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

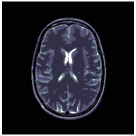
Motivation for BC approaches: non convex problems

Blind deconvolution:

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

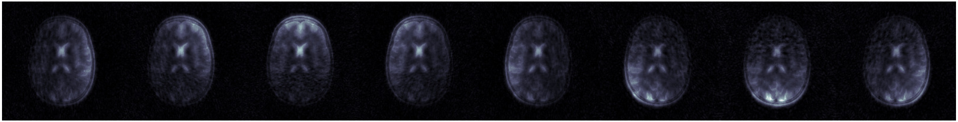
Motivation for BC approaches: non convex problems

BLIND DECONVOLUTION: Example in parallel MRI



acquisition ↓

↑ reconstruction?

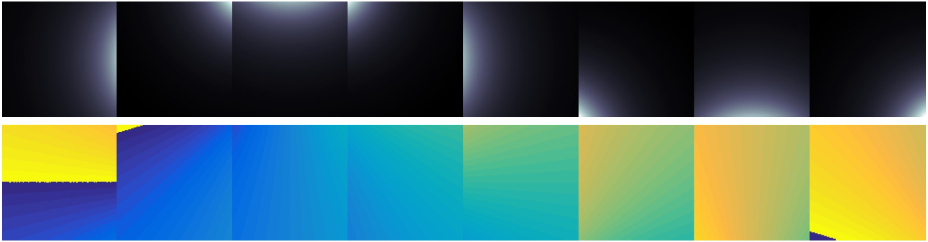


Motivation for BC approaches: non convex problems

BLIND DECONVOLUTION: Example in parallel MRI

$$(\forall c \in \{1, \ldots, C\}) \quad \boxed{\mathbf{z}_c = \mathbf{MF}(\overline{\mathbf{s}}_c \cdot \overline{\mathbf{x}})}$$

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



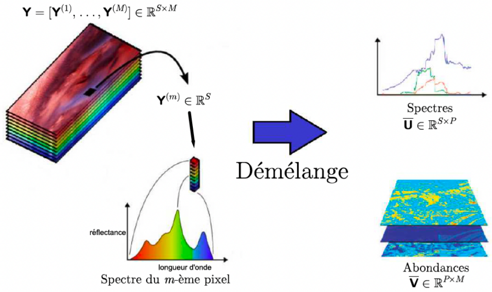
Example of simulated sensitivity maps (magnitude and phase)

Motivation for BC approaches: non convex problems

Other non-convex problems:

- Dictionary learning: $\underset{\mathbf{x},\boldsymbol{\Psi}}{\text{minimise}}\;\; H(\mathbf{x}) + R(\boldsymbol{\Psi}\mathbf{x})$
  - ⇝ $\mathbf{x}$ unknown image
  - ⇝ $\boldsymbol{\Psi}$ unknown sparsifying operator

- Non-negative Matrix Factorisation (NMF): $\underset{\mathbf{U},\mathbf{V}}{\text{minimise}}\;\; H(\mathbf{UV}) + R_1(\mathbf{U}) + R_2(\mathbf{V})$



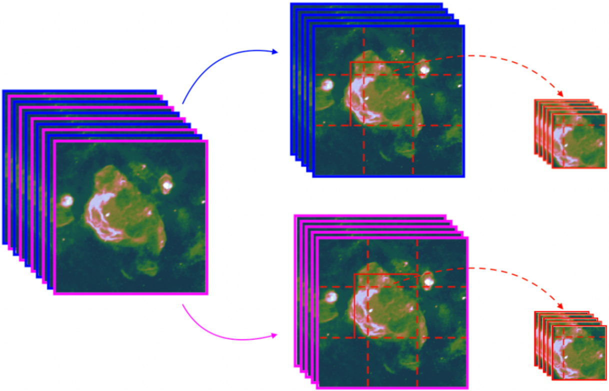
Motivation for block-coordinate approaches

Large dimensions: High dimensional images

- **Inverse problem:** $\mathbf{z} = \mathbf{\Phi}\overline{\mathbf{x}} + \mathbf{w}$
  - $\rightsquigarrow$ $\overline{\mathbf{x}} \in \mathbb{R}^N$ unknown with N very large ($\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ fat matrix)
  - $\rightsquigarrow$ Split $\overline{\boldsymbol{x}} = (\overline{\mathsf{x}}^{(b)})_{1 \leqslant b \leqslant B}$ into blocks
- **Minimisation problem:** $\underset{\mathbf{x},\mathbf{\Phi}}{\text{minimise}} \ \frac{1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{z}\|^2 + R(\mathbf{x})$
  - $\rightsquigarrow$ $R(\mathbf{x}) = \sum_{1 \leqslant b \leqslant B} R_b(\mathsf{x}^{(b)})$
- **Approach:** Use *block splitting* algorithms
  - $\rightsquigarrow$ Only handle a subset of $(\mathsf{x}^{(b)})_{1 \leqslant b \leqslant B}$ at each iteration
  - $\rightsquigarrow$ OR parallelise $(\mathsf{x}^{(b)})_{1 \leqslant b \leqslant B}$

Motivation for block-coordinate approaches

LARGE DIMENSIONS: **High dimensional images**



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

(source: Thouvenin *et al*, 2022)

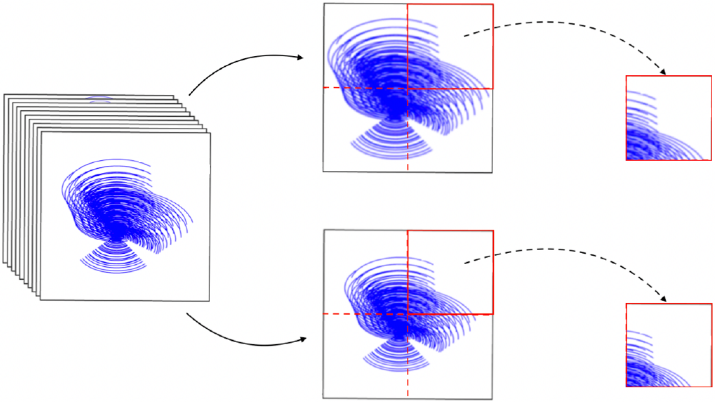
Motivation for block-coordinate approaches

LARGE DIMENSIONS: High dimensional data

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

Motivation for block-coordinate approaches

Large dimensions: **High dimensional data**



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

(source: Thouvenin *et al*, 2022)

Motivation for block-coordinate approaches

LARGE DIMENSIONS: High dimensional data

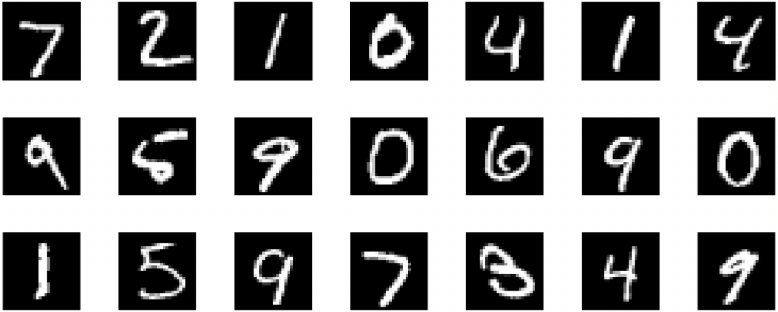
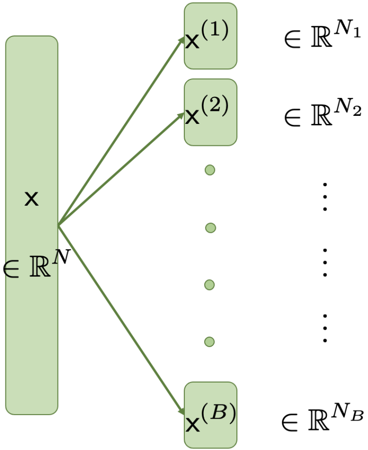


Figure 1: Sample of the MNIST dataset (http://yann.lecun.com/exdb/mnist/).

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

Variable splitting



$$N_1 + N_2 + \ldots + N_B = N$$

WHY? Split a large object into smaller objects

Operator splitting

$$\mathbf{L} \in \mathbb{R}^{L\times N} = \begin{array}{cccc|l} \mathbf{L}_{1,1} & \mathbf{L}_{1,2} & \cdots & \mathbf{L}_{1,B} & L_1 \\ \mathbf{L}_{2,1} & \mathbf{L}_{2,2} & \cdots & \mathbf{L}_{2,B} & L_2 \\ \vdots & & & \vdots & \\ \vdots & & & \vdots & \\ \mathbf{L}_{D,1} & \mathbf{L}_{D,2} & \cdots & \mathbf{L}_{D,B} & L_D \\ \hline N_1 & N_2 & & N_B & \end{array} \quad \begin{array}{l} L_1 + \dots + L_D = L \\ N_1 + \dots + N_B = N \end{array}$$

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

  - 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. $\mathbf{L} = (\mathbf{L}_{d,b})_{1 \leqslant d \leqslant D,\, 1 \leqslant b \leqslant B}$

  - 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 $\mathbf{x}^\star = ([\mathrm{x}^\star]^{(b)})_{1\leqslant b\leqslant B} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \; \{\mathrm{f}(\mathbf{x}) := \mathrm{h}(\mathbf{x}) + \mathrm{g}(\mathbf{x})\}$.

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

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

For $k = 0, 1, \dots$
  for $b = 1, \dots, B$
    $\mathbf{x}_{k+1}^{(b)} = \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathrm{f}(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \dots, \mathbf{x}_k^{(B)})$

Nonconvex minimisation problem

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

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

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

For $k = 0, 1, \dots$
  for $b = 1, \dots, B$
    $\mathbf{x}_{k+1}^{(b)} = \underset{\mathbf{u}}{\operatorname{argmin}}\, \mathsf{f}(\mathbf{x}_k^{(1)}, \dots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \dots, \mathbf{x}_k^{(B)})$

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

Nonconvex minimisation problem

Find $\mathbf{x}^\star = ([\mathrm{x}^\star]^{(b)})_{1\leqslant b\leqslant B} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}} \; \{\mathrm{f}(\mathbf{x}) := \mathrm{h}(\mathbf{x}) + \mathrm{g}(\mathbf{x})\}.$

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

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

For $k = 0, 1, \ldots$
for $b = 1, \ldots, B$

$$\mathbf{x}_{k+1}^{(b)} = \underset{\mathbf{u}}{\mathrm{argmin}}\; \mathrm{f}(\mathbf{x}_k^{(1)}, \ldots, \mathbf{x}_k^{(b-1)}, \mathbf{u}, \mathbf{x}_k^{(b+1)}, \ldots, \mathbf{x}_k^{(B)}) + \frac{1}{2\gamma_k}\|\mathbf{u} - \mathbf{x}_k^{(b)}\|^2$$

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

Nonconvex minimisation problem

Find $\mathbf{x}^\star = ([\mathrm{x}^\star]^{(b)})_{1\leqslant b\leqslant B} \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \; \{\mathsf{f}(\mathbf{x}) := \mathsf{h}(\mathbf{x}) + \mathsf{g}(\mathbf{x})\}$.

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

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

For $k = 0, 1, \ldots$
  for $b = 1, \ldots, B$
    $\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k \mathsf{f}_b}(\mathbf{x}_k^{(b)})$

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

Nonconvex minimisation problem

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

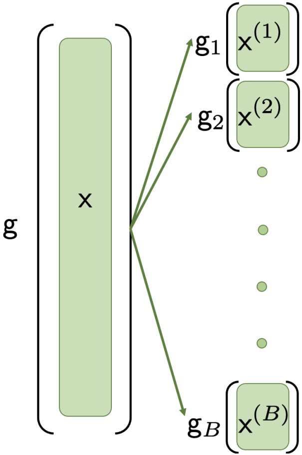
- $\mathsf{h}\colon \mathbb{R}^N \to \mathbb{R}$ is β-Lipschitz differentiable
- $\mathsf{g}\colon \mathbb{R}^N \to]-\infty, +\infty]$ additively separable

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

For $k = 0, 1, \ldots$
  for $b = 1, \ldots, B$
    $\mathbf{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k \mathsf{f}_b}(\mathbf{x}_k^{(b)})$

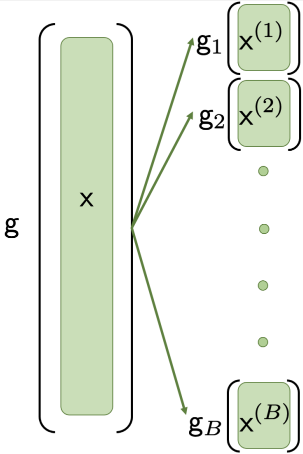
- Convergence? Usually necessitates strong conditions (e.g., strong convexity)
- Add an ℓ_2 term to obtain convergence... Leading to alternated proximal method
- Idea: Use alternated FB iterations to fully exploit the splitting structure of f

Additively separable functions: g



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

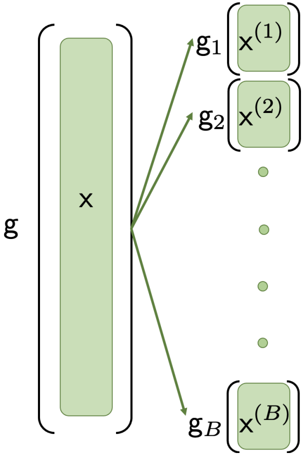
Additively separable functions: g



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

$$(\forall b \in \{1, \dots, B\}) \quad \mathbf{g}_b : \mathbb{R}^{N_b} \to]-\infty, +\infty] \text{ l.s.c.}$$

Additively separable functions: g



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

$$(\forall b \in \{1, \dots, B\})$$
$$\mathrm{g}_b : \mathbb{R}^{N_b} \to]-\infty, +\infty] \text{ l.s.c.}$$

$$\operatorname{prox}_{\mathrm{g}}(\mathbf{x}) = \left(\operatorname{prox}_{\mathrm{g}_b}(\mathbf{x}^{(b)})\right)_{1 \le b \le B}$$

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \operatorname{dom} \mathrm{g}$
For $k = 0, 1, \ldots$
  Let $b_k \in \{1, \ldots, B\}$ and $\gamma_k \in]0, \beta_{b_k}^{-1}[$.
  $\mathrm{x}_{k+1}^{(b_k)} \in \operatorname{prox}_{\gamma_k^{-1} \mathrm{g}_{b_k}} \left(\mathrm{x}_k^{(b_k)} - \gamma_k \nabla_{b_k} \mathrm{h}(\mathbf{x}_k) \right)$
  for $b \in \{1, \ldots, B\} \setminus \{b_k\}$
    $\mathrm{x}_{k+1}^{(b)} = \mathrm{x}_k^{(b)}$

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

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \operatorname{dom} \mathrm{g}$
For $k = 0, 1, \ldots$
Let $b_k \in \{1, \ldots, B\}$, $\gamma_k \in]0,1[$ and $\mathbf{A}_{b_k} \in \mathbb{R}^{N_{b_k} \times N_{b_k}}$.

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

for $b \in \{1, \ldots, B\} \setminus \{b_k\}$
$\mathrm{x}_{k+1}^{(b)} = \mathrm{x}_k^{(b)}$

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

Block coordinate (VM)FB algorithm

Let $\mathbf{x}_0 \in \operatorname{dom} \mathrm{g}$
For $k = 0, 1, \ldots$
  Let $b_k \in \{1, \ldots, B\}$, $\gamma_k \in]0,1[$ and $\mathbf{A}_{b_k} \in \mathbb{R}^{N_{b_k} \times N_{b_k}}$.
  $\mathrm{x}_{k+1}^{(b_k)} \in \operatorname{prox}^{\mathbf{A}_{b_k}}_{\gamma_k^{-1} \mathrm{g}_{b_k}} \left(\mathrm{x}_k^{(b_k)} - \gamma_k \mathbf{A}_{b_k}^{-1} \nabla_{b_k} \mathrm{h}(\mathbf{x}_k) \right)$
  for $b \in \{1, \ldots, B\} \setminus \{b_k\}$
    $\mathrm{x}_{k+1}^{(b)} = \mathrm{x}_k^{(b)}$

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

- Quasi-cyclic rule: there exists $K \geqslant B$ such that, for every $k \in \mathbb{N}$, $\{1, \ldots, B\} \subset \{b_k, \ldots, b_{k+K-1}\}$.

BCFB algorithm: cyclic/quasi-cyclic updating rules

⋆ Cyclic rule: update sequentially blocks $1, 2, \ldots, B$.

⋆ Quasi-cyclic rule: there exists $K \geqslant B$ such that, for every $k \in \mathbb{N}$,
$\{1, \ldots, B\} \subset \{b_k, \ldots, b_{k+K-1}\}$.

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

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

BCFB algorithm: cyclic/quasi-cyclic updating rules

⋆ **Cyclic rule:** update sequentially blocks $1, 2, \ldots, B$.

⋆ **Quasi-cyclic rule:** there exists $K \geqslant B$ such that, for every $k \in \mathbb{N}$,
$\{1, \ldots, B\} \subset \{b_k, \ldots, b_{k+K-1}\}$.

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

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

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 $\mathrm{x}_k^{(b_k)}$ for the restriction of h to the block b_k:
$$\mathrm{y} \in \mathbb{R}^{N_{b_k}} \mapsto \mathrm{h}\left(\mathrm{x}_k^{(1)},\dots,\mathrm{x}_k^{(b_k-1)},\mathrm{y},\mathrm{x}_k^{(b_k+1)},\dots,\mathrm{x}_k^{(J)}\right).$$
- $(\exists(\underline{\nu},\overline{\nu}) \in \,]0,+\infty[^2)(\forall k\in\mathbb{N})(\forall \mathbf{u}\in\mathbb{R}^{N_{b_k}})\ \underline{\nu}\|\mathbf{u}\|^2 \leqslant \mathbf{u}^\top \mathbf{A}_{b_k}\mathbf{u} \leqslant \overline{\nu}\|\mathbf{u}\|^2$
- 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:

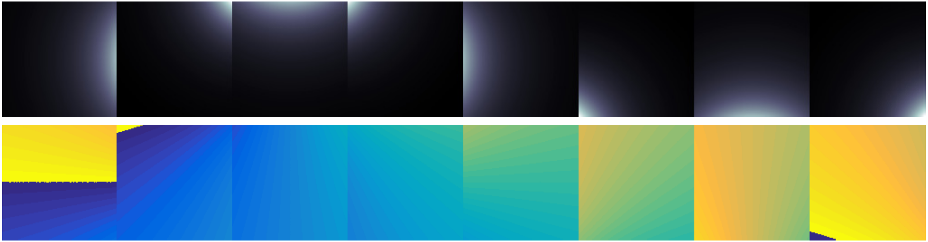
- $(\forall b_k \in \{1, \ldots, B\})$ $\mathbf{A}_{b_k}$ SPD satisfies the MM assumption at $x_k^{(b_k)}$
- $(\exists(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2)(\forall k \in \mathbb{N})(\forall \mathbf{u} \in \mathbb{R}^{N_{b_k}})$ $\underline{\nu}\|\mathbf{u}\|^2 \leqslant \mathbf{u}^\top \mathbf{A}_{b_k} \mathbf{u} \leqslant \overline{\nu}\|\mathbf{u}\|^2$
- 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: $(\forall k \in \mathbb{N})$ $\gamma_k \in]0, 1[$

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

Example: Calibration and Imaging in MR

Non-linear inverse problem: $(\forall c \in \{1,\ldots,C\})$ $\boxed{\mathbf{z}_c = \mathbf{MF}(\overline{\mathbf{s}}_c \cdot \overline{\mathbf{x}}) + \mathbf{w}_c}$

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



Example of simulated sensitivity maps (magnitude and phase)

Example: Calibration and Imaging in MR

Non-linear inverse problem: $(\forall c \in \{1, \dots, C\})$ $\boxed{\mathbf{z}_c = \mathbf{MF}(\overline{\mathbf{s}}_c \cdot \overline{\mathbf{x}}) + \mathbf{w}_c}$

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

Non-convex minimisation problem:

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

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

Example: Calibration and Imaging in MR

NON-LINEAR INVERSE PROBLEM: $(\forall c \in \{1, \ldots, C\})$ $\boxed{\mathbf{z}_c = \mathbf{MF}(\overline{\mathbf{s}}_c \cdot \overline{\mathbf{x}}) + \mathbf{w}_c}$

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

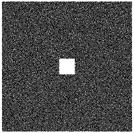
NON-CONVEX MINIMISATION PROBLEM:

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

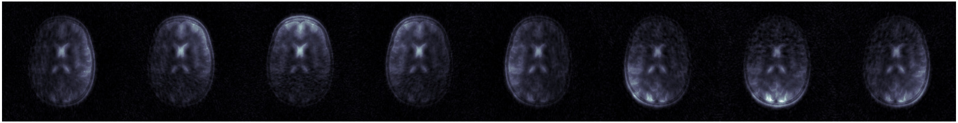
- $R_{\mathbf{x}}(\mathbf{x}) = \eta \|\boldsymbol{\Psi}\mathbf{x}\|_1$ with $\boldsymbol{\Psi}$ db8 wavelet transform
- $\mathbf{s}_c$ are smooth spatial functions, i.e., band limited
  $\rightsquigarrow$ Estimate (only) the non-zero central Fourier coefficients $\overline{\mathbf{d}}_c \in \mathbb{C}^S$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

Example: Calibration and Imaging in MR

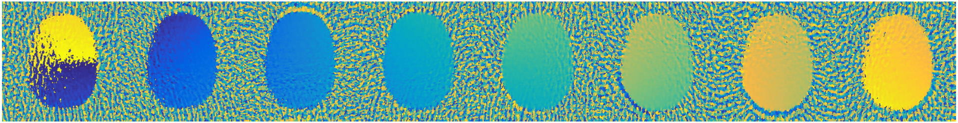
SIMULATED DATA: $N = 200 \times 200,\ C = 8,\ M = 8 \times 10683$



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



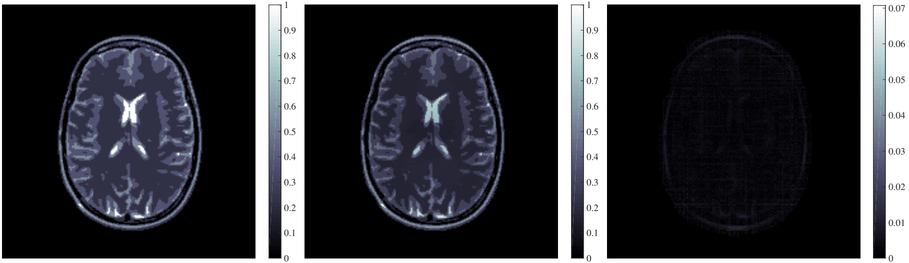
Magnitude physical coil images



Phase physical coil images

Example: Calibration and Imaging in MR

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



$\bar{\mathbf{x}}$ $\mathbf{x}^\star$ residual image

Example: Calibration and Imaging in radio-astronomy

NON-LINEAR INVERSE PROBLEM: $\boxed{\mathbf{z} = \overline{\mathbf{G}}\mathbf{F}\overline{\mathbf{x}} + \mathbf{w}}$

- $\overline{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown sky intensity image
- $\overline{\mathbf{G}}_c \in \mathbb{C}^{M \times N} \rightsquigarrow$ 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 $\overline{\mathbf{u}}_\alpha$ and $\overline{\mathbf{u}}_\beta$ associated with antennas (α, β) [Repetti *et al*, 2017]

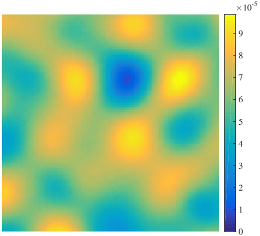
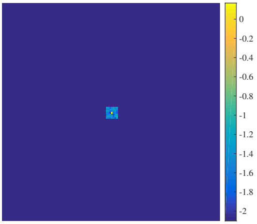
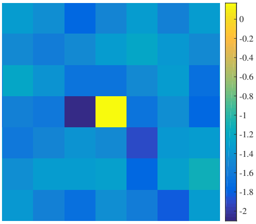


Image space



Fourier space



Fourier kernels

Example: Calibration and Imaging in radio-astronomy

NON-LINEAR INVERSE PROBLEM: $\boxed{\mathbf{z} = \overline{\mathbf{G}}\mathbf{F}\overline{\mathbf{x}} + \mathbf{w}}$

- $\overline{\mathbf{x}} \in \mathbb{R}^N \rightsquigarrow$ unknown sky intensity image
- $\overline{\mathbf{G}}_c \in \mathbb{C}^{M\times N} \rightsquigarrow$ 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 $\overline{\mathbf{u}}_\alpha$ and $\overline{\mathbf{u}}_\beta$ associated with antennas (α, β)

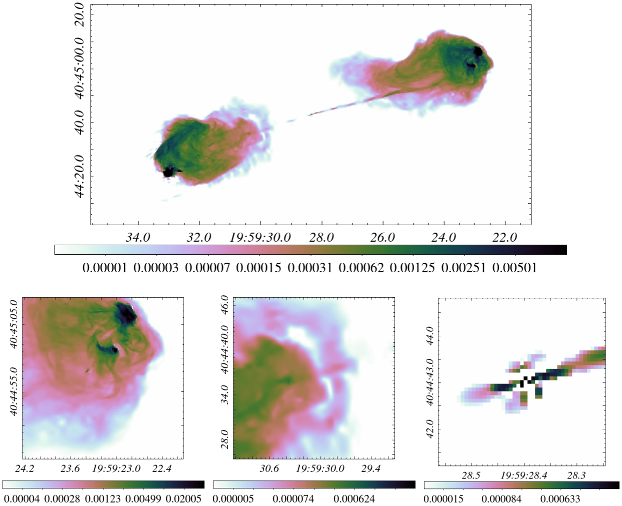
NON-CONVEX MINIMISATION PROBLEM:

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

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

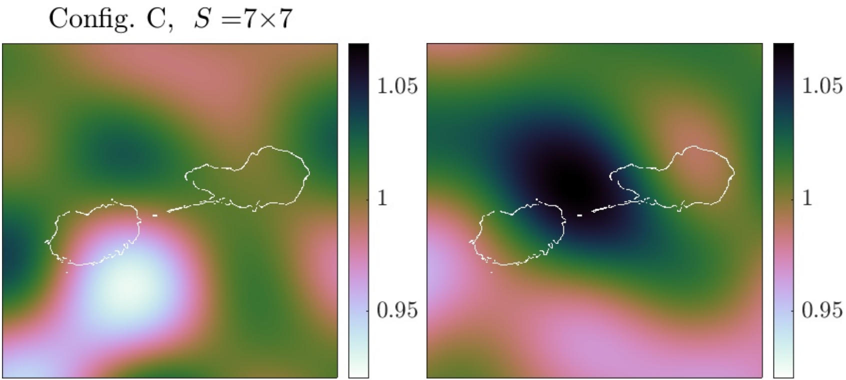
Example: Calibration and Imaging in radio-astronomy

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)



Stochastic Block-Coordinate
Forward-Backward Algorithm

Minimisation problem involving a linear operator

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

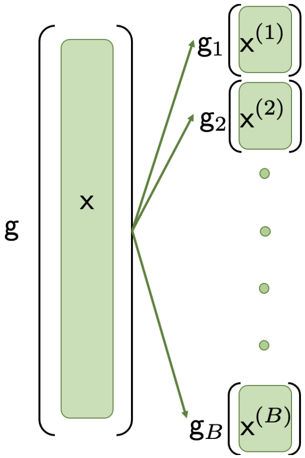
- $\mathrm{h}\colon \mathbb{R}^L \to \mathbb{R}$ is convex, proper and β-Lipschitz differentiable
- $\mathbf{L} \in \mathbb{R}^{L\times N}$
- $\mathrm{g} \in \Gamma_0(\mathbb{R}^N)$

☛ This problem can be solved using FB:

Let $\mathbf{x}_0 \in \operatorname{dom} \mathrm{g}$
For $k = 0, 1, \ldots$
  Let $\gamma_k \in]0, 2/(\beta\|\mathbf{L}\|^2)[$
  $\mathbf{x}_{k+1} = \operatorname{prox}_{\gamma_k g}\Big(\mathbf{x}_k - \gamma_k \mathbf{L}^* \nabla \mathrm{h}(\mathbf{L}\mathbf{x}_k)\Big)$

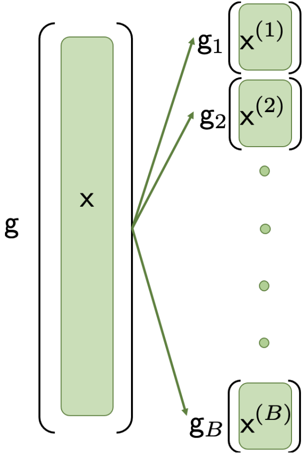
ISSUE: Not completely **scalable**

Additively separable functions: g



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

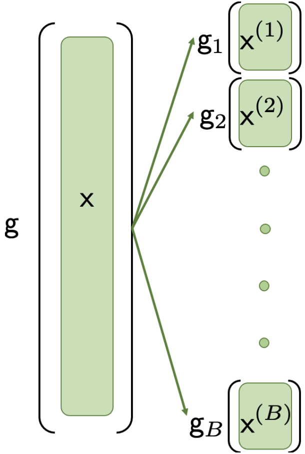
Additively separable functions: g



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

$$(\forall b \in \{1, \ldots, B\}) \quad \mathbf{g}_b \in \Gamma_0(\mathbb{R}^{N_b})$$

Additively separable functions: g

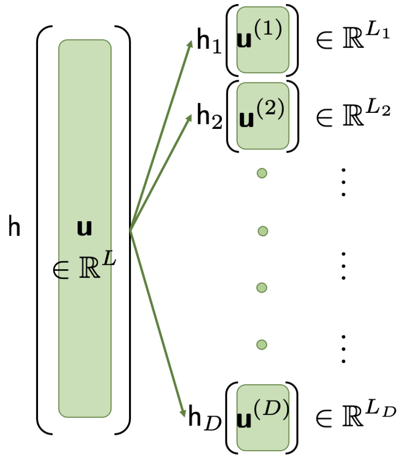


$$\mathbf{g}(\mathbf{x}) = \sum_{b=1}^{B} \mathbf{g}_b(\mathbf{x}^{(b)})$$

$$(\forall b \in \{1, \ldots, B\}) \quad \mathbf{g}_b \in \Gamma_0(\mathbb{R}^{N_b})$$

$$\mathrm{prox}_{\mathbf{g}}(\mathbf{x}) = \left(\mathrm{prox}_{\mathbf{g}_b}(\mathbf{x}^{(b)})\right)_{1 \le b \le B}$$

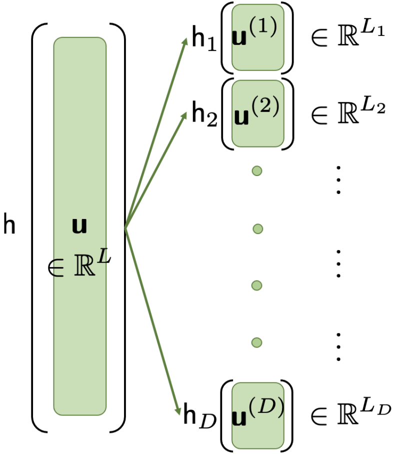
Additively separable functions: h



$(\forall d \in \{1, \ldots, D\})$

- $\mathbf{u}^{(d)} = \sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)}$

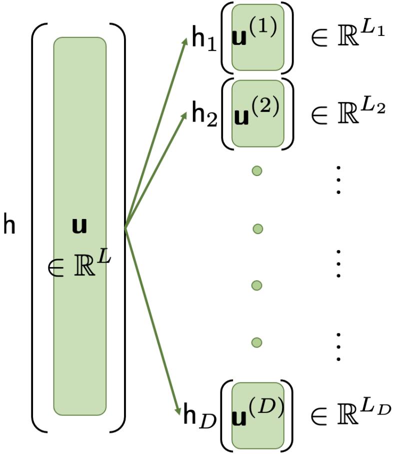
Additively separable functions: h



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

- $\mathbf{u}^{(d)} = \sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)}$
- h_d convex and β_d-Lipschitz differentiable

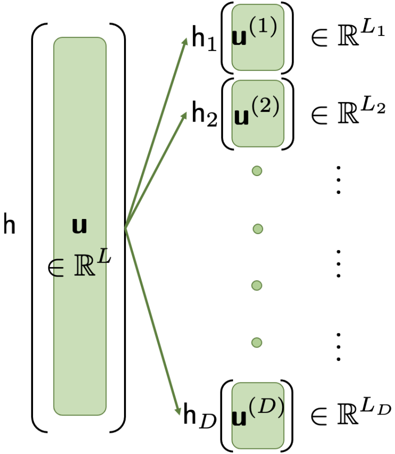
Additively separable functions: h



$(\forall d \in \{1, \ldots, D\})$

- $\mathbf{u}^{(d)} = \sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)}$
- h_d convex and β_d-Lipschitz differentiable
- $\mathsf{h}(\mathbf{Lx}) = \sum_{d=1}^{D} \mathsf{h}_d \left(\sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$

Additively separable functions: h



$(\forall d \in \{1, \ldots, D\})$

- $\mathbf{u}^{(d)} = \sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)}$
- h_d convex and β_d-Lipschitz differentiable
- $h(\mathbf{L}\mathbf{x}) = \sum_{d=1}^{D} h_d \left(\sum_{b=1}^{B} \mathbf{L}_{d,b} \mathbf{x}^{(b)} \right)$
- $\nabla h(\mathbf{L}\mathbf{x})$?

Additively separable functions: h (gradient)

$$
(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathsf{h}(\mathbf{L}\mathbf{x}) = \sum_{d=1}^{D} \mathsf{h}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\mathbf{x}^{(b)}\Big) = \sum_{d=1}^{D} \mathsf{h}_d\Big([\mathbf{L}_{d,1},\ldots,\mathbf{L}_{d,B}]\,\mathbf{x}\Big)
$$

Gradient?

$$
(\forall \mathbf{x} \in \mathbb{R}^N) \quad \mathsf{h}(\mathbf{L}\mathbf{x}) = \sum_{d=1}^{D} [\mathbf{L}_{d,1},\ldots,\mathbf{L}_{d,B}]^* \, \nabla \mathsf{h}_d\Big([\mathbf{L}_{d,1},\ldots,\mathbf{L}_{d,B}]\,\mathbf{x}\Big) = \sum_{d=1}^{D} \begin{bmatrix} \mathbf{L}_{d,1}^* \\ \vdots \\ \mathbf{L}_{d,B}^* \end{bmatrix} \nabla \mathsf{h}_d\Big([\mathbf{L}_{d,1},\ldots,\mathbf{L}_{d,B}]\,\mathbf{x}\Big)
$$

Block-minimisation problem

Find $\mathbf{x}^\star = (\mathbf{x}_1^\star, \ldots, \mathbf{x}_B^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \sum_{d=1}^{D} \mathsf{h}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\mathbf{x}^{(b)}\Big) + \sum_{b=1}^{B} \mathsf{g}_b(\mathbf{x}^{(b)})$.

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

- $\mathsf{h}_d\colon \mathbb{R}^{L_d} \to \mathbb{R}$ is convex, proper and β_d-Lipschitz differentiable
- $\mathbf{L}_{d,b} \in \mathbb{R}^{L_d \times N_b}$
- $\mathsf{g}_b \in \Gamma_0(\mathbb{R}^{N_b})$

OBJECTIVE:

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

☛ Block-coordinate approach

Stochastic BC-FB algorithm in a nutshell

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - Update $\boldsymbol{x}_{k+1}^{(b)}$ with “sub”-FB step
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - Keep $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm in a nutshell

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - Update $\boldsymbol{x}_{k+1}^{(b)}$ with "sub"-FB step
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - Keep $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm in a nutshell

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - Update $\boldsymbol{x}_{k+1}^{(b)}$ with “sub”-FB step
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - Keep $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\Big(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \nabla \mathsf{h}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\boldsymbol{x}_k^{(b)}\Big)\Big)$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\Big(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \nabla \mathsf{h}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\boldsymbol{x}_k^{(b)}\Big)\Big)$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm

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

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\Big(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \nabla \mathsf{h}_d\big([\mathbf{L}_{d,1}, \ldots, \mathbf{L}_{d,B}]\boldsymbol{x}_k\big)\Big)$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm

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

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

For $k = 0, 1, \dots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \mathrm{prox}_{\gamma_k g_b}\Big(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \nabla \mathsf{h}_d\big([\mathbf{L}_{d,1}, \dots, \mathbf{L}_{d,B}]\boldsymbol{x}_k\big)\Big)$
- For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm: Conditions for convergence

ASSUMPTIONS:

Let $b \in \{1, \ldots, B\}$, 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 $b \in \{1, \ldots, B\}$, 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 $(\boldsymbol{x}_k)_{k \in \mathbb{N}}$ converges almost surely to a random variable $\boldsymbol{x}^\star \in \operatorname{Argmin} \mathsf{f}$.

[Combettes & Pesquet, 2014]

Stochastic BC-FB algorithm: Particular case

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

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

For $k = 0, 1, \dots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \dots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\left(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{b=1}^{B} \mathbf{L}_{b,b}^* \nabla \mathsf{h}_b\left(\mathbf{L}_{b,b}\boldsymbol{x}_k^{(b)}\right)\right)$
- For $b \in \{1, \dots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm: Particular case

Find $\mathbf{x}^\star = (\mathbf{x}_1^\star, \ldots, \mathbf{x}_B^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \left\{ \mathsf{f}(\mathbf{x}) = \sum_{b=1}^{B} \mathsf{h}_b(\mathbf{L}_{b,b}\mathbf{x}^{(b)}) + \sum_{b=1}^{B} \mathsf{g}_b(\mathbf{x}^{(b)}) \right\}$

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\Big(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{b=1}^{B} \mathbf{L}_{b,b}^* \nabla \mathsf{h}_b\big(\mathbf{L}_{b,b}\boldsymbol{x}_k^{(b)}\big)\Big)$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic BC-FB algorithm: Particular case

Find $\mathbf{x}^\star = (\mathbf{x}_1^\star, \ldots, \mathbf{x}_B^\star) \in \underset{\mathbf{x} \in \mathbb{R}^N}{\operatorname{Argmin}} \left\{ \mathsf{f}(\mathbf{x}) = \sum_{b=1}^{B} \mathsf{h}_b(\mathbf{L}_{b,b}\mathbf{x}^{(b)}) + \sum_{b=1}^{B} \mathsf{g}_b(\mathbf{x}^{(b)}) \right\}$

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

For $k = 0, 1, \ldots$

- Let $\gamma_k \in]0, 2\tau^{-1}[$
- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$
- For $b \in \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\gamma_k g_b}\left(\boldsymbol{x}_k^{(b)} - \gamma_k \sum_{b=1}^{B} \mathbf{L}_{b,b}^* \nabla \mathsf{h}_b\left(\mathbf{L}_{b,b}\boldsymbol{x}_k^{(b)}\right)\right)$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$

Stochastic Block-Coordinate Primal-Dual Algorithm

General minimisation problem

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

where

- $\mathsf{h}\colon \mathbb{R}^N \to \mathbb{R}$ is convex, proper and β-Lipschitz differentiable
- $\mathsf{f} \in \Gamma_0(\mathbb{R}^N)$
- $(\forall d \in \{1,\ldots,D\})\ \mathsf{g}_d \in \Gamma_0(\mathbb{R}^{L_d})$
- $(\forall d \in \{1,\ldots,D\})\ \mathbf{L}_d \in \mathbb{R}^{L_d\times N}$

☛ Use Condat-Vũ primal-dual algorithm

Block-minimisation problem

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

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

- $\mathsf{h}_b \colon \mathbb{R}^{N_b} \to \mathbb{R}$ is convex, proper and β_b-Lipschitz differentiable
- $\mathsf{f}_b \in \Gamma_0(\mathbb{R}^{N_b})$
- $\mathsf{g}_d \in \Gamma_0(\mathbb{R}^{L_d})$
- $\mathbf{L}_{d,b} \in \mathbb{R}^{L_d \times N_b}$

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

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

For $k = 0, 1, \ldots$

- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$ and $\mathbb{D}_k \subset \{1, \ldots, D\}$
- For $b \in \mathbb{B}_k$
  - $$\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\tau_b \mathsf{f}_b}\Big(\boldsymbol{x}_k^{(b)} - \tau_b\big(\nabla \mathsf{h}_b(\boldsymbol{x}_k^{(b)}) + \sum_{d=1}^{D} \mathbf{L}_{d,b}^{*}\boldsymbol{u}_k^{(d)}\big)\Big)$$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$
- For $d \in \mathbb{D}_k$
  - $$\boldsymbol{u}_{k+1}^{(d)} = \operatorname{prox}_{\sigma_d \mathsf{g}_d^{*}}\Big(\boldsymbol{u}_k^{(d)} + \sigma_d \sum_{b=1}^{B} \mathbf{L}_{d,b}(2\boldsymbol{x}_{k+1}^{(b)} - \boldsymbol{x}_k^{(b)})\Big)$$
- For $d \in \{1, \ldots, D\} \setminus \mathbb{D}_k$
  - $\boldsymbol{u}_{k+1}^{(d)} = \boldsymbol{u}_k^{(d)}$

Stochastic BC-PD algorithm

$$\text{Find } \mathbf{x}^\star = (\mathbf{x}_1^\star, \ldots, \mathbf{x}_B^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}} \sum_{b=1}^{B} \mathsf{h}_b(\mathbf{x}^{(b)}) + \mathsf{f}_b(\mathbf{x}^{(b)}) + \sum_{d=1}^{D} \mathsf{g}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\mathbf{x}^{(b)}\Big)$$

For $k = 0, 1, \ldots$

- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$ and $\mathbb{D}_k \subset \{1, \ldots, D\}$
- For $b \in \mathbb{B}_k$
  - $$\boldsymbol{x}_{k+1}^{(b)} = \mathrm{prox}_{\tau_b \mathsf{f}_b}\Big(\boldsymbol{x}_k^{(b)} - \tau_b\big(\nabla \mathsf{h}_b(\boldsymbol{x}_k^{(b)}) + \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \boldsymbol{u}_k^{(d)}\big)\Big)$$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$
- For $d \in \mathbb{D}_k$
  - $$\boldsymbol{u}_{k+1}^{(d)} = \mathrm{prox}_{\sigma_d \mathsf{g}_d^*}\Big(\boldsymbol{u}_k^{(d)} + \sigma_d \sum_{b=1}^{B} \mathbf{L}_{d,b}(2\boldsymbol{x}_{k+1}^{(b)} - \boldsymbol{x}_k^{(b)})\Big)$$
- For $d \in \{1, \ldots, D\} \setminus \mathbb{D}_k$
  - $\boldsymbol{u}_{k+1}^{(d)} = \boldsymbol{u}_k^{(d)}$

Stochastic BC-PD algorithm

Find $\mathbf{x}^\star = (\mathbf{x}_1^\star, \ldots, \mathbf{x}_B^\star) \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \sum_{b=1}^{B} \mathsf{h}_b(\mathbf{x}^{(b)}) + \mathsf{f}_b(\mathbf{x}^{(b)}) + \sum_{d=1}^{D} \mathsf{g}_d\Big(\sum_{b=1}^{B} \mathbf{L}_{d,b}\mathbf{x}^{(b)}\Big)$

For $k = 0, 1, \ldots$

- Select randomly $\mathbb{B}_k \subset \{1, \ldots, B\}$ and $\mathbb{D}_k \subset \{1, \ldots, D\}$
- For $b \in \mathbb{B}_k$
  - $$\boldsymbol{x}_{k+1}^{(b)} = \operatorname{prox}_{\tau_b \mathsf{f}_b}\Big(\boldsymbol{x}_k^{(b)} - \tau_b\big(\nabla \mathsf{h}_b(\boldsymbol{x}_k^{(b)}) + \sum_{d=1}^{D} \mathbf{L}_{d,b}^* \boldsymbol{u}_k^{(d)}\big)\Big)$$
- For $b \in \{1, \ldots, B\} \setminus \mathbb{B}_k$
  - $\boldsymbol{x}_{k+1}^{(b)} = \boldsymbol{x}_k^{(b)}$
- For $d \in \mathbb{D}_k$
  - $$\boldsymbol{u}_{k+1}^{(d)} = \operatorname{prox}_{\sigma_d \mathsf{g}_d^*}\Big(\boldsymbol{u}_k^{(d)} + \sigma_d \sum_{b=1}^{B} \mathbf{L}_{d,b}(2\boldsymbol{x}_{k+1}^{(b)} - \boldsymbol{x}_k^{(b)})\Big)$$
- For $d \in \{1, \ldots, D\} \setminus \mathbb{D}_k$
  - $\boldsymbol{u}_{k+1}^{(d)} = \boldsymbol{u}_k^{(d)}$

Stochastic BC-PD algorithm: Conditions for convergence

Assumptions:

Let $b \in \{1, \ldots, B\}$, and p_b be the probability to select block b.
Let $d \in \{1, \ldots, D\}$, 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)
- $(p_b)_{1 \leqslant b \leqslant B}$ and $(p_d)_{1 \leqslant d \leqslant D}$ are **not** independent

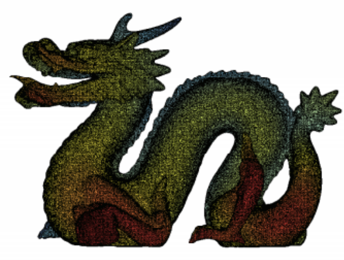
The sequence $(\boldsymbol{x}_k)_{k \in \mathbb{N}}$ converges almost surely to a random variable $\boldsymbol{x}^\star$, solution to the primal problem.

[Pesquet & Repetti, 2015] [Chambolle, Ehrhardt, Richtarik, Schönlieb, 2018]

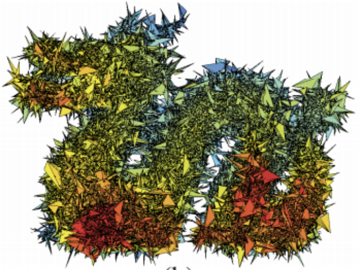
Application to 3D mesh denoising

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

- $\overline{\mathbf{x}} = (\overline{\mathbf{x}}^{(n)})_{1 \leqslant n \leqslant N} \in \mathbb{R}^{N \times 3}$ 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 $\overline{\mathbf{x}}$



(a)



(b)

Application to 3D mesh denoising

Inverse problem: $\mathbf{z} = \overline{\mathbf{x}} + \mathbf{w}$ where

- $\overline{\mathbf{x}} = (\overline{\mathbf{x}}^{(n)})_{1 \leqslant n \leqslant N} \in \mathbb{R}^{N \times 3}$ 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 $\overline{\mathbf{x}}$

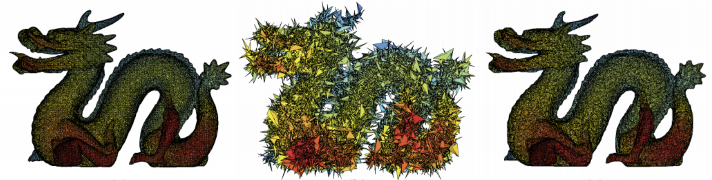
Minimisation problem:

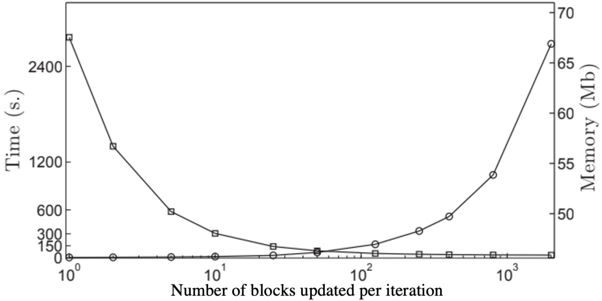
$$\underset{\mathbf{x} \in \mathbb{R}^{N \times 3}}{\text{minimise}} \quad \sum_{n=1}^{N} \mathsf{h}_n(\mathbf{x}^{(n)}) + \mathsf{f}_n(\mathbf{x}^{(n)}) + \sum_{n=1}^{N} \mathsf{g}_n\Big(\sum_{m=1}^{N} \mathbf{L}_{n,m}\mathbf{x}^{(m)}\Big)$$

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

- $\mathsf{h}_n(\mathbf{x}^{(n)}) = \frac{1}{2}\|\mathbf{x}^{(n)} - \mathbf{z}^{(n)}\|^2$
- $\mathsf{f}_n(\mathbf{x}^{(n)} = \iota_{[\underline{\delta},\overline{\delta}]^3}(\mathbf{x}^{(n)})$
- $\mathsf{g}_n(\sum_{m=1}^{N} \mathbf{L}_{n,m}\mathbf{x}^{(m)}) = \eta \sum_{D \in \{X,Y,Z\}} \big\| \big(\mathsf{x}^{(n,D)} - \mathsf{x}^{(i,D)}\big)_{i \in \mathcal{V}_n} \big\|$

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


(a) (b) (c)



Proximal Stochastic Gradient Algorithm

Minimisation problem

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\, \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

where

- $(\forall d \in \{1,\ldots,D\})$ $\mathsf{h}_d \colon \mathbb{R}^{L_d} \to \mathbb{R}$ is convex, proper and β_d-Lipschitz differentiable
- $(\forall d \in \{1,\ldots,D\})$ $\mathbf{L}_d \in \mathbb{R}^{L_d\times N}$
- $\mathsf{g} \in \Gamma_0(\mathbb{R}^N)$

Objective:

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

☛ Stochastic gradient approach

Proximal stochastic gradient algorithm in a nutshell

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}}\, \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$

- Select randomly $\mathbb{D}_k \subset \{1, \ldots, D\}$
- Build $\boldsymbol{u}^{(k)} \approx \nabla(\mathsf{h}\circ\mathbf{L})(\mathbf{x}^{(k)})$ only with functions associated with $\mathbb{D}_k$
- Update $\boldsymbol{x}_{k+1}$ with a FB step using $\boldsymbol{u}_k$

Proximal stochastic gradient algorithm in a nutshell

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \, \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$
- Select randomly $\mathbb{D}_k \subset \{1, \ldots, D\}$
- Build $\boldsymbol{u}^{(k)} \approx \nabla(\mathsf{h} \circ \mathbf{L})(\mathbf{x}^{(k)})$ only with functions associated with $\mathbb{D}_k$
- Update $\boldsymbol{x}_{k+1}$ with a FB step using $\boldsymbol{u}_k$

Proximal stochastic gradient algorithm

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \dfrac{1}{D}\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$

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

$\boldsymbol{x}_{k+1} = \mathrm{prox}_{\gamma_k \mathsf{g}}\left(\boldsymbol{x}_k - \gamma_k \boldsymbol{u}_k\right)$

Proximal stochastic gradient algorithm

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \dfrac{1}{D}\displaystyle\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$
- Let $\boldsymbol{u}_k \approx \nabla(\mathsf{h}\circ\mathbf{L})(\mathbf{x}_k)$
- $\boldsymbol{x}_{k+1} = \mathrm{prox}_{\gamma_k \mathsf{g}}\left(\boldsymbol{x}_k - \gamma_k \boldsymbol{u}_k\right)$

Questions:

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

Proximal stochastic gradient: Approximation strategies

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\mathrm{Argmin}}\ \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$
- Let $\boldsymbol{u}_k \approx \nabla(\mathsf{h}\circ\mathbf{L})(\mathbf{x}_k)$
- $\boldsymbol{x}_{k+1} = \mathrm{prox}_{\gamma_k \mathsf{g}}\left(\boldsymbol{x}_k - \gamma_k \boldsymbol{u}_k\right)$

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

  Set $\boldsymbol{u}_k = \dfrac{1}{\sharp\,\mathbb{D}_k}\displaystyle\sum_{d\in\mathbb{D}_k}\nabla(\mathsf{h}_d\circ\mathbf{L}_d)(\boldsymbol{x}_k)$

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

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

Proximal stochastic gradient: Approximation strategies

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

For $k = 0, 1, \ldots$
- Let $\boldsymbol{u}_k \approx \nabla(\mathsf{h}\circ\mathbf{L})(\mathbf{x}_k)$
- $\boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k \mathsf{g}}\left(\boldsymbol{x}_k - \gamma_k \boldsymbol{u}_k\right)$

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

Set $\boldsymbol{u}_k = \boldsymbol{u}^{(k-1)} + \dfrac{1}{D}\displaystyle\sum_{d\in\mathbb{D}_k}\Big(\nabla(\mathsf{h}_d\circ\mathbf{L}_d)(\mathbf{x}_k) - \nabla(\mathsf{h}_d\circ\mathbf{L}_d)(\mathbf{x}^{(k-1)})\Big)$

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

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

Proximal stochastic gradient: Convergence

Find $\mathbf{x}^\star \in \underset{\mathbf{x}\in\mathbb{R}^N}{\operatorname{Argmin}} \, \mathsf{f}(\mathbf{x}) = \Big\{\mathsf{h}(\mathbf{L}\mathbf{x}) + \mathsf{g}(\mathbf{x})\Big\}$ with $\mathsf{h}(\mathbf{L}\mathbf{x}) = \frac{1}{D}\sum_{d=1}^{D} \mathsf{h}_d(\mathbf{L}_d\mathbf{x})$

Let $\theta \in]1/2, 1]$
For $k = 0, 1, \ldots$

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

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

$$\boldsymbol{u}_k = \frac{1}{\sharp\,\mathbb{D}_k} \sum_{d\in\mathbb{D}_k} \nabla(\mathsf{h}_d \circ \mathbf{L}_d)(\boldsymbol{x}_k)$$

$$\boldsymbol{x}_{k+1} = \operatorname{prox}_{\gamma_k \mathsf{g}}\left(\boldsymbol{x}_k - \gamma_k \boldsymbol{u}_k\right)$$

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

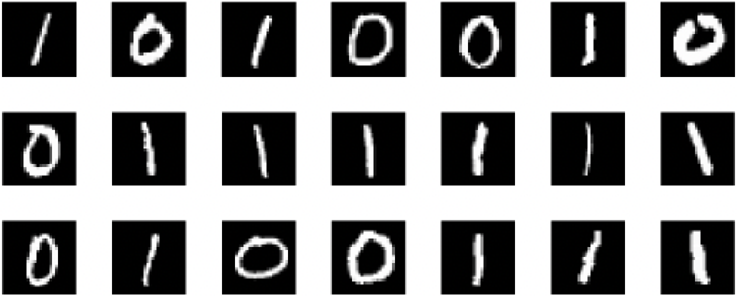
[Combettes & Pesquet, 2015][Rosasco, Villa, Vũ, 2020]

Example: binary SVM

Binary classification:

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

$$\mathcal{S} = \left\{ (\mathbf{x}_\ell, z_\ell)_{1 \leqslant \ell \leqslant L} \;\middle|\; (\forall \ell \in \{1, \ldots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N,\ z_\ell \in \{-1, +1\} \right\}$$



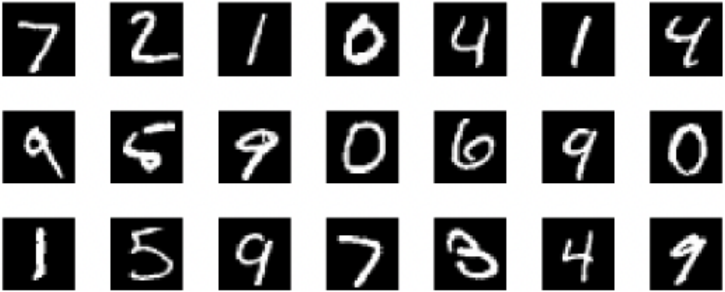
If digit on $\mathbf{x}_\ell$ is 0, then $z_\ell = -1$, otherwise $z_\ell = 1$

Example: binary SVM

Binary classification:

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

$$\mathcal{S} = \left\{(\mathbf{x}_\ell, \mathsf{z}_\ell)_{1 \leqslant \ell \leqslant L} \;\middle|\; (\forall \ell \in \{1, \ldots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N,\ \mathsf{z}_\ell \in \{-1, +1\}\right\}$$



If digit on $\mathbf{x}_\ell$ is 0, then $\mathsf{z}_\ell = -1$, otherwise $\mathsf{z}_\ell = 1$

Example: binary SVM

Binary classification:

- Pair an input $\mathbf{x} \in \mathbb{R}^N$ to a binary output $\mathsf{z} \in \{-1, +1\}$: $\mathsf{z} = \mathrm{sign}(\mathbf{x}^\top \mathbf{w})$
- Classifier trained on a training set of pairs
$$\mathcal{S} = \big\{ (\mathbf{x}_\ell, \mathsf{z}_\ell)_{1 \leqslant \ell \leqslant L} \;\big|\; (\forall \ell \in \{1, \ldots, L\} \quad \mathbf{x}_\ell \in \mathbb{R}^N,\ \mathsf{z}_\ell \in \{-1, +1\} \big\}$$

Minimisation problem:

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

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

Example: binary SVM

Result comparison: