \in \mathbb{B}_k$

 Update $\mathbf{x}_{k+1}^{(b)}$ with “sub”-FB step

 For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

 Keep $\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$

Stochastic BC-FB algorithm in a nutshell

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

 Let $\gamma_k \in]0, 2\tau^{-1}[$

 Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

 For $b \in \mathbb{B}_k$

 Update $\mathbf{x}_{k+1}^{(b)}$ with “sub”-FB step

 For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

 Keep $\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$

Stochastic BC-FB algorithm in a nutshell

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

 Let $\gamma_k \in]0, 2\tau^{-1}[$

 Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

 For $b \in \mathbb{B}_k$

 Update $\mathbf{x}_{k+1}^{(b)}$ with “sub”-FB step

 For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

 Keep $\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$

Stochastic BC-FB algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{d=1}^D \mathbf{L}_{d,b}^* \nabla h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}_k^{(b)} \right) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{d=1}^D \mathbf{L}_{d,b}^* \nabla h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}_k^{(b)} \right) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{d=1}^D \mathbf{L}_{d,b}^* \nabla h_d([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}] \mathbf{x}_k) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{d=1}^D h_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

$$\quad \text{Let } \gamma_k \in]0, 2\tau^{-1}[$$

 Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

 For $b \in \mathbb{B}_k$

$$\quad \quad \quad \mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{d=1}^D \mathbf{L}_{d,b}^* \nabla h_d([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}] \mathbf{x}_k) \right)$$

 For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\quad \quad \quad \mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm: Conditions for convergence

ASSUMPTIONS:

Let $b \in \{1, \dots, B\}$, and p_b be the probability to select block b .

- p_b must be the same for every iteration $k \in \mathbb{N}$
(i.e. identically distributed over the iterations)
- $p_b \in]0, 1]$
(i.e. all blocks must have “a chance” to be selected)

EXAMPLES: For every $k \in \mathbb{N}$, one can choose, e.g.

- $p_b = \frac{1}{B}$ (all blocks selected with the same probability)
- $p_b = \frac{N_b}{N}$ (probability is proportional to the size of the block)

Stochastic BC-FB algorithm: Conditions for convergence

ASSUMPTIONS:

Let $b \in \{1, \dots, B\}$, and p_b be the probability to select block b .

- p_b must be the same for every iteration $k \in \mathbb{N}$
(i.e. identically distributed over the iterations)
- $p_b \in]0, 1]$
(i.e. all blocks must have “a chance” to be selected)

The sequence $(x_k)_{k \in \mathbb{N}}$ converges almost surely to a random variable $x^* \in \text{Argmin } f$.

Stochastic BC-FB algorithm: Particular case

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{b=1}^B h_b(\mathbf{L}_{b,b} \mathbf{x}^{(b)}) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{b=1}^B \mathbf{L}_{b,b}^* \nabla h_b(\mathbf{L}_{b,b} \mathbf{x}_k^{(b)}) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm: Particular case

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{b=1}^B h_b(\mathbf{L}_{b,b} \mathbf{x}^{(b)}) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{b=1}^B \mathbf{L}_{b,b}^* \nabla h_b(\mathbf{L}_{b,b} \mathbf{x}_k^{(b)}) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic BC-FB algorithm: Particular case

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ f(\mathbf{x}) = \sum_{b=1}^B h_b(\mathbf{L}_{b,b} \mathbf{x}^{(b)}) + \sum_{b=1}^B g_b(\mathbf{x}^{(b)}) \right\}$$

$$\text{Let } \tau = \sum_{d=1}^D \beta_d \left\| \sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{L}_{d,b}^* \right\|$$

For $k = 0, 1, \dots$

Let $\gamma_k \in]0, 2\tau^{-1}[$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b} \left(\mathbf{x}_k^{(b)} - \gamma_k \sum_{b=1}^B \mathbf{L}_{b,b}^* \nabla h_b(\mathbf{L}_{b,b} \mathbf{x}_k^{(b)}) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

Stochastic Block-Coordinate Primal-Dual Algorithm

General minimisation problem

$$\text{Find } \mathbf{x}^* \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} h(\mathbf{x}) + f(\mathbf{x}) + \sum_{d=1}^D g_d(\mathbf{L}_d \mathbf{x})$$

where

- $h: \mathbb{R}^N \rightarrow \mathbb{R}$ is convex, proper and β -Lipschitz differentiable
- $f \in \Gamma_0(\mathbb{R}^N)$
- $(\forall d \in \{1, \dots, D\}) g_d \in \Gamma_0(\mathbb{R}^{L_d})$
- $(\forall d \in \{1, \dots, D\}) \mathbf{L}_d \in \mathbb{R}^{L_d \times N}$

- Use Condat-Vũ primal-dual algorithm

Block-minimisation problem

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \sum_{b=1}^B h_b(\mathbf{x}^{(b)}) + f_b(\mathbf{x}^{(b)}) + \sum_{d=1}^D g_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$$

where $(\forall d \in \{1, \dots, D\})(\forall b \in \{1, \dots, B\})$

- $h_b: \mathbb{R}^{N_b} \rightarrow \mathbb{R}$ is convex, proper and β_b -Lipschitz differentiable
- $f_b \in \Gamma_0(\mathbb{R}^{N_b})$
- $g_d \in \Gamma_0(\mathbb{R}^{L_d})$
- $\mathbf{L}_{d,b} \in \mathbb{R}^{L_d \times N_b}$

OBJECTIVE: Design a convergent primal-dual algorithm such that:

- Update only a subset of the variable $\mathbf{x}^{(b)}$ per iteration
- Handle only a subset of the functions g_d per iteration
- Block-coordinate approach

Stochastic BC-PD algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \sum_{b=1}^B h_b(\mathbf{x}^{(b)}) + f_b(\mathbf{x}^{(b)}) + \sum_{d=1}^D g_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$$

For $k = 0, 1, \dots$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$ and $\mathbb{D}_k \subset \{1, \dots, D\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\tau_b f_b} \left(\mathbf{x}_k^{(b)} - \tau_b \left(\nabla h_b(\mathbf{x}_k^{(b)}) + \sum_{d=1}^D \mathbf{L}_{d,b}^* \mathbf{u}_k^{(d)} \right) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

For $d \in \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \operatorname{prox}_{\sigma_d g_d^*} \left(\mathbf{u}_k^{(d)} + \sigma_d \sum_{b=1}^B \mathbf{L}_{d,b} (2\mathbf{x}_{k+1}^{(b)} - \mathbf{x}_k^{(b)}) \right)$$

For $d \in \{1, \dots, D\} \setminus \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \mathbf{u}_k^{(d)}$$

Stochastic BC-PD algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \sum_{b=1}^B \mathbf{h}_b(\mathbf{x}^{(b)}) + \mathbf{f}_b(\mathbf{x}^{(b)}) + \sum_{d=1}^D \mathbf{g}_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$$

For $k = 0, 1, \dots$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$ and $\mathbb{D}_k \subset \{1, \dots, D\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\tau_b \mathbf{f}_b} \left(\mathbf{x}_k^{(b)} - \tau_b \left(\nabla \mathbf{h}_b(\mathbf{x}_k^{(b)}) + \sum_{d=1}^D \mathbf{L}_{d,b}^* \mathbf{u}_k^{(d)} \right) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

For $d \in \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \operatorname{prox}_{\sigma_d \mathbf{g}_d^*} \left(\mathbf{u}_k^{(d)} + \sigma_d \sum_{b=1}^B \mathbf{L}_{d,b} (2\mathbf{x}_{k+1}^{(b)} - \mathbf{x}_k^{(b)}) \right)$$

For $d \in \{1, \dots, D\} \setminus \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \mathbf{u}_k^{(d)}$$

Stochastic BC-PD algorithm

$$\text{Find } \mathbf{x}^* = (\mathbf{x}_1^*, \dots, \mathbf{x}_B^*) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \sum_{b=1}^B h_b(\mathbf{x}^{(b)}) + f_b(\mathbf{x}^{(b)}) + \sum_{d=1}^D g_d \left(\sum_{b=1}^B \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$$

For $k = 0, 1, \dots$

Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$ and $\mathbb{D}_k \subset \{1, \dots, D\}$

For $b \in \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\tau_b f_b} \left(\mathbf{x}_k^{(b)} - \tau_b \left(\nabla h_b(\mathbf{x}_k^{(b)}) + \sum_{d=1}^D \mathbf{L}_{d,b}^* \mathbf{u}_k^{(d)} \right) \right)$$

For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$

$$\mathbf{x}_{k+1}^{(b)} = \mathbf{x}_k^{(b)}$$

For $d \in \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \operatorname{prox}_{\sigma_d g_d^*} \left(\mathbf{u}_k^{(d)} + \sigma_d \sum_{b=1}^B \mathbf{L}_{d,b} (2\mathbf{x}_{k+1}^{(b)} - \mathbf{x}_k^{(b)}) \right)$$

For $d \in \{1, \dots, D\} \setminus \mathbb{D}_k$

$$\mathbf{u}_{k+1}^{(d)} = \mathbf{u}_k^{(d)}$$

Stochastic BC-PD algorithm: Conditions for convergence

ASSUMPTIONS:

Let $b \in \{1, \dots, B\}$, and p_b be the probability to select block b .

Let $d \in \{1, \dots, D\}$, and p_d be the probability to select functions d .

- p_b and p_d must be the same for every iteration $k \in \mathbb{N}$
(i.e. identically distributed over the iterations)
- $p_b \in]0, 1]$ and $p_d \in]0, 1]$
(i.e. all blocks/functions must have “a chance” to be selected)
- $(p_b)_{1 \leq b \leq B}$ and $(p_d)_{1 \leq d \leq D}$ are **not** independent

The sequence $(x_k)_{k \in \mathbb{N}}$ converges almost surely to a random variable x^* , solution to the primal problem.

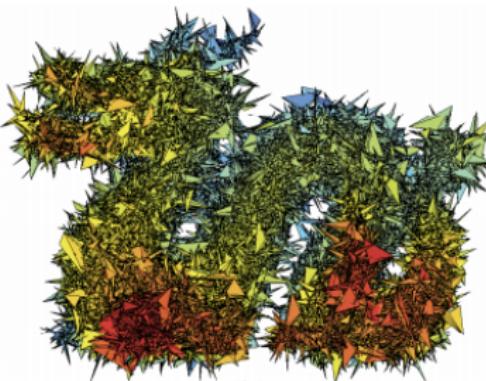
Application to 3D mesh denoising

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

- $\bar{\mathbf{x}} = (\bar{\mathbf{x}}^{(n)})_{1 \leq n \leq N} \in \mathbb{R}^{N \times 3}$ are the 3D coordinates of the graph nodes
- $\mathbf{w} \in \mathbb{R}^{N \times 3}$ is a realisation of a random white Gaussian noise
- $\mathbf{z} \in \mathbb{R}^{N \times 3}$ are observed noisy coordinates of $\bar{\mathbf{x}}$



(a)



(b)

Application to 3D mesh denoising

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

- $\bar{\mathbf{x}} = (\bar{\mathbf{x}}^{(n)})_{1 \leq n \leq N} \in \mathbb{R}^{N \times 3}$ are the 3D coordinates of the graph nodes
- $\mathbf{w} \in \mathbb{R}^{N \times 3}$ is a realisation of a random white Gaussian noise
- $\mathbf{z} \in \mathbb{R}^{N \times 3}$ are observed noisy coordinates of $\bar{\mathbf{x}}$

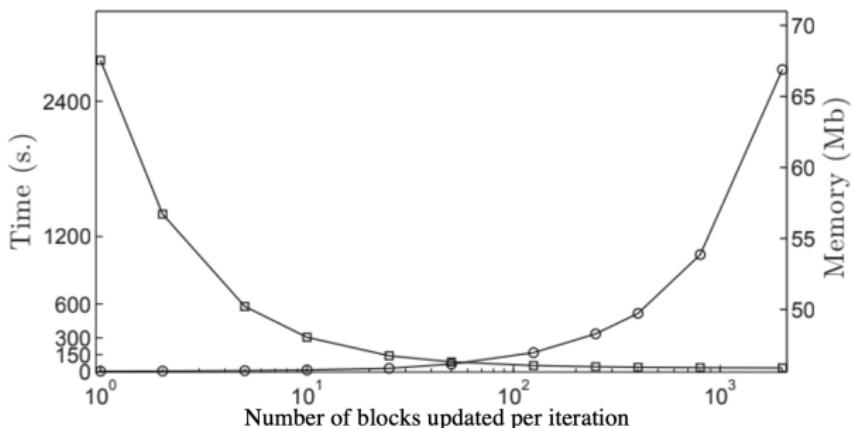
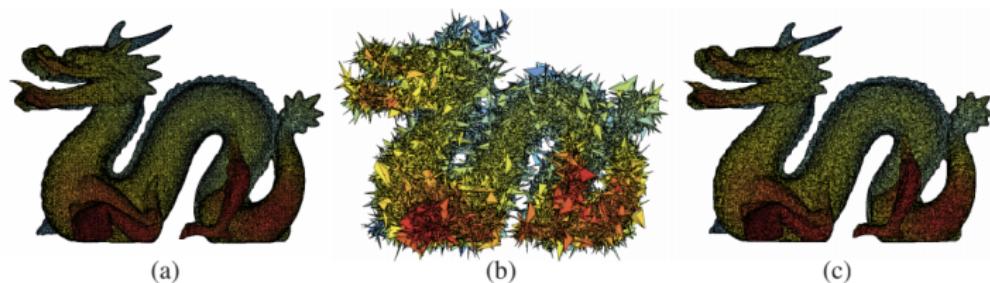
MINIMISATION PROBLEM:

$$\underset{\mathbf{x} \in \mathbb{R}^{N \times 3}}{\text{minimise}} \sum_{n=1}^N h_n(\mathbf{x}^{(n)}) + f_n(\mathbf{x}^{(n)}) + \sum_{n=1}^N g_n\left(\sum_{m=1}^N \mathbf{L}_{n,m} \mathbf{x}^{(m)}\right)$$

where ($\forall n \in \{1, \dots, N\}$)

- $h_n(\mathbf{x}^{(n)}) = \frac{1}{2} \|\mathbf{x}^{(n)} - \mathbf{z}^{(n)}\|^2$
- $f_n(\mathbf{x}^{(n)}) = \iota_{[\delta, \bar{\delta}]^3}(\mathbf{x}^{(n)})$
- $g_n\left(\sum_{m=1}^N \mathbf{L}_{n,m} \mathbf{x}^{(m)}\right) = \eta \sum_{D \in \{X, Y, Z\}} \|(\mathbf{x}^{(n,D)} - \mathbf{x}^{(i,D)})_{i \in \mathcal{V}_n}\|$

Application to 3D mesh denoising (source: Repetti *et al*, 2015)



Proximal Stochastic Gradient Algorithm

Minimisation problem

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

where

- $(\forall d \in \{1, \dots, D\}) h_d: \mathbb{R}^{L_d} \rightarrow \mathbb{R}$ is convex, proper and β_d -Lipschitz differentiable
- $(\forall d \in \{1, \dots, D\}) \mathbf{L}_d \in \mathbb{R}^{L_d \times N}$
- $g \in \Gamma_0(\mathbb{R}^N)$

OBJECTIVE:

Design a convergent forward-backward algorithm that will **approximate** the gradient of $\sum_d h_d \circ \mathbf{L}_d$ at each iteration

- Stochastic gradient approach

Proximal stochastic gradient algorithm in a nutshell

$$\text{Find } \mathbf{x}^* \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\} \text{ with } h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$$

For $k = 0, 1, \dots$

Select randomly $\mathbb{D}_k \subset \{1, \dots, D\}$

Build $\mathbf{u}^{(k)} \approx \nabla(h \circ \mathbf{L})(\mathbf{x}^{(k)})$ only with functions associated with \mathbb{D}_k

Update \mathbf{x}_{k+1} with a FB step using \mathbf{u}_k

Proximal stochastic gradient algorithm in a nutshell

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

For $k = 0, 1, \dots$

 Select randomly $\mathbb{D}_k \subset \{1, \dots, D\}$

 Build $\mathbf{u}^{(k)} \approx \nabla(h \circ \mathbf{L})(\mathbf{x}^{(k)})$ only with functions associated with \mathbb{D}_k

 Update \mathbf{x}_{k+1} with a FB step using \mathbf{u}_k

Proximal stochastic gradient algorithm

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

For $k = 0, 1, \dots$

Let $\mathbf{u}_k \approx \nabla(h \circ \mathbf{L})(\mathbf{x}_k)$

$\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \mathbf{u}_k)$

Proximal stochastic gradient algorithm

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

For $k = 0, 1, \dots$

Let $\mathbf{u}_k \approx \nabla(h \circ \mathbf{L})(\mathbf{x}_k)$
 $\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \mathbf{u}_k)$

QUESTIONS:

- Strategy to choose $(\mathbf{u}_k)_{k \in \mathbb{N}}$?
- How to choose the step-size?

Proximal stochastic gradient: Approximation strategies

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

For $k = 0, 1, \dots$

Let $\mathbf{u}_k \approx \nabla(h \circ \mathbf{L})(\mathbf{x}_k)$
 $\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \mathbf{u}_k)$

- **Usual strategy:** Select randomly $\mathbb{D}_k \subset \{1, \dots, D\}$

Set $\mathbf{u}_k = \frac{1}{\#\mathbb{D}_k} \sum_{d \in \mathbb{D}_k} \nabla(h_d \circ \mathbf{L}_d)(\mathbf{x}_k)$

Choose $\gamma_k \propto \frac{1}{k}$

Particular case: Select randomly $d_k \in \{1, \dots, D\}$
Set $\mathbf{u}_k = \nabla(h_{d_k} \circ \mathbf{L}_{d_k})(\mathbf{x}_k)$

Proximal stochastic gradient: Approximation strategies

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

For $k = 0, 1, \dots$

Let $\mathbf{u}_k \approx \nabla(h \circ \mathbf{L})(\mathbf{x}_k)$
 $\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \mathbf{u}_k)$

- Other possible strategy: Select randomly $\mathbb{D}_k \subset \{1, \dots, D\}$,

Set $\mathbf{u}_k = \mathbf{u}^{(k-1)} + \frac{1}{D} \sum_{d \in \mathbb{D}_k} \left(\nabla(h_d \circ \mathbf{L}_d)(\mathbf{x}_k) - \nabla(h_d \circ \mathbf{L}_d)(\mathbf{x}^{(k-1)}) \right)$

Choose $\gamma_k \propto \beta^{-1}$ where β is the Lipschitz constant of $\nabla(h \circ \mathbf{L})$



This strategy may necessitate to store D gradients of size N !

Proximal stochastic gradient: Convergence

Find $\mathbf{x}^* \in \operatorname{Argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) = \left\{ h(\mathbf{L}\mathbf{x}) + g(\mathbf{x}) \right\}$ with $h(\mathbf{L}\mathbf{x}) = \frac{1}{D} \sum_{d=1}^D h_d(\mathbf{L}_d \mathbf{x})$

Let $\theta \in]1/2, 1]$

For $k = 0, 1, \dots$

$$\gamma_k = \frac{1}{(k+1)^\theta}$$

Select randomly $\mathbb{D}_k \subset \{1, \dots, D\}$

$$\mathbf{u}_k = \frac{1}{\#\mathbb{D}_k} \sum_{d \in \mathbb{D}_k} \nabla(h_d \circ \mathbf{L}_d)(\mathbf{x}_k)$$

$$\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}(\mathbf{x}_k - \gamma_k \mathbf{u}_k)$$

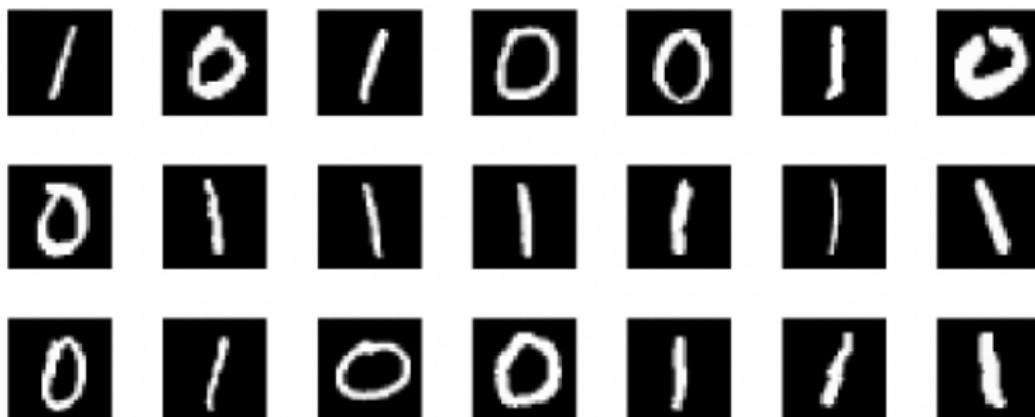
The sequence $(\mathbf{x}_k)_{k \in \mathbb{N}}$ converges almost surely to a random variable $\mathbf{x}^* \in \operatorname{Argmin} f$.

Example: binary SVM

BINARY CLASSIFICATION:

- Pair an input $\mathbf{x} \in \mathbb{R}^N$ to a binary output $z \in \{-1, +1\}$: $z = \text{sign}(\mathbf{x}^\top \mathbf{w})$
- Classifier trained on a training set of pairs

$$\mathcal{S} = \{(\mathbf{x}_\ell, z_\ell)_{1 \leq \ell \leq L} \mid (\forall \ell \in \{1, \dots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N, \quad z_\ell \in \{-1, +1\}\}$$



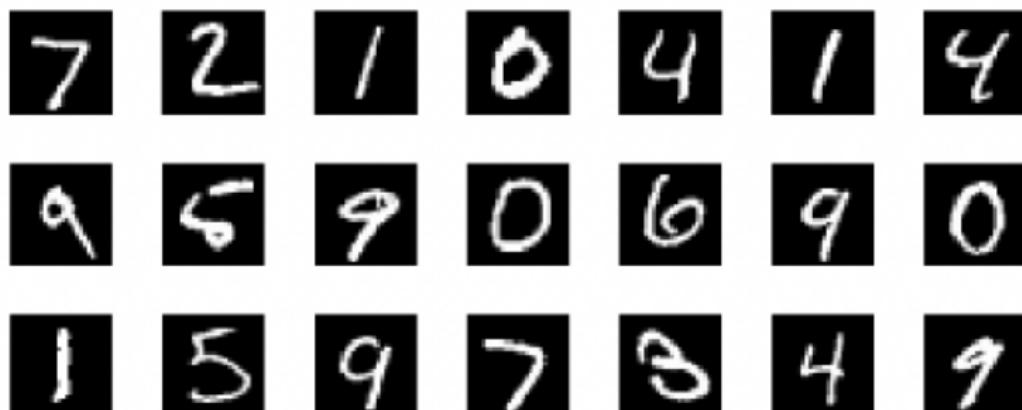
If digit on \mathbf{x}_ℓ is 0, then $z_\ell = -1$, otherwise $z_\ell = 1$

Example: binary SVM

BINARY CLASSIFICATION:

- Pair an input $\mathbf{x} \in \mathbb{R}^N$ to a binary output $z \in \{-1, +1\}$: $z = \text{sign}(\mathbf{x}^\top \mathbf{w})$
- Classifier trained on a training set of pairs

$$\mathcal{S} = \{(\mathbf{x}_\ell, z_\ell)_{1 \leq \ell \leq L} \mid (\forall \ell \in \{1, \dots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N, \quad z_\ell \in \{-1, +1\}\}$$



If digit on \mathbf{x}_ℓ is 0, then $z_\ell = -1$, otherwise $z_\ell = 1$

Example: binary SVM

BINARY CLASSIFICATION:

- Pair an input $\mathbf{x} \in \mathbb{R}^N$ to a binary output $z \in \{-1, +1\}$: $z = \text{sign}(\mathbf{x}^\top \mathbf{w})$
- Classifier trained on a training set of pairs
$$\mathcal{S} = \{(\mathbf{x}_\ell, z_\ell)_{1 \leq \ell \leq L} \mid (\forall \ell \in \{1, \dots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N, z_\ell \in \{-1, +1\}\}$$

MINIMISATION PROBLEM:

$$\underset{\mathbf{w}}{\text{minimise}} \ h(\mathbf{w}) + g(\mathbf{w})$$

- $h(\mathbf{w}) = \frac{1}{L} \sum_{\ell=1}^L h(z_\ell - \mathbf{x}_\ell^\top \mathbf{w})$ loss function (e.g., h Huber loss)
- $g(\mathbf{w}) = \eta \|\mathbf{w}\|_1$ regularisation (sparse learning)

Example: binary SVM

RESULT COMPARISON:

