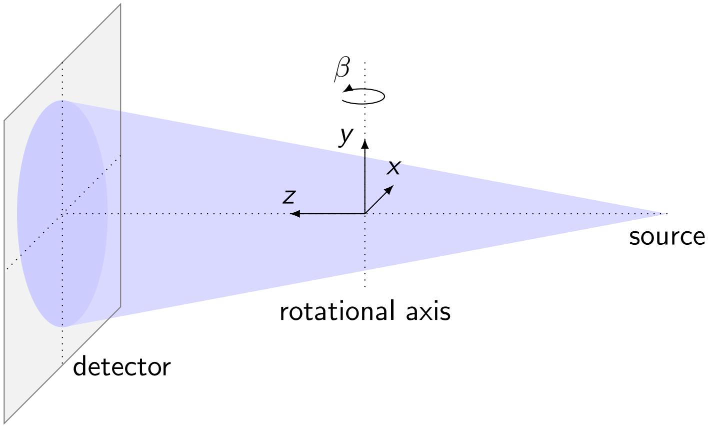
A primal-dual algorithm with Krylov subspace projection for computed tomography

AIP 2025

Patricio Guerrero
joint work with Mirjeta Pasha (Virginia Tech) and Wim Dewulf

Manufacturing Metrology Group
KU Leuven

Rio de Janeiro, July 2025

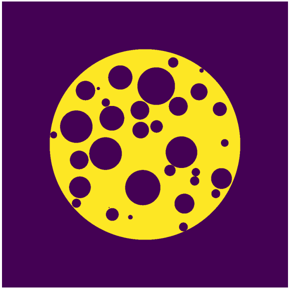


- Tomographic inverse problem:

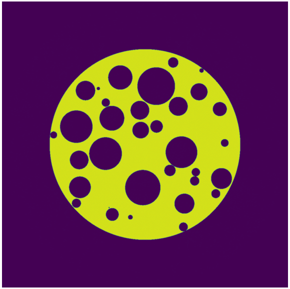
given data b and operator A, find u such that
$$\mathrm{A}u = b + \text{noise}$$

- First goal: Study *regularized* recostructions in cone-beam tomogphy

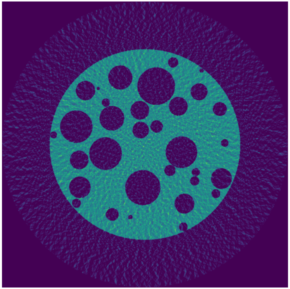
- Why regularized?: undersampled (few-view) data



Phantom

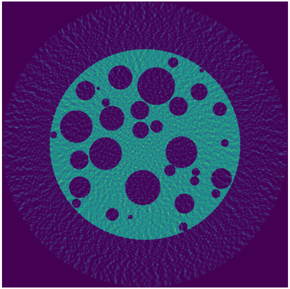


FDK 1024 views

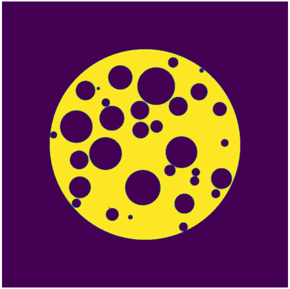


FDK 60 views

- Why regularized?: undersampled (few-view) data



FDK 60 views



TV 60 views (Condat-Vu)

- We pose the reconstruction problem as

$$u^\dagger = \underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{A}u - b\|^2 + \lambda\mathcal{R}(u), \quad \lambda_i > 0,$$

where $\mathcal{R}$ is a convex lower semicontinuos function and λ is a multivalued parameter.

- We pose the reconstruction problem as

$$u^{\dagger} = \underset{u}{\operatorname{argmin}} \, \tfrac{1}{2}\|\mathrm{A}u - b\|^2 + \lambda\mathcal{R}(u), \quad \lambda_i > 0,$$

where $\mathcal{R}$ is a convex lower semicontinuos function and λ is a multivalued parameter.

- and the parameter learning problem as

$$\begin{aligned} \lambda^{\dagger} = \quad & \underset{\lambda_i>0}{\operatorname{argmin}} \left\{ L(\lambda) := \tfrac{1}{2}\|u_\lambda - u_{\mathrm{true}}\|^2 \right\} \\ & \text{subject to } u_\lambda \in \underset{u}{\operatorname{argmin}} \, \tfrac{1}{2}\|\mathrm{A}u - b\|^2 + \lambda\mathcal{R}(u), \end{aligned}$$

where u_{true} is some available ground-truth image.

- For some $\mathcal{R}$, FISTA is a valid choise:

$$\begin{cases} u_{k+1} = \mathrm{prox}_{\gamma\lambda\mathcal{R}}\left(\hat{u}_k - \gamma \mathrm{A}^* \left(\mathrm{A}\hat{u}_k - b\right)\right) \\ t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right) \\ \hat{u}_{k+1} = u_{k+1} + \dfrac{t_k - 1}{t_{k+1}}\left(u_{k+1} - u_k\right) \end{cases}$$

- For some $\mathcal{R}$, FISTA is a valid choise:

$$\begin{cases} u_{k+1} = \operatorname{prox}_{\gamma\lambda\mathcal{R}}\left(\hat{u}_k - \gamma \mathsf{A}^*\left(\mathsf{A}\hat{u}_k - b\right)\right) \\ t_{k+1} = \frac{1}{2}\left(1 + \sqrt{1 + 4t_k^2}\right) \\ \hat{u}_{k+1} = u_{k+1} + \dfrac{t_k - 1}{t_{k+1}}\left(u_{k+1} - u_k\right) \end{cases}$$

- where computing the proximity operator

$$\operatorname{prox}_{\gamma\lambda\mathcal{R}}(b) = \underset{u}{\operatorname{argmin}}\left\{\frac{1}{2}\|u - b\|^2 + \gamma\lambda\mathcal{R}(u)\right\}$$

is now a *denosing* problem for a noisy image b regularized with $\mathcal{R}$.

- To avoid the denoising step, the Condat-Vu algorithm extends $\mathcal{R}$ as $\mathcal{R} \circ \mathsf{L} + h$ to now solve

$$\underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}u) + \lambda h(u), \quad \lambda_i > 0,$$

with L a linear operator and h also convex and lower semicontinuos,

- To avoid the denoising step, the Condat-Vu algorithm extends $\mathcal{R}$ as $\mathcal{R} \circ \mathsf{L} + h$ to now solve

$$\underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}u) + h(u), \quad \lambda_i > 0,$$

with L a linear operator and h also convex and lower semicontinuos,

- To avoid the denoising step, the Condat-Vu algorithm extends $\mathcal{R}$ as $\mathcal{R} \circ \mathrm{L} + h$ to now solve

$$\underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{A}u - b\|^2 + \lambda\mathcal{R}(\mathrm{L}u) + h(u), \quad \lambda_i > 0,$$

with L a linear operator and h also convex and lower semicontinuos,

- It iterates as

$$\begin{cases} u_{k+1} = \operatorname{prox}_{\gamma\lambda h}\left(u_k - \gamma \mathrm{L}^* w_k - \gamma \mathrm{A}^*\left(\mathrm{A}u_k - b\right)\right) \\ w_{k+1} = \operatorname{prox}_{\tau(\lambda\mathcal{R})^*}\left(w_k + \tau \mathrm{L} u_{k+1}\right) \end{cases}$$

- $\mathcal{R}$ could be *simple* now if we *split* it from L, the idea is to have a close-form expression of $\operatorname{prox}_{\mathcal{R}^*}(w_k)$

- Recall our learning problem

$$\lambda^\dagger = \underset{\lambda>0}{\operatorname{argmin}} \left\{ L(\lambda) := \frac{1}{2}\|u_\lambda - u_{\text{true}}\|^2 \right\}$$
$$\text{subject to } u_\lambda \in \underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathrm{A}u - b\|^2 + \lambda \mathcal{R}(u),$$

[1] De los Reyes, Schönlieb, Valkonen. J. Math. Imaging Vis. (2017)
[2] De los Reyes, Villacis. (2022)

- Recall our learning problem

$$\lambda^\dagger = \underset{\lambda>0}{\operatorname{argmin}} \left\{ L(\lambda) := \frac{1}{2}\|u_\lambda - u_{\text{true}}\|^2 \right\}$$
$$\text{subject to } u_\lambda \in \underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(u),$$

- We will solve it with a gradient-based algorithm, challenge: computing the gradient of L with respect to λ
- Appraches based on smoothing $\mathcal{R}$ and the adjoint state[1] exist, or based on nonsmooth trust regions[2]. However we will explore *automatic differentiation* (AD) approaches.
- The main challengue (with high-resolution problems) is now memory-related.

[1] De los Reyes, Schönlieb, Valkonen. J. Math. Imaging Vis. (2017)
[2] De los Reyes, Villacis. (2022)

- We need the derivatives

$$\frac{\partial L}{\partial \lambda_i} = \frac{\partial L}{\partial u_\lambda} \frac{\partial u_\lambda}{\partial \lambda_i}$$

- We need the derivatives

$$\frac{\partial L}{\partial \lambda_i} = \frac{\partial L}{\partial u_\lambda} \frac{\partial u_\lambda}{\partial \lambda_i}$$

- therefore we need $\frac{\partial u_k}{\partial \lambda_i}$ where u_k is the k-iteration of e.g. Condat-Vu

$$\begin{cases} u_k = \mathrm{prox}_{\gamma\lambda h}\left(u_{k-1} - \gamma \mathrm{L}^* w_{k-1} - \gamma \mathrm{A}^*\left(\mathrm{A}u_k - b\right)\right) \\ w_k = \mathrm{prox}_{\tau(\lambda\mathcal{R})^*}\left(w_{k-1} + \tau \mathrm{L} u_k\right) \end{cases}$$

- We need the derivatives

$$\frac{\partial L}{\partial \lambda_i} = \frac{\partial L}{\partial u_\lambda} \frac{\partial u_\lambda}{\partial \lambda_i}$$

- therefore we need $\frac{\partial u_k}{\partial \lambda_i}$ where u_k is the k-iteration of e.g. Condat-Vu

$$\begin{cases} u = \mathrm{prox}_{\gamma h}\left(u - \gamma \mathsf{L}^* w - \gamma \mathsf{A}^* \left(\mathsf{A} u - b\right)\right) \\ w = \mathrm{prox}_{\tau(\lambda \mathcal{R})^*}\left(w + \tau \mathsf{L} u\right) \end{cases}$$

- We need the derivatives

$$\frac{\partial L}{\partial \lambda_i} = \frac{\partial L}{\partial u_\lambda} \frac{\partial u_\lambda}{\partial \lambda_i}$$

- therefore we need $\frac{\partial u_k}{\partial \lambda_i}$ where u_k is the k-iteration of e.g. Condat-Vu

$$\begin{cases} u = \mathrm{prox}_{\gamma h}\left(u - \gamma \mathsf{L}^* w - \gamma \mathsf{A}^*\left(\mathsf{A}u - b\right)\right) \\ w = \mathrm{prox}_{\tau(\lambda\mathcal{R})^*}\left(w + \tau \mathsf{L}u\right) \end{cases}$$

which has the form

$$\begin{cases} u = p\left(f(u, w)\right) \\ w = q\left(g(u, w), \lambda\right) \end{cases}$$

- Finally, we need the derivatives

$$\frac{\partial u}{\partial \lambda_i} = \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\frac{\partial w}{\partial \lambda_i}\right)$$

- Finally, we need the derivatives

$$\begin{aligned}\frac{\partial u}{\partial \lambda_i} &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\frac{\partial w}{\partial \lambda_i}\right) \\ &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\left(\frac{\partial q}{\partial g}\left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial g}{\partial w}\frac{\partial w}{\partial \lambda_i}\right) + \frac{\partial q}{\partial \lambda_i}\right)\right)\end{aligned}$$

- Finally, we need the derivatives

$$\begin{aligned}\frac{\partial u}{\partial \lambda_i} &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\frac{\partial w}{\partial \lambda_i}\right)\\ &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\left(\frac{\partial q}{\partial g}\left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial g}{\partial w}\frac{\partial w}{\partial \lambda_i}\right) + \frac{\partial q}{\partial \lambda_i}\right)\right)\end{aligned}$$

- Mind the terms $\left\{\frac{\partial q}{\partial g}, \frac{\partial q}{\partial \lambda_i}\right\}$

- Finally, we need the derivatives

$$\begin{aligned}\frac{\partial u}{\partial \lambda_i} &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\frac{\partial w}{\partial \lambda_i}\right)\\ &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\left(\frac{\partial q}{\partial g}\left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial g}{\partial w}\frac{\partial w}{\partial \lambda_i}\right) + \frac{\partial q}{\partial \lambda_i}\right)\right)\end{aligned}$$

- Mind the terms $\left\{\frac{\partial q}{\partial g}, \frac{\partial q}{\partial \lambda_i}\right\}$
- Stored RAM memory in one GP / CV / informed CV iteration

	GP	CV	iCV
stored memory (MB.)	153	82	25

- Finally, we need the derivatives

$$\begin{aligned}\frac{\partial u}{\partial \lambda_i} &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\frac{\partial w}{\partial \lambda_i}\right)\\ &= \frac{\partial p}{\partial f}\left(\frac{\partial f}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial f}{\partial w}\left(\frac{\partial q}{\partial g}\left(\frac{\partial g}{\partial u}\frac{\partial u}{\partial \lambda_i} + \frac{\partial g}{\partial w}\frac{\partial w}{\partial \lambda_i}\right) + \frac{\partial q}{\partial \lambda_i}\right)\right)\end{aligned}$$

- Mind the terms $\left\{\frac{\partial q}{\partial g}, \frac{\partial q}{\partial \lambda_i}\right\}$
- Stored RAM memory in one GP / CV / informed CV iteration

	GP	CV	iCV
stored memory (MB.)	153	82	25

- If needed, truncated backpropagation could be employed as well

- Example: total variation and nonnegativity :

$$\mathcal{R} = \| \cdot \|_{2,1}, \quad L = \nabla, \quad h = \imath_{\mathbb{R}_+}$$

$$q(w, \lambda) = \text{prox}_{\lambda \mathcal{R}^*}(w) = \Pi_{\|w\|_{2,\infty} \leq \lambda}(w) = \frac{w}{\max\left\{1, \lambda^{-1}|w|_2\right\}}$$

- Example: total variation and nonnegativity :

$$\mathcal{R} = \|\cdot\|_{2,1}, \quad L = \nabla, \quad h = \imath_{\mathbb{R}_+}$$

$$q(w,\lambda) = \mathrm{prox}_{\lambda\mathcal{R}^*}(w) = \Pi_{\|w\|_{2,\infty}\leq\lambda}(w) = \frac{w}{\max\left\{1, \lambda^{-1}|w|_2\right\}}$$

then we have the derivatives

$$\frac{\partial}{\partial\lambda} q(w,\lambda) = \frac{w|w|_2}{\lambda^2 \max^2\left\{1, \lambda^{-1}|w|_2\right\}} H(\lambda^{-1}|w|_2)),$$

$$\frac{\partial}{\partial w} q(w,\lambda) = \frac{H(\lambda^{-1}|w|_2)w^2/|w|_2}{\lambda \max^2\left\{1, \lambda^{-1}|w|_2\right\}} - \frac{1}{\max^2\left\{1, \lambda^{-1}|w|_2\right\}}$$

with H the Heaviside step function

- Example: sparsity on some new basis and nonnegativity :

$$\mathcal{R} = \|\cdot\|_1, \quad L = \mathcal{W}, \quad h = \imath_{\mathbb{R}_+}$$

$$q(w,\lambda) = \operatorname{prox}_{\lambda\mathcal{R}^*}(w) = \Pi_{\|w\|_{2,\infty}\leq\lambda}(w) = \frac{w}{\max\left\{1, \lambda^{-1}|w|\right\}}$$

then we have the derivatives

$$\frac{\partial}{\partial\lambda} q(w,\lambda) = \frac{w|w|}{\lambda^2 \max^2\left\{1, \lambda^{-1}||w|\right\}} H(\lambda^{-1}|w|)),$$

$$\frac{\partial}{\partial w} q(w,\lambda) = \frac{H(\lambda^{-1}|w|)w^2/|w|}{\lambda \max^2\left\{1, \lambda^{-1}|w|\right\}} - \frac{1}{\max^2\left\{1, \lambda^{-1}|w|\right\}}$$

with H the Heaviside step function

- Example: total *generalized* variation and nonnegativity :

$$
\begin{aligned}
& \underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda \mathcal{R}(\mathsf{L}(u,v)) + \lambda h(u,v), \quad \lambda_i > 0, \\
= \; & \underset{u,v}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda_1 \|\nabla u - v\|_{2,1} + \lambda_2 \|Ev\|_{2,1} + h(u,v)
\end{aligned}
$$

and we can still apply Condat-Vu with

$$
\mathsf{L} = \begin{pmatrix} \nabla & -Id \\ 0 & E \end{pmatrix}, \quad \mathcal{R} = \begin{pmatrix} \|\cdot\|_{2,1} \\ \|\cdot\|_{2,1} \end{pmatrix}, \quad h = \begin{pmatrix} \imath_{\mathbb{R}_+} \\ 0 \end{pmatrix}
$$

- Example: total *generalized* variation and nonnegativity :

$$\begin{aligned} & \underset{u}{\operatorname{argmin}} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}(u,v)) + \lambda h(u,v), \quad \lambda_i > 0, \\ = \; & \underset{u,v}{\operatorname{argmin}} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda_1\|\nabla u - v\|_{2,1} + \lambda_2\|Ev\|_{2,1} + h(u,v) \end{aligned}$$

and we can still apply Condat-Vu with

$$\mathsf{L} = \begin{pmatrix} \nabla & -Id \\ 0 & E \end{pmatrix}, \quad \mathcal{R} = \begin{pmatrix} \|\cdot\|_{2,1} \\ \|\cdot\|_{2,1} \end{pmatrix}, \quad h = \begin{pmatrix} \imath_{\mathbb{R}_+} \\ 0 \end{pmatrix}$$

then we know from tv the derivatives

$$\frac{\partial}{\partial\lambda}\mathrm{prox}_{\lambda\mathcal{R}^*}(w), \quad \frac{\partial}{\partial w}\mathrm{prox}_{\lambda\mathcal{R}^*}(w)$$

with now $w = (w_1, w_2)$ the dual variable of (u, v)

- Example: *infimal convolution* total variation and nonnegativity :

$$
\begin{aligned}
& \underset{u}{\operatorname{argmin}} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}(u,v)) + \lambda h(u,v), \quad \lambda_i > 0, \\
= \; & \underset{u,v}{\operatorname{argmin}} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda_1\|\nabla u - \nabla v\|_{2,1} + \lambda_2\|E\nabla v\|_{2,1} + h(u,v)
\end{aligned}
$$

and we can still apply Condat-Vu with

$$
\mathsf{L} = \begin{pmatrix} \nabla & -\nabla \\ 0 & E\nabla \end{pmatrix}, \quad \mathcal{R} = \begin{pmatrix} \|\cdot\|_{2,1} \\ \|\cdot\|_{2,1} \end{pmatrix}, \quad h = \begin{pmatrix} \imath_{\mathbb{R}_+} \\ 0 \end{pmatrix}
$$

then we know from tv the derivatives

$$
\frac{\partial}{\partial\lambda}\operatorname{prox}_{\lambda\mathcal{R}^*}(w), \quad \frac{\partial}{\partial w}\operatorname{prox}_{\lambda\mathcal{R}^*}(w)
$$

with now $w = (w_1, w_2)$ the dual variable of (u, v)

- Example: *infimal convolution* total variation and nonnegativity :

$$\begin{aligned} &\operatorname*{argmin}_{u} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}(u,v)) + \lambda h(u,v), \quad \lambda_i > 0, \\ = \ &\operatorname*{argmin}_{u,v} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda_1\|\nabla u - \nabla v\|_{2,1} + \lambda_2\|E\nabla v\|_{2,1} + h(u,v) \end{aligned}$$

and we can still apply Condat-Vu with

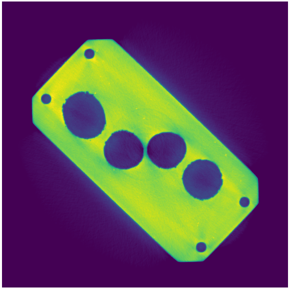
$$\mathsf{L} = \begin{pmatrix} \nabla & -\nabla \\ 0 & E\nabla \end{pmatrix}, \quad \mathcal{R} = \begin{pmatrix} \|\cdot\|_{2,1} \\ \|\cdot\|_{2,1} \end{pmatrix}, \quad h = \begin{pmatrix} \imath_{\mathbb{R}_+} \\ 0 \end{pmatrix}$$

then we know from tv the derivatives

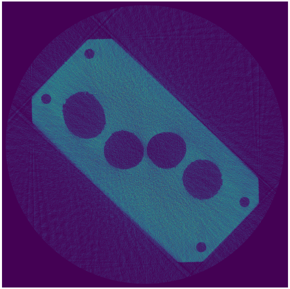
$$\frac{\partial}{\partial\lambda}\mathrm{prox}_{\lambda\mathcal{R}^*}(w), \quad \frac{\partial}{\partial w}\mathrm{prox}_{\lambda\mathcal{R}^*}(w)$$

with now $w = (w_1, w_2)$ the dual variable of (u, v)

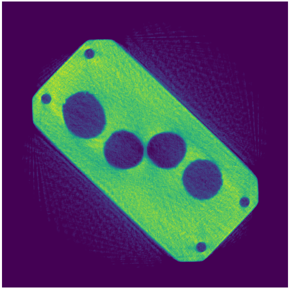
- PG, Bellens, Dewulf, arXiv:2412.10034 (2025)



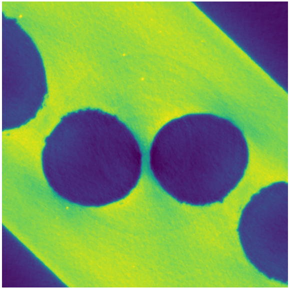
3000 projections



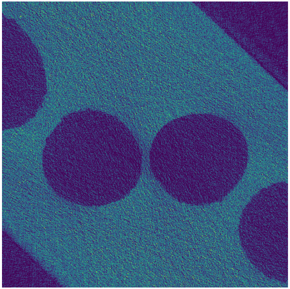
100 projections FDK



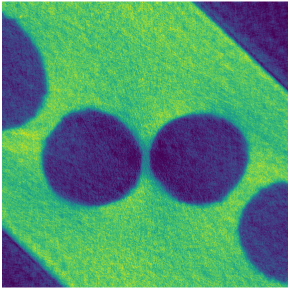
100 projections LSQR



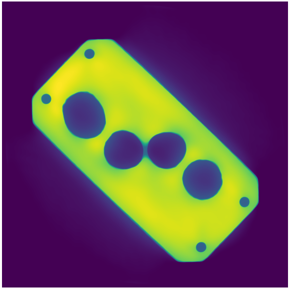
3000 projections



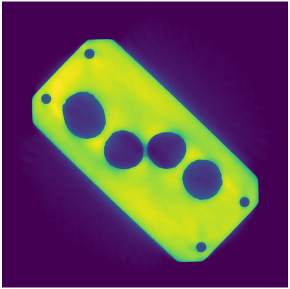
100 projections FDK



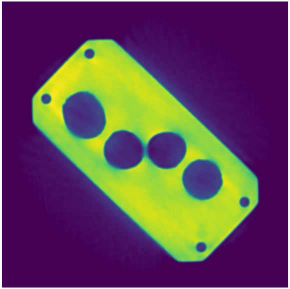
100 projections LSQR



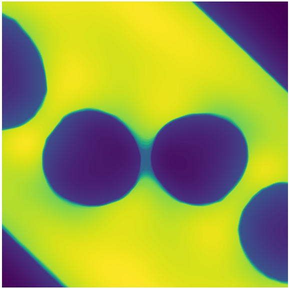
tv

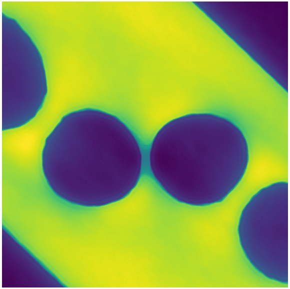


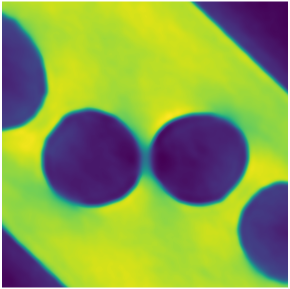
ictv

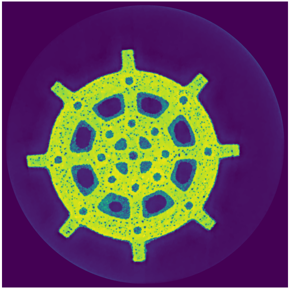


tgv

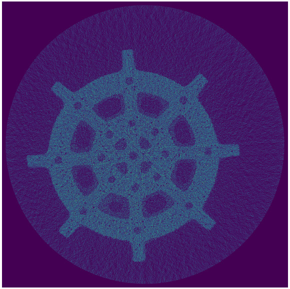

tv


ictv

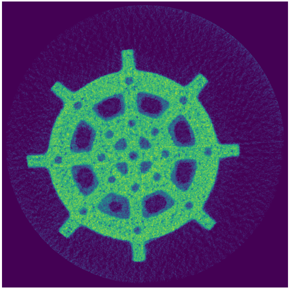

tgv



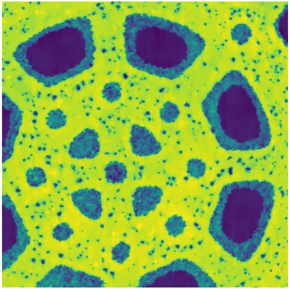
3000 projections



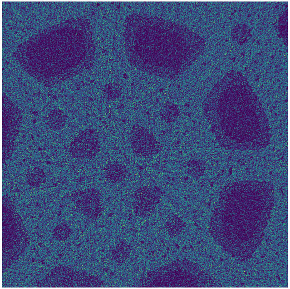
100 projections FDK



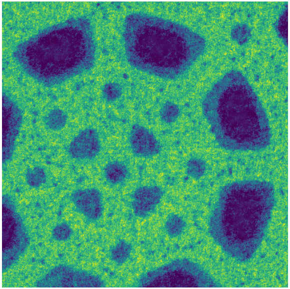
100 projections LSQR



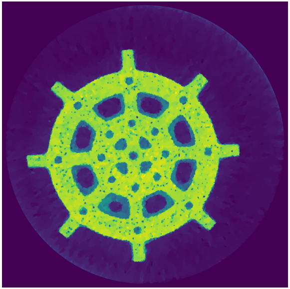
3000 projections



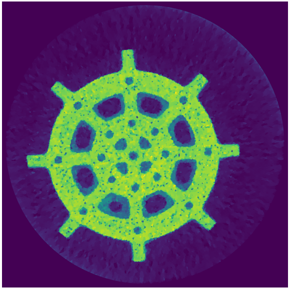
100 projections FDK



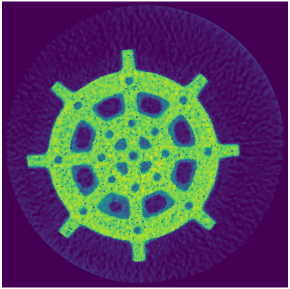
100 projections LSQR



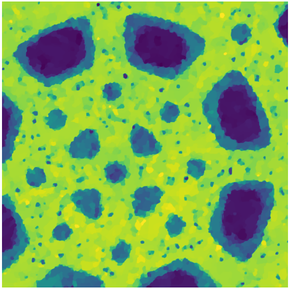
tv



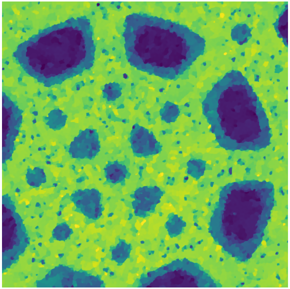
ictv



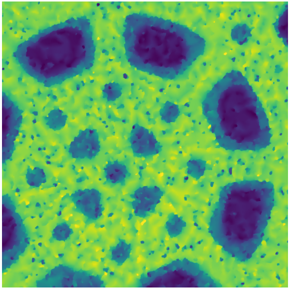
tgv



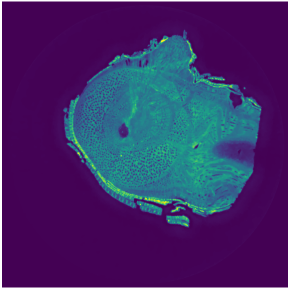
tv



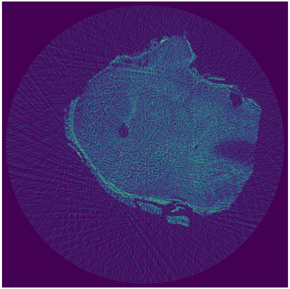
ictv



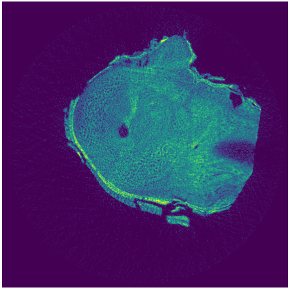
tgv



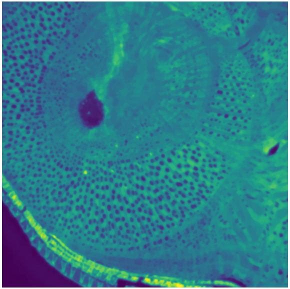
3000 projections



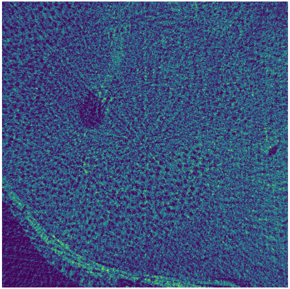
100 projections FDK



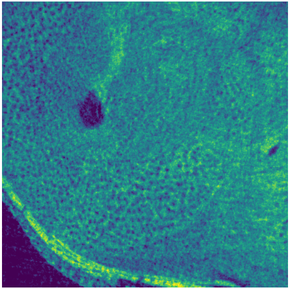
100 projections LSQR



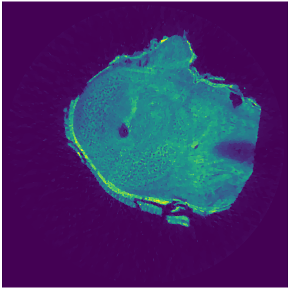
3000 projections



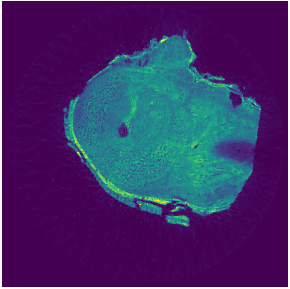
100 projections FDK



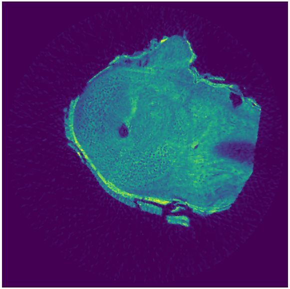
100 projections LSQR



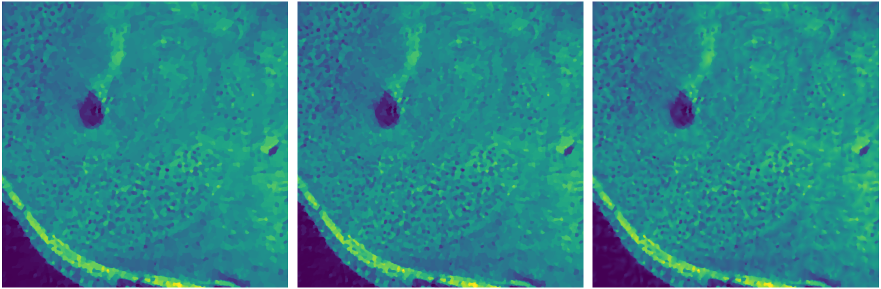
tv



ictv



tgv



tv ictv tgv

- Recall our reconstruction problem

$$\underset{u}{\operatorname{argmin}} \frac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}u) + h(u), \quad \lambda_i > 0,$$

- Recall our reconstruction problem

$$\operatorname*{argmin}_{u} \tfrac{1}{2}\|\mathsf{A}u - b\|^2 + \lambda\mathcal{R}(\mathsf{L}u) + h(u), \quad \lambda_i > 0,$$

- A can be reduced by performing l ($\ll \mathrm{pixels}$) steps of Golub-Kahan bidiagonalization giving the decomposition

$$\mathsf{AV} = \mathsf{UB},$$

where the orthogonal matrix $\mathsf{V} \in \mathbb{R}^{n\times\ell}$ spans the Krylov subspace

$$K_\ell(\mathsf{A}^\mathsf{T}\mathsf{A}, \mathsf{A}^\mathsf{T}b) = \mathrm{span}\{\mathsf{A}^\mathsf{T}b, (\mathsf{A}^\mathsf{T}\mathsf{A})\mathsf{A}^\mathsf{T}b, \ldots, (\mathsf{A}^\mathsf{T}\mathsf{A})^{\ell-1}\mathsf{A}^\mathsf{T}b\},$$

then we solve

$$\min_{y\in\mathbb{R}^\ell} \tfrac{1}{2}\|\mathsf{UB}y - b\|^2 + \mathcal{R}\left(\mathsf{LV}y\right) + h\left(\mathsf{V}y\right),$$

and because of $\mathsf{U}^\mathsf{T} b = \|b\| \mathsf{e}_1$ with $\mathsf{e}_1 = [1, 0, ..., 0]^\mathsf{T}$, we are solving

$$\min_{y \in \mathbb{R}^\ell} \frac{1}{2} \|\mathsf{B} y - \|b\| \mathsf{e}_1\|^2 + \mathcal{R}(\mathsf{LV} y) + h(\mathsf{V} y) \tag{1}$$

which can already be solved with Condat-Vu

$$\begin{cases} y_{k+1} = \mathrm{prox}_{\gamma h \circ \mathsf{V}} \left(y_k - \gamma \mathsf{V}^\mathsf{T} \mathsf{L}^\mathsf{T} w_k - \gamma \mathsf{B}^\mathsf{T} (\mathsf{B} y_k - \beta \mathsf{e}_1) \right) \\ w_{k+1} = \mathrm{prox}_{\tau (\lambda \mathcal{R})^*} (w_k + \tau \mathsf{LV} y_{k+1}) \end{cases}$$

and because of $\mathsf{U}^\mathsf{T} b = \|b\| \mathsf{e}_1$ with $\mathsf{e}_1 = [1, 0, ..., 0]^\mathsf{T}$, we are solving

$$\min_{y \in \mathbb{R}^\ell} \frac{1}{2} \|\mathsf{B} y - \|b\| \mathsf{e}_1\|^2 + \mathcal{R}\left(\mathsf{LV} y\right) + h\left(\mathsf{V} y\right) \tag{1}$$

which can already be solved with Condat-Vu

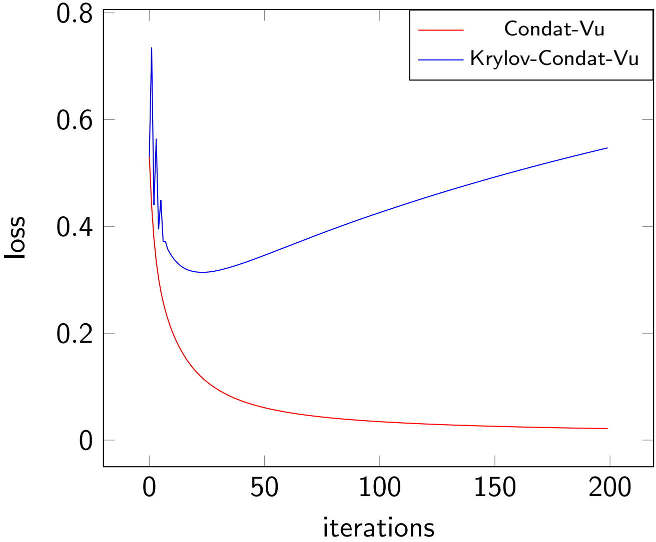
$$\begin{cases} y_{k+1} = \operatorname{prox}_{\gamma h \circ \mathsf{V}} \left(y_k - \gamma \mathsf{V}^\mathsf{T} \mathsf{L}^\mathsf{T} w_k - \gamma \mathsf{B}^\mathsf{T} (\mathsf{B} y_k - \beta \mathsf{e}_1) \right) \\ w_{k+1} = \operatorname{prox}_{\tau (\lambda \mathcal{R})^*} \left(w_k + \tau \mathsf{LV} y_{k+1} \right) \end{cases}$$

we also have $\operatorname{prox}_{\gamma h \circ \mathsf{V}} = \mathsf{V}^\mathsf{T} \circ \operatorname{prox}_{\gamma h} \circ \mathsf{V}$, the iteration becomes

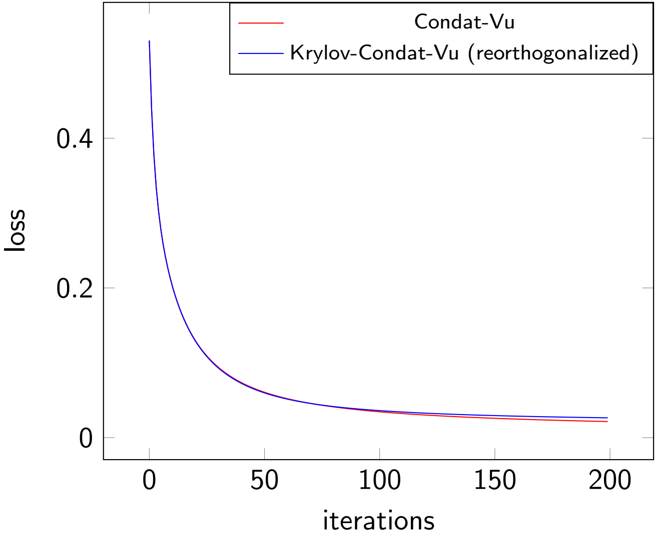
$$\begin{cases} y_{k+1} = \mathsf{V}^\mathsf{T} \operatorname{prox}_{\gamma h} \left(-\gamma \mathsf{L}^\mathsf{T} w_k + \mathsf{V} \left(y_k - \gamma \mathsf{B}^\mathsf{T} (\mathsf{B} y_k - \beta \mathsf{e}_1) \right) \right) \\ w_{k+1} = \operatorname{prox}_{\tau (\lambda \mathcal{R})^*} \left(w_k + \tau \mathsf{LV} y_{k+1} \right) \end{cases}$$

then (y_k) convergences weakly to a solution y of (1) and $x = \mathsf{V} y$ is our final reconstruction.

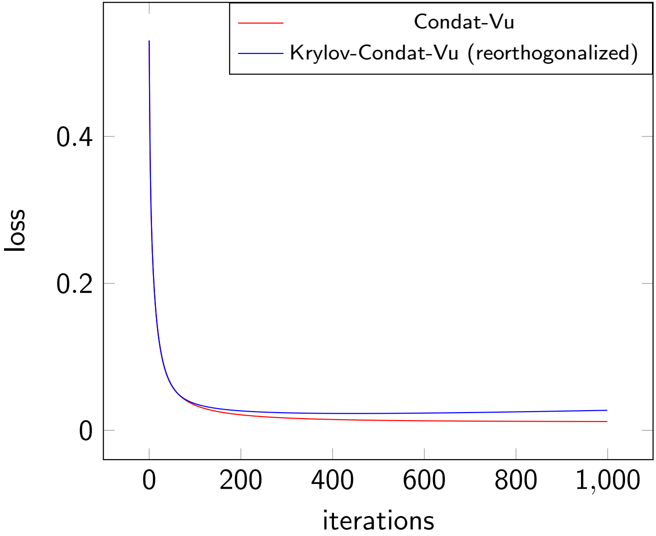
Convergence behaviour

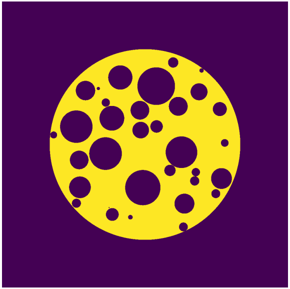


Convergence behavior

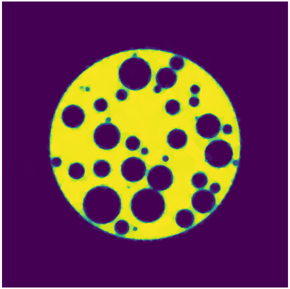


Convergence behavior

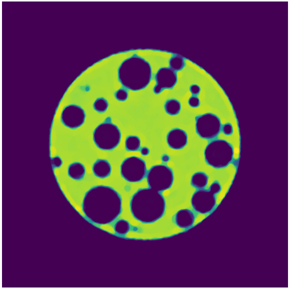




sample



Condat-Vu 60 projections



Krylov-Condat-Vu 60 projections

- Automatic differentiation in bilevel learning can be feasible even in high dimensional problems.
- Condat-Vu acceleration seems promising as well[3]

[3]Driggs, Ehrhardt, Schönlieb, Tang. 2024

- Automatic differentiation in bilevel learning can be feasible even in high dimensional problems.
- Condat-Vu acceleration seems promising as well[3]

[3] Driggs, Ehrhardt, Schönlieb, Tang. 2024

- Automatic differentiation in bilevel learning can be feasible even in high dimensional problems.
- Condat-Vu acceleration seems promising as well[3]
- Krylov-Condat-Vu could accelerate considerably the computing time in this kind og problems (particularly in the learning step)
- Still needs a better understanding around numerical instabilities

[3] Driggs, Ehrhardt, Schönlieb, Tang. 2024

$\rightarrow$ thanks !