MAJORIZATION-MINIMIZATION SUBSPACE ALGORITHMS FOR LARGE SCALE DATA PROCESSING

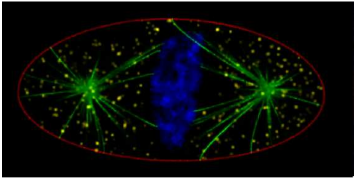
Emilie Chouzenoux, Inria Saclay, OPIS, France

(joint work with *Mathieu Chalvidal, CNRS, ANITI, Toulouse, France)*

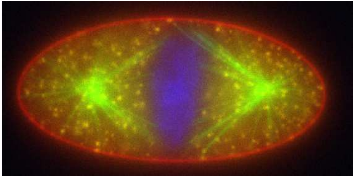
Séminaire Français d'Optimisation

14 October 2020

Inverse problems and large scale optimization

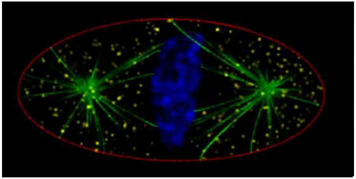


Original image



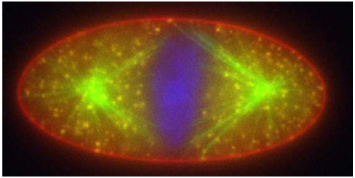
Degraded image

Inverse problems and large scale optimization



Original image

$\overline{\boldsymbol{x}} \in \mathbb{R}^N$



Degraded image

$\boldsymbol{y} = \mathcal{D}(\boldsymbol{H}\overline{\boldsymbol{x}}) \in \mathbb{R}^M$

- $\boldsymbol{H} \in \mathbb{R}^{M \times N}$: matrix associated with the degradation operator.
- $\mathcal{D} \colon \mathbb{R}^M \to \mathbb{R}^M$: noise degradation.

How to find a good estimate of $\overline{\boldsymbol{x}}$ from the observations $\boldsymbol{y}$ and the model $\boldsymbol{H}$ in the context of large scale processing?

Inverse problems and large scale optimization

Variational approach:

An image estimate $\hat{\boldsymbol{x}} \in \mathbb{R}^N$ is generated by minimizing

$$(\forall \boldsymbol{x} \in \mathbb{R}^N) \quad F(\boldsymbol{x}) = \sum_{s=1}^{S} f_s(\boldsymbol{L}_s \boldsymbol{x})$$

with $f_s : \mathbb{R}^{P_s} \to \mathbb{R}$, $\boldsymbol{L}_s \in \mathbb{R}^{P_s \times N}$, $P_s > 0$.

In the context of maximum a posteriori estimation :

- $\boldsymbol{L}_1$: Degradation operator, i.e. $\boldsymbol{H}$;
- f_1: Data fidelity (e.g. least squares);
- $(f_s)_{2 \leqslant s \leqslant S}$: Regularization functions on some linear transforms $(\boldsymbol{L}_s)_{2 \leqslant s \leqslant S}$ of the sought solution.

$\rightarrow$ Often no closed form expression or solution expensive to compute (especially in large scale context).

▶ Need for an efficient iterative minimization strategy !

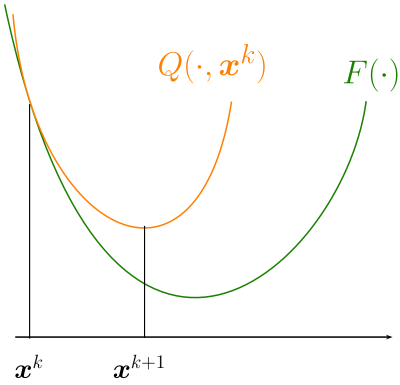
Outline

* MAJORIZE-MINIMIZE MEMORY GRADIENT ALGORITHM
  - Majorize-Minimize principle
  - Subspace acceleration
  - Convergence theorem
* BLOCK DISTRIBUTED 3MG ALGORITHM
  - Block alternating 3MG
  - Block distributed 3MG
  - Convergence theorem
  - Practical implementation
* APPLICATION TO 3D DECONVOLUTION
  - Variational approach
  - Distributed implementation
  - Numerical results

Majorize-Minimize Memory Gradient algorithm

Majorize-Minimize principle

1. **Find a majorant for F** **M**ajorization step



Majorize-Minimize principle

1. Find a majorant for F Majorization step

Quadratic tangent majorant of F at $\boldsymbol{x}_k$

$$(\forall \boldsymbol{x} \in \mathbb{R}^N) \quad Q(\boldsymbol{x}, \boldsymbol{x}^k) = F(\boldsymbol{x}^k) + \nabla F(\boldsymbol{x}^k)^\top (\boldsymbol{x} - \boldsymbol{x}^k) + \frac{1}{2}(\boldsymbol{x} - \boldsymbol{x}^k)^\top \boldsymbol{A}(\boldsymbol{x}^k)(\boldsymbol{x} - \boldsymbol{x}^k)$$

where, for every $\boldsymbol{x} \in \mathbb{R}^N$, $\boldsymbol{A}(\boldsymbol{x}) \in \mathbb{R}^{N \times N}$ is a symmetric definite positive matrix such that

$$(\forall \boldsymbol{x} \in \mathbb{R}^N) \quad Q(\boldsymbol{x}, \boldsymbol{x}^k) \geqslant F(\boldsymbol{x}).$$

$\ast$ Several methods available to construct matrix $\boldsymbol{A}(\boldsymbol{x})$ in the context of inverse problems in image processing.

Subspace acceleration

2. Minimize in a subspace

Minimization step

$$(\forall k \in \mathbb{N}^*) \qquad \boldsymbol{x}^{k+1} \in \underset{\boldsymbol{x} \in \operatorname{ran} \boldsymbol{D}^k}{\operatorname{Argmin}} \; Q(\boldsymbol{x}, \boldsymbol{x}^k),$$

with $\boldsymbol{D}^k \in \mathbb{R}^{N \times M_k}$.

- $\operatorname{ran} \boldsymbol{D}^k = \mathbb{R}^N \Rightarrow$ half-quadratic algorithm.
- M_k small $\Rightarrow$ low-complexity per iteration.

Memory-Gradient subspace:

$$\boldsymbol{D}^k = \begin{cases} [-\nabla F(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x}^{k-1}] & \text{if } k \geqslant 1 \\ -\nabla F(\boldsymbol{x}^0) & \text{if } k = 0 \end{cases}$$

3MG algorithm

(similar ideas in NLCG, L-BFGS, TWIST, FISTA, ...)

3MG algorithm

Initialize $\boldsymbol{x}^0 \in \mathbb{R}^N$
For $k = 0, 1, 2, \dots$
- Compute $\nabla F(\boldsymbol{x}^k)$
- If $k = 0$
  - $\boldsymbol{D}^k = -\nabla F(\boldsymbol{x}^0)$
- Else
  - $\boldsymbol{D}^k = [-\nabla F(\boldsymbol{x}^k), \boldsymbol{x}^k - \boldsymbol{x}^{k-1}]$
- $\boldsymbol{B}^k = ((\boldsymbol{D}^k)^\top \boldsymbol{A}(\boldsymbol{x}^k)\boldsymbol{D}^k)^\dagger$
- $\boldsymbol{d}^k = -\boldsymbol{D}^k \boldsymbol{B}^k (\boldsymbol{D}^k)^\top \nabla F(\boldsymbol{x}^k)$
- $\boldsymbol{x}^{k+1} = \boldsymbol{x}^k + \boldsymbol{d}^k$

✓ Low computational cost since $\boldsymbol{B}^k$ is of dimension $M_k \times M_k$, with $M_k \in \{1, 2\}$.

✓ Complexity reductions possible by taking into account the structures of F and $\boldsymbol{D}^k$.

Convergence theorem [Chouzenoux *et al.*, 2011] [Chouzenoux *et al.*, 2013]

Let assume that:

1. $F : \mathbb{R}^N \to \mathbb{R}$ is a coercive, differentiable function.
2. There exists $(\underline{\nu}, \overline{\nu}) \in]0, +\infty[^2$ such that $(\forall k \in \mathbb{N})$
   $\underline{\nu}\,\mathbf{Id} \preceq \boldsymbol{A}(\boldsymbol{x}^k) \preceq \overline{\nu}\,\mathbf{Id}$,

Then, the following hold:

- $\|\nabla F(\boldsymbol{x}^k)\| \to 0$ and $F(\boldsymbol{x}^k) \searrow F(\widehat{\boldsymbol{x}})$ where $\widehat{\boldsymbol{x}}$ is a critical point of F.
- If F is convex, any sequential cluster point of $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ is a minimizer of F.
- If F is strongly convex, then $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ converges to the unique (global) minimizer $\widehat{\boldsymbol{x}}$ of F
- If F is semi-algebraic, then the sequence $\left(\boldsymbol{x}^k\right)_{k\in\mathbb{N}}$ converges to a critical point of F.

3MG in practical situations

3MG algorithm outperforms state-of-the arts optimization algorithms in many image processing applications.

Problem: Computational issues with very large-size problems.

Main reasons:

- ✗ High computational time for calculating the gradient direction $\nabla F(\boldsymbol{x}^k)$ and the matrix $\boldsymbol{B}^k$;
- ✗ High storage cost for $\nabla F(\boldsymbol{x}^k)$, $\boldsymbol{D}^k$ and $\boldsymbol{x}^k$.

$\downarrow$

Block distributed approach

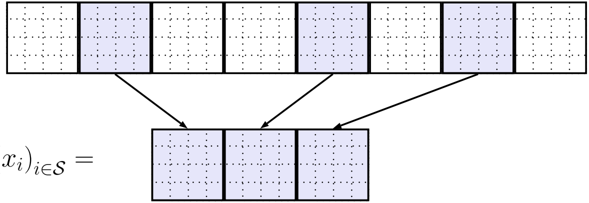
Block distributed 3MG algorithm

Block minimization strategy

At each iteration, a subset (i.e. a block) of the entries of the vector $\boldsymbol{x}$, indexed by $\mathcal{S} \subset [\![1, N]\!]$ with cardinality $|\mathcal{S}|$, are updated, while the others remain fixed.

$\boldsymbol{x} =$

$\boldsymbol{x}_{(\mathcal{S})} = (x_i)_{i \in \mathcal{S}} =$



Advantages:

- ✓ Control of the memory thanks to the block alternating strategy;
- ✓ Scalability using asynchronous implementation (see in few slides).

Block alternating 3MG

1. **Select a block**: Choose a non empty $\mathcal{S}^k \subset [\![1, N]\!]$.

Block alternating 3MG

1. Select a block: Choose a non empty $\mathcal{S}^k \subset [\![1, N]\!]$.

2. Find a majorant within this block:

Set $\boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k) = \left([\boldsymbol{A}(\boldsymbol{x}^k)]_{p,p}\right)_{p \in \mathcal{S}^k}$. The restriction of F to $\mathcal{S}^k$ is majorized at $\boldsymbol{x}^k$ by

$$(\forall \boldsymbol{v} \in \mathbb{R}^{|\mathbb{S}^k|}) \quad Q_{(\mathcal{S}^k)}(\boldsymbol{v}, \boldsymbol{x}^k) = F(\boldsymbol{x}^k) + \nabla F_{(\mathcal{S}^k)}(\boldsymbol{x}^k)^\top (\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}) + \frac{1}{2}(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)})^\top \boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k)(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}).$$

Block alternating 3MG

1. Select a block: Choose a non empty $\mathcal{S}^k \subset [\![1, N]\!]$.

2. Find a majorant within this block:

Set $\boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k) = \big([\boldsymbol{A}(\boldsymbol{x}^k)]_{p,p}\big)_{p\in\mathcal{S}^k}$. The restriction of F to $\mathcal{S}^k$ is majorized at $\boldsymbol{x}^k$ by

$$(\forall \boldsymbol{v} \in \mathbb{R}^{|\mathbb{S}^k|}) \quad Q_{(\mathcal{S}^k)}(\boldsymbol{v}, \boldsymbol{x}^k) = F(\boldsymbol{x}^k) + \nabla F_{(\mathcal{S}^k)}(\boldsymbol{x}^k)^\top (\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}) + \frac{1}{2}(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)})^\top \boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k)(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}).$$

3. Minimize within the memory gradient subspace

$$\boldsymbol{x}^{k+1}_{(\mathcal{S}^k)} = \underset{\boldsymbol{v}\in \operatorname{ran} \boldsymbol{D}^k_{(\mathcal{S}^k)}}{\operatorname{Argmin}} \quad Q_{(\mathcal{S}^k)}(\boldsymbol{v}, \boldsymbol{x}^k)$$

where

$$(\forall i \in \mathcal{S}^k) \quad (\boldsymbol{D}^k)_i = \begin{cases} -\nabla F_i(\boldsymbol{x}^k) & \text{if} \quad i \notin \bigcup_{\ell=0}^{k-1} \mathcal{S}_\ell, \\ \big[-\nabla F_i(\boldsymbol{x}^k) \big| x_i^k - x_i^{k-1}\big] & \text{otherwise.} \end{cases}$$

Block alternating 3MG

1. Select a block: Choose a non empty $\mathcal{S}^k \subset [\![1, N]\!]$.

2. Find a majorant within this block:

Set $\boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k) = \big([\boldsymbol{A}(\boldsymbol{x}^k)]_{p,p}\big)_{p\in\mathcal{S}^k}$. The restriction of F to $\mathcal{S}^k$ is majorized at $\boldsymbol{x}^k$ by

$$
\begin{aligned}
(\forall \boldsymbol{v} \in \mathbb{R}^{|\mathbb{S}^k|}) \quad Q_{(\mathcal{S}^k)}(\boldsymbol{v}, \boldsymbol{x}^k) = F(\boldsymbol{x}^k) &+ \nabla F_{(\mathcal{S}^k)}(\boldsymbol{x}^k)^\top (\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}) \\
&+ \frac{1}{2}(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)})^\top \boldsymbol{A}_{(\mathcal{S}^k)}(\boldsymbol{x}^k)(\boldsymbol{v} - \boldsymbol{x}^k_{(\mathcal{S}^k)}).
\end{aligned}
$$

3. Minimize within the memory gradient subspace

$$
\boldsymbol{x}^{k+1}_{(\mathcal{S}^k)} = \underset{\boldsymbol{v}\in\operatorname{ran} \boldsymbol{D}^k_{(\mathcal{S}^k)}}{\operatorname{Argmin}} \quad Q_{(\mathcal{S}^k)}(\boldsymbol{v}, \boldsymbol{x}^k)
$$

where

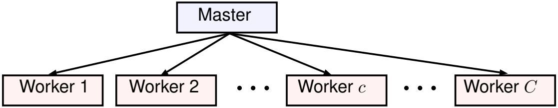
$$
(\forall i \in \mathcal{S}^k) \quad (\boldsymbol{D}^k)_i = \begin{cases} -\nabla F_i(\boldsymbol{x}^k) & \text{if} \quad i \notin \bigcup_{\ell=0}^{k-1} \mathcal{S}_\ell, \\ \big[-\nabla F_i(\boldsymbol{x}^k) \big| x_i^k - x_i^{k-1}\big] & \text{otherwise.} \end{cases}
$$

✗ Block alternating MM algorithms may exhibit **slow convergence**

✗ Parallel versions require **specific form of majorant** functions (e.g., block diagonal $\boldsymbol{A}(\boldsymbol{x})$, see [Cadoni *et al.*, 2016]).

Block distributed 3MG

Distributed architecture:



Principle:

* After an initialization phase, the **master** iterates, for $k = 1, 2, \ldots$, over the following tasks:
  - **Receive** update from a worker c, acting on block $\mathcal{S}_c^k$
  - **Update** block $\mathcal{S}_c^k$ of $\boldsymbol{x}^k$, to create $\boldsymbol{x}^{k+1}$
  - **Send** to worker c a new block $\mathcal{S}_c^{k+1}$ to process.
* At each request from the **master**, the **worker** applies one iteration of 3MG on the required block and sends back its result.

BD3MG (Master)

Initialization :

Set $k = 0, \quad \boldsymbol{x}^0 \in \mathbb{R}^N$.

For all $c \in [\![1, C]\!]$, set $\mathcal{S}_c^0 \subset [\![1, N]\!]$ s.t. $\bigcap_{c \in [\![1,C]\!]} \mathcal{S}_c^0 = \emptyset$,

and send $(\boldsymbol{x}^0, \mathcal{S}_c^0, 0_{|\mathcal{S}_c^0|})$ to worker c.

Define $\mathbb{S}_0 = \bigcup_{c \in [\![1,C]\!]} \mathcal{S}_c^0$.

While a stopping criterion is not met:

(0) Wait for any worker to send an update

(1) **Receive** $(\boldsymbol{d}_{(\mathcal{S}_c^k)})$ from a worker c

(2) Update $\begin{cases} \boldsymbol{x}^{k+1}_{(\mathcal{S}_c^k)} = \boldsymbol{x}^k_{(\mathcal{S}_c^k)} + \boldsymbol{d}_{(\mathcal{S}_c^k)} \\ \boldsymbol{x}^{k+1}_{\overline{(\mathcal{S}_c^k)}} = \boldsymbol{x}^k_{\overline{(\mathcal{S}_c^k)}} \end{cases}$

(3) Choose $\mathcal{S}_c^{k+1} \subset [\![1, N]\!] / (\mathbb{S}_k / \mathcal{S}_c^k)$

For every $c' \in [\![1, C]\!] / \{c\}$, set $\mathcal{S}_{c'}^{k+1} = \mathcal{S}_{c'}^k$.

Define $\mathbb{S}_{k+1} = (\mathbb{S}_k / \mathcal{S}_c^k) \cup \mathcal{S}_c^{k+1}$

(4) **Send** $(\boldsymbol{x}^{k+1}, \mathcal{S}_c^{k+1}, (\boldsymbol{x}^{k+1} - \boldsymbol{x}^k)_{(\mathcal{S}_c^{k+1})})$ to worker c

(5) Increment $k = k + 1$

BD3MG (Worker)

While the Master stopping criterion is not met:

(1) **Receive** $(\boldsymbol{x}, \mathcal{S}, \boldsymbol{d}_{(\mathcal{S})})$ from Master

(2) Set $\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}) = [-\nabla_{(\mathcal{S})} F(\boldsymbol{x}) \mid \boldsymbol{d}_{(\mathcal{S})}]$

(3) Compute $\boldsymbol{A}_{(\mathcal{S})}(\boldsymbol{x})$ and $\nabla_{(\mathcal{S})} F(\boldsymbol{x})$

(4) $\boldsymbol{B}_{(\mathcal{S})}(\boldsymbol{x}) = (\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x})^\top \boldsymbol{A}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}))^\dagger$

(6) $\boldsymbol{d}'_{(\mathcal{S})} = -\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{B}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x})^\top \nabla_{(\mathcal{S})} F(\boldsymbol{x})$

(7) **Send** $(\boldsymbol{d}'_{(\mathcal{S})})$ to the Master

BD3MG (Worker)

While the Master stopping criterion is not met:

(1) **Receive** $(\boldsymbol{x}, \mathcal{S}, \boldsymbol{d}_{(\mathcal{S})})$ from Master

(2) Set $\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}) = [-\nabla_{(\mathcal{S})} F(\boldsymbol{x}) \mid \boldsymbol{d}_{(\mathcal{S})}]$

(3) Compute $\boldsymbol{A}_{(\mathcal{S})}(\boldsymbol{x})$ and $\nabla_{(\mathcal{S})} F(\boldsymbol{x})$

(4) $\boldsymbol{B}_{(\mathcal{S})}(\boldsymbol{x}) = (\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x})^\top \boldsymbol{A}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}))^\dagger$

(6) $\boldsymbol{d}'_{(\mathcal{S})} = -\boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{B}_{(\mathcal{S})}(\boldsymbol{x}) \boldsymbol{D}_{(\mathcal{S})}(\boldsymbol{x})^\top \nabla_{(\mathcal{S})} F(\boldsymbol{x})$

(7) **Send** $(\boldsymbol{d}'_{(\mathcal{S})})$ to the Master

- By construction, there is no overlap in the coordinates updated by the workers at a given time, i.e. $(\forall k \in \mathbb{N}) \quad \bigcap_{c \in [\![1,C]\!]} \mathcal{S}_c^k = \emptyset$.
- Assuming no delay (i.e. the worker returns immediately his feedback to the master), BD3MG becomes equivalent to the aforementioned block alternating 3MG method.

Modeling the latency effects

Asynchronous algorithm: No locking condition between workers, thus latency may appear in local variables, that must be modeled for analyzing the convergence of the algorithm.

- Each local entry x_i used by a worker to perform its update at time k belongs to a past element $x_i^{k'_i}$ of the sequence $\{\boldsymbol{x}^k\}_{k\in\mathbb{N}}$ with $k'_i = \max\{k' \in [\![0,k]\!] \mid i \in \mathbb{S}_{k'}\}$.
- We define $\delta_{k,n} = k - k'_n \in [\![0,k]\!]$, the delay at a coordinate $n \in [\![1,N]\!]$ and $\boldsymbol{\delta}_k = (\delta_{k,n})_{n\in[\![1,N]\!]}$ the complete vector of delays.
- The majorizing property leads, for every $k \in \mathbb{N}$, for every $c \in [\![1,C]\!]$,

$$F(\boldsymbol{x}^{k+1}) \leqslant Q_{(\mathcal{S}_c^k)}(\boldsymbol{x}^{k+1}_{(\mathcal{S}_c^k)}, \boldsymbol{x}^{k,\boldsymbol{\delta}_k}) \leqslant Q_{(\mathcal{S}_c^k)}(\boldsymbol{x}^{k,\boldsymbol{\delta}_k}_{(\mathcal{S}_c^k)}, \boldsymbol{x}^{k,\boldsymbol{\delta}_k}) = F(\boldsymbol{x}^{k,\boldsymbol{\delta}_k}).$$

with $\boldsymbol{x}^{k,\boldsymbol{\delta}_k} = (x_n^{k-\delta_{k,n}})_{n\in[\![1,N]\!]}$.

Convergence theorem [Chalvidal *et al.*, 2020]

Let assume that:

1. $F : \mathbb{R}^N \to \mathbb{R}$ is L-Lipschitz differentiable, bounded from below and semi-algebraic.
2. (bounded delay) There exists $\tau \in \mathbb{N}$ such that for every $k \in \mathbb{N}$, we have $[\![1, N]\!] \subset \bigcup_{i=k-\tau}^{k} \mathbb{S}_i$, with the convention $\forall\, l \in \mathbb{N}^*,\ \mathbb{S}_{-l} = \emptyset$.
3. (bounded majorant error) There exists $(\underline{\nu}, \overline{\nu}) > 0$ such that, for every $k \in \mathbb{N}$, for every $c \in [\![1, C]\!]$, $(L\sqrt{\tau} + \underline{\nu})\, \mathbf{Id} \preceq \boldsymbol{\Gamma}_c^k \preceq \overline{\nu}\, \mathbf{Id}$, with, for every $k \in \mathbb{N}$, $c \in [\![1, C]\!]$, $\boldsymbol{\Gamma}_c^k = \boldsymbol{A}_{(\mathcal{S}_c^k)}(\boldsymbol{x}^{k,\delta_k}) - \frac{1}{2}\boldsymbol{A}_{(\mathcal{S}_c^k)}(\boldsymbol{x}^k)$.

Then, the sequence $(\boldsymbol{x}^k)_{k\in\mathbb{N}}$ built by BD3MG algorithm is of finite length and converges to a critical point of F.

Control of information sharing

In practice, it is usually not necessary to send the full vector $\boldsymbol{x}$ to the core working on the update of a given block $\mathcal{S}$.

If $F(\boldsymbol{x}) = \sum_{s=1}^{S} f_s(\boldsymbol{L}_s \boldsymbol{x})$, then $\boldsymbol{A}(\boldsymbol{x}) = \sum_{s=1}^{S} \boldsymbol{L}_s{}^\top \mathrm{Diag}\left\{\omega_s(\boldsymbol{L}_s \boldsymbol{x})\right\} \boldsymbol{L}_s$, with $\omega_s : \mathbb{R}^{P_s} \to \mathbb{R}^{P_s}$ a smooth positive valued mapping.

- The update of block $\mathcal{S}$ only requires sharing x_i with i in

$$\mathcal{N}^{(\mathcal{S})} = \bigcup_{s=1}^{S} \left\{n \in \{1,\dots,N\} | (\exists p \in \mathcal{P}_s^{(\mathcal{S})})\ [\boldsymbol{L}_s]_{p,n} \neq 0\right\},$$

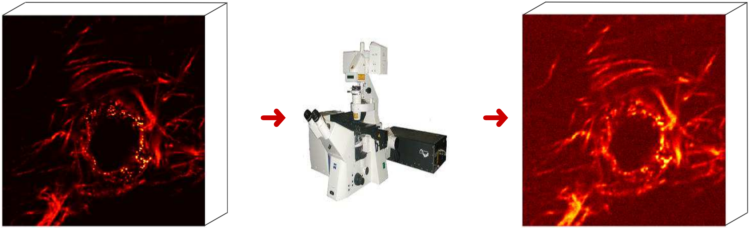
where $\mathcal{P}_s^{(\mathcal{S})} = \left\{p \in \{1,\dots,P_s\}\,|(\exists j \in \mathcal{S})\ [\boldsymbol{L}_s]_{p,j} \neq 0\right\}$

$*$ For sparse $(\boldsymbol{L}_s)_{1 \leqslant s \leqslant S}$, the cardinality of $\mathcal{N}^{(\mathcal{S})}$ can be very small with respect to N.

Examples: discrete gradient operators, convolution operators with small kernels.

Application to 3D image deconvolution

Problem statement



Original 3D image $\overline{\boldsymbol{x}} \in \mathbb{R}^N$

Degradations $\boldsymbol{H} \in \mathbb{R}^{N \times N}$, $\boldsymbol{b} \in \mathbb{R}^N$

Measured 3D image $\boldsymbol{y} = \boldsymbol{H}\overline{\boldsymbol{x}} + \boldsymbol{b}$

- $\boldsymbol{H}$: 3D convolution operator representing depth-variant 3D Gaussian blurs. For each depth $z \in \{1, \dots, N_{\mathrm{Z}}\}$, different variance and rotation parameters.
- $\boldsymbol{b}$: additive Gaussian i.i.d. zero-mean noise.

Variational approach

Objective function

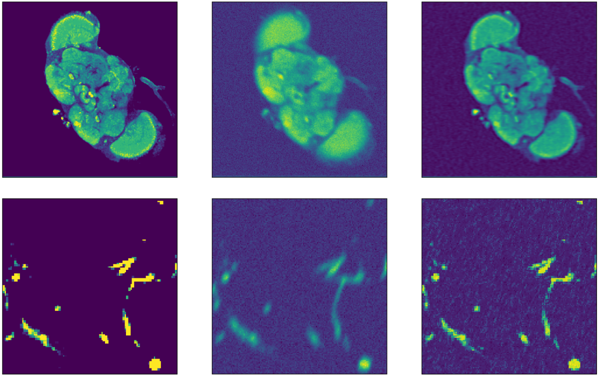
$$(\forall x \in \mathbb{R}^N) \quad F(\boldsymbol{x}) = \frac{1}{2}\|\boldsymbol{H}\boldsymbol{x} - \boldsymbol{y}\|^2 + R(\boldsymbol{x})$$

⇝ Hybrid penalization term $R = R_1 + R_2 + R_3$:

- $R_1(\boldsymbol{x}) = \eta \sum_{n=1}^{N} d^2_{[x_{\min}, x_{\max}]}(x_n)$
- $R_2(\boldsymbol{x}) = \lambda \sum_{n=1}^{N} \sqrt{([\boldsymbol{V}^{\mathsf{X}}\boldsymbol{x}]_n)^2 + ([\boldsymbol{V}^{\mathsf{Y}}\boldsymbol{x}]_n)^2 + \delta^2}$
- $R_3(\boldsymbol{x}) = \kappa \sum_{n=1}^{N} ([\boldsymbol{V}^{\mathsf{Z}}\boldsymbol{x}]_n)^2$

- $(\eta, \lambda, \delta, \kappa) \in (0, +\infty)^4$: regularization parameters;
- $[x_{\min}, x_{\max}]$: range of pixel intensity values; d_C: distance to set C;
- $\boldsymbol{V}^{\mathsf{X}}$,$\boldsymbol{V}^{\mathsf{Y}}$,$\boldsymbol{V}^{\mathsf{Z}} \in \mathbb{R}^{N \times N}$: discrete gradients along X,Y and Z.

Restoration results



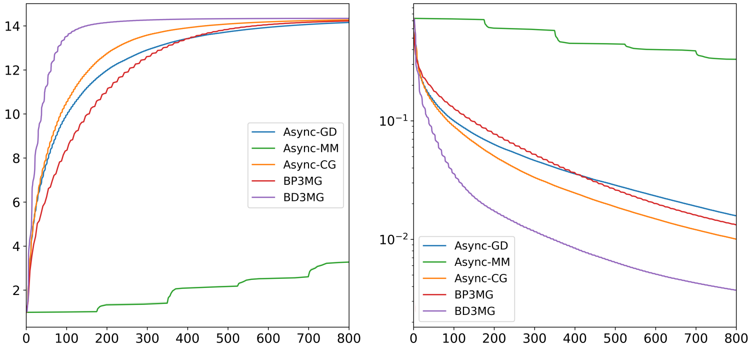
Comparison between original (**left**), degraded (**middle**) and restored (**right**) slices ($z = 10$) of `FlyBrain` ($N = 256 \times 256 \times 24$ degraded by $11 \times 11 \times 21$ blurs) and `Aneurysm` ($N = 155 \times 154 \times 79$ degraded by $5 \times 5 \times 11$ blurs).

Ablation study

- Comparison of BD3MG with four parallel/distributed optimization algorithms, obtained by removing some/all acceleration features of our algorithm:

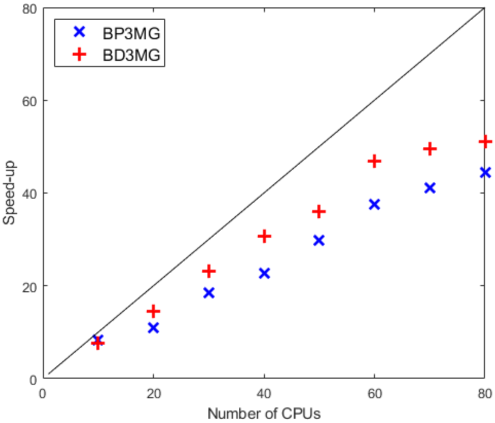
Name	Asynchrony	Memory	MM scaling
Async-GD [Niu *et al.*, 2011]	✓	✗	✗
Async-CG	✓	✓	✗
Async-MM	✓	✗	✓
BP3MG [Cadoni *et al.*, 2016]	✗	✓	✓
BD3MG	✓	✓	✓

Ablation study



Evolution of SNR in dB (left) and relative distance to solution $\|\boldsymbol{x}^k - \boldsymbol{x}^*\|/\|\boldsymbol{x}^*\|$ (right) along time (in seconds) for `FlyBrain` restoration. Intel Xeon(R) W-2135 CPU with 12 cores clocked at 3.70GHz.

Linear speedup



Speed-up ratio for BD3MG (blue) and BP3MG (red), with respect to the number of active cores for the restoration of `Aneurysm`. Intel Xeon CPU 6148 with up to 80 physical cores at 2.4 GHz (Skylake) and 1.5 Tio of RAM.

Conclusion

The *Block Distributed Majorize-Minimize Memory Gradient (BD3MG) Algorithm* handles non convex smooth optimization problems of very large dimension.

- ✓ Reduced complexity / memory requirement.
- ✓ Convergence assessed under mild assumptions.
- ✓ High efficiency in the context of 3D image restoration.

https://github.com/mathieuchal/BD3MG

Some references

M. Chalvidal and E. Chouzenoux
Block Distributed 3MG Algorithm and its Application to 3D Image Restoration. In Proc. IEEE Conf. Image Process. (ICIP 2020).

E. Chouzenoux, J. Idier, S. Moussaoui
A Majorize-Minimize strategy for subspace optimization applied to image restoration
in *IEEE Trans. Image Process.*, vol.20, no.18, pp. 1517-1528, 2011.

E. Chouzenoux, A. Jezierska, J.-C. Pesquet, H. Talbot
A Majorize-Minimize Subspace Approach for ℓ_2-ℓ_0 Image Regularization
in *SIAM J. Imag. Sci.*, vol.6, no.1, pp. 563-591,2013.

S. Cadoni, E. Chouzenoux, J.-C. Pesquet, C. Chaux
A Block Parallel Majorize-Minimize Memory Gradient Algorithm
in Proc. IEEE Conf. Image Process. (ICIP 2016), pp. 3194–3198, Phoenix, AZ, 25-28 Sep 2016.