Plug-and-Play Methods and Their Convergence

Wotao Yin

Joint with Ernest Ryu, Jialin Liu, Robert Hannah Sicheng Wang, Xiaohan Chen, Zhangyang Wang

IMA Workshop on Computational Imaging
October 2019

1 UCLA Math
2 Texas A&M CSE
Image processing via optimization

Consider recovering or denoising an image through the optimization

\[
\min_{x \in \mathbb{R}^d} f(x) + \gamma g(x),
\]

- \(x\) is image
- \(f(x)\) is data fidelity (a posteriori knowledge)
- \(g(x)\) is noisiness of the image (a priori knowledge)
- \(\gamma \geq 0\) is relative importance between \(f\) and \(g\)
We often use first-order methods, such as ADMM

\[
x^{k+1} = \arg\min_{x \in \mathbb{R}^d} \left\{ \sigma^2 g(x) + \frac{1}{2} \| x - (y^k - u^k) \|^2 \right\}
\]

\[
y^{k+1} = \arg\min_{y \in \mathbb{R}^d} \left\{ \alpha f(y) + \frac{1}{2} \| y - (x^{k+1} + u^k) \|^2 \right\}
\]

\[
u^{k+1} = u^k + x^{k+1} - y^{k+1}
\]

with \( \sigma^2 = \alpha \gamma \).
More concise notation

\[ x^{k+1} = \text{Prox}_{\sigma^2 g}(y^k - u^k) \]
\[ y^{k+1} = \text{Prox}_{\alpha f}(x^{k+1} + u^k) \]
\[ u^{k+1} = u^k + x^{k+1} - y^{k+1}. \]

The proximal operator of \( h \) is

\[ \text{Prox}_{\alpha h}(z) = \arg\min_{x \in \mathbb{R}^d} \left\{ \alpha h(x) + \frac{1}{2} \| x - z \|^2 \right\}. \]

(Well-defined if \( h \) is proper, closed, and convex.)
Interpretations of ADMM subroutines

The subroutine $\text{Prox}_{\sigma^2 g} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ is a denoiser, i.e.,

$\text{Prox}_{\sigma^2 g} : \text{noisy image} \mapsto \text{less noisy image}$

$\text{Prox}_{\alpha f} : \mathbb{R}^d \rightarrow \mathbb{R}^d$ enforces consistency with measured data, i.e.,

$\text{Prox}_{\alpha f} : \text{less consistent} \mapsto \text{more consistent with data}$
Other denoisers

However, some state-of-the-art image denoisers do not originate from optimization problems. (E.g. NLM, BM3D, and CNN.) Nevertheless, such a denoiser $H_\sigma : \mathbb{R}^d \rightarrow \mathbb{R}^d$ still has the interpretation

$$H_\sigma : \text{noisy image} \mapsto \text{less noisy image}$$

where $\sigma \geq 0$ is a noise parameter.

It is possible to integrate such denoisers with existing algorithms such as ADMM or proximal gradient?
To address this question, Venkatakrishnan et al.\textsuperscript{3} proposed Plug-and-Play ADMM (PnP-ADMM), which simply replaces the proximal operator $\text{Prox}_{\sigma^2 g}$ with the denoiser $H_\sigma$:

$$
\begin{align*}
  x^{k+1} &= H_\sigma(y^k - u^k) \\
  y^{k+1} &= \text{Prox}_{\alpha f}(x^{k+1} + u^k) \\
  u^{k+1} &= u^k + x^{k+1} - y^{k+1}.
\end{align*}
$$

Surprisingly and remarkably, this ad-hoc method exhibited great empirical success, and spurred much follow-up work.

\textsuperscript{3}Venkatakrishnan, Bouman, and Wohlberg, Plug-and-play priors for model based reconstruction, IEEE GlobalSIP, 2013.
Example: Inpainting

Original image

5% random sampling

Example: Inpainting

Other method PnP-ADMM with NLM

Example: Super resolution

Low resolution input  Other method  Other method

Other method  Other method  Other method  PnP-ADMM with BM3D

Example: Single photon imaging

Corrupted image

other method

other method

PnP-ADMM with BM3D

Example: Single photon imaging

Corrupted image

other method

PnP-ADMM with BM3D

Given empirical success of PnP, we ask: **when does PnP converge?**

- We prove convergence under a Lipschitz condition
- We enforce the condition by spectral normalization
- We present numerical results validating our theory
- We introduce a novel proof technique
Plug-and-play forward-backward splitting:

\[ x^{k+1} = H_\sigma(I - \alpha \nabla f)(x^k) \]  

(PPN-FBS)

where \( \alpha > 0 \).
PNP-FBS is a fixed-point iteration, and \( x^* \) is a fixed point if

\[
x^* = H_\sigma(I - \alpha \nabla f)(x^*).
\]

Interpretation of fixed points: A compromise between making the image agree with measurements and making the image less noisy.
Plug-and-play alternating directions method of multipliers:

\[ x^{k+1} = H_\sigma (y^k - u^k) \]
\[ y^{k+1} = \text{Prox}_{\alpha f}(x^{k+1} + u^k) \]  
(PNP-ADMM)
\[ u^{k+1} = u^k + x^{k+1} - y^{k+1} \]

where \( \alpha > 0 \).
PnP convergence
What we do not assume

If we assume $2H_\sigma - I$ is nonexpansive, standard tools of monotone operator theory tell us that PnP-ADMM converges. However, this assumption is unrealistic\(^5\) so we do not assume it.

We do not assume $H_\sigma$ is continuously differentiable.


Convergence via contraction
Main assumption

Rather, we assume $H_\sigma : \mathbb{R}^d \to \mathbb{R}^d$ satisfies

$$\|(H_\sigma - I)(x) - (H_\sigma - I)(y)\| \leq \varepsilon \|x - y\|$$  \hspace{1cm} (A)

for all $x, y \in \mathbb{R}^d$ for some $\varepsilon \geq 0$. Since $\sigma$ controls the strength of the denoising, we can expect $H_\sigma$ to be close to identity for small $\sigma$. If so, Assumption (A) is reasonable.
Convergence of PNP-FBS

Theorem

Assume $H_\sigma$ satisfies assumption (A) for some $\varepsilon \geq 0$. Assume $f$ is $\mu$-strongly convex, $f$ is differentiable, and $\nabla f$ is $L$-Lipschitz. Then

$$T = H_\sigma(I - \alpha \nabla f)$$

satisfies

$$\|T(x) - T(y)\| \leq \max\{|1 - \alpha \mu|, |1 - \alpha L|\}(1 + \varepsilon)\|x - y\|$$

for all $x, y \in \mathbb{R}^d$. The coefficient is less than 1 if

$$\frac{1}{\mu(1 + 1/\varepsilon)} < \alpha < \frac{2}{L} - \frac{1}{L(1 + 1/\varepsilon)}.$$ 

Such an $\alpha$ exists if $\varepsilon < 2\mu/(L - \mu)$.
Convergence of PNP-DRS

Theorem
Assume $H_\sigma$ satisfies assumption (A) for some $\varepsilon \geq 0$. Assume $f$ is $\mu$-strongly convex and differentiable. Then

$$T = \frac{1}{2}I + \frac{1}{2}(2H_\sigma - I)(2\text{Prox}_{\alpha f} - I)$$

satisfies

$$\|T(x) - T(y)\| \leq \frac{1 + \varepsilon + \varepsilon \alpha \mu + 2\varepsilon^2 \alpha \mu}{1 + \alpha \mu + 2\varepsilon \alpha \mu} \|x - y\|$$

for all $x, y \in \mathbb{R}^d$. The coefficient is less than 1 if

$$\frac{\varepsilon}{(1 + \varepsilon - 2\varepsilon^2)\mu} < \alpha, \quad \varepsilon < 1.$$
PnP-FBS vs. PnP-ADMM

PNP-FBS and PNP-ADMM share the same fixed points\(^6\)\(^7\).

PNP-FBS is easier to implement as it requires \(\nabla f\) rather than \(\text{Prox}_{\alpha f}\).

PNP-ADMM has better convergence properties as demonstrated by Theorems 1 and 2 and our experiments.


Convergence via contraction
Real spectral normalization: Enforcing Assumption (A)
We use DnCNN\textsuperscript{9}, which learns the residual mapping with a 17-layer CNN.

Given a noisy observation $y = x + e$, where $x$ is the clean image and $e$ is noise, the residual mapping $R$ outputs the noise, i.e., $R(y) = e$ so that $y - R(y)$ is the clean recovery. Learning the residual mapping is a common approach in deep learning-based image restoration.

Lipschitz constrained deep denoising

Note

\[(I - H_\sigma)(y) = y - H_\sigma(y) = R(y),\]

with denoiser \(H_\sigma\), residual \(R\), and identity \(I\).

Enforcing

\[\|(I - H_\sigma)(x) - (I - H_\sigma)(y)\| \leq \varepsilon\|x - y\| \quad (A)\]

is equivalent to constraining the Lipschitz constant of \(R\). We propose a variant of the spectral normalization for this.

Real spectral normalization: Enforcing Assumption (A)
Real Spectral Normalization (realSN) accurately constrains the network’s Lipschitz constant through a power iteration with the convolutional linear operator $\mathcal{K}_l : \mathbb{R}^{C_{\text{in}} \times h \times w} \rightarrow \mathbb{R}^{C_{\text{out}} \times h \times w}$, where $h, w$ are input’s height and width, and its conjugate (transpose) operator $\mathcal{K}_l^*$. The iteration maintains $U_l \in \mathbb{R}^{C_{\text{out}} \times h \times w}$ and $V_l \in \mathbb{R}^{C_{\text{in}} \times h \times w}$ to estimate the leading left and right singular vectors respectively. During each forward pass of the neural network, realSN conducts:

1. Apply one step of the power method with operator $\mathcal{K}_l$:

$$V_l \leftarrow \frac{\mathcal{K}_l^*(U_l)}{\|\mathcal{K}_l^*(U_l)\|_2}, \quad U_l \leftarrow \frac{\mathcal{K}_l(V_l)}{\|\mathcal{K}_l(V_l)\|_2}.$$ 

2. Normalize the convolutional kernel $K_l$ with estimated spectral norm:

$$K_l \leftarrow \frac{K_l}{\sigma(K_l)}, \text{ where } \sigma(K_l) = \langle U_l, \mathcal{K}_l(V_l) \rangle$$

We can view realSN as an approximate projected gradient enforcing the Lipschitz continuity constraint.
Implementation details

We train SimpleCNN and DnCNN in the setting of Gaussian denoising with $40 \times 40$ patches of the BSD500 dataset, natural images. RealSN constrains the Lipschitz constant to no more than 1.

On an Nvidia GTX 1080 Ti, DnCNN took 4.08 hours and realSN-DnCNN took 5.17 hours to train, so the added cost of realSN is mild.
We run PnP iterations, calculate $\frac{\| (I - H_\sigma)(x) - (I - H_\sigma)(y) \|}{\| x - y \|}$ between the iterates and the limit, and plot the histogram. The maximum value, the red bar, lower-bounds $\varepsilon$ of (A). Convergence of PnP-ADMM requires $\varepsilon < 1$. The results prove BM3D violates this assumption and illustrate that RealSN indeed controls (reduces) the Lipschitz constant.
Numerical validation
Given a true image $x_{\text{true}} \in \mathbb{R}^d$, we observe Poisson random variables

$$y_i \sim \text{Poisson}((x_{\text{true}})_i)$$

for $i = 1, \ldots, d$. We use the negative log-likelihood

$$f(x) = \sum_{i=1}^{d} -y_i \log(x_i) + x_i.$$  

For further details of the experimental setup, see the main paper or \textsuperscript{11}.


Experimental validation
Poisson denoising

Corrupted 3.36dB

Recovery 20.28dB

Experimental validation
**Poisson denoising**

<table>
<thead>
<tr>
<th>Method</th>
<th>BM3D</th>
<th>RealSN-DnCNN</th>
<th>RealSN-SimpleCNN</th>
</tr>
</thead>
<tbody>
<tr>
<td>PNP-ADMM</td>
<td>23.4617</td>
<td><strong>23.5873</strong></td>
<td>18.7890</td>
</tr>
<tr>
<td>PNP-FBS</td>
<td>18.5835</td>
<td>22.2154</td>
<td><strong>22.7280</strong></td>
</tr>
</tbody>
</table>

PSNR of the PnP methods with BM3D, RealSN-DnCNN, and RealSN-SimpleCNN plugged in. In both PnP methods, one of the two denoisers using RealSN, for which we have theory, outperforms BM3D.
Single photon imaging

The measurement model of quanta image sensors is

\[ z = 1(y \geq 1), \quad y \sim \text{Poisson}(\alpha_{sg}Gx_{\text{true}}) \]

where \( x_{\text{true}} \in \mathbb{R}^d \) is the true image, \( G : \mathbb{R}^d \to \mathbb{R}^{dK} \) duplicates each pixel to \( K \) pixels, \( \alpha_{sg} \in \mathbb{R} \) is sensor gain, \( K \) is the oversampling rate, \( z \in \{0, 1\}^{dK} \) is the observed binary photons. (\( y \) is not measured.) The likelihood function is

\[
f(x) = \sum_{j=1}^{n} -K_j^0 \log(e^{-\alpha_{sg}x_j/K}) - K_j^1 \log(1 - e^{-\alpha_{sg}x_j/K}),
\]

where \( K_j^1 \) is the number of ones in the \( j \)-th unit pixel, \( K_j^0 \) is the number of zeros in the \( j \)-th unit pixel.

For further details of the experimental setup, see the main paper or \(^{12}\).

\(^{12}\) Elgendy and Chan, Image reconstruction and threshold design for quanta image sensors, IEEE ICIP, 2016.
Single photon imaging

Measurement pixels take integer values between 0 and $K = 64$. 

Experimental validation
Single photon imaging

PnP-ADMM with RealSN-DnCNN provides best PSNR. We also observe that RealSN makes PnP converge more stably.

<table>
<thead>
<tr>
<th></th>
<th>PnP-FBS, $\alpha = 0.005$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR</td>
<td>BM3D</td>
<td>RealSN-DnCNN</td>
<td>RealSN-SimpleCNN</td>
</tr>
<tr>
<td>Iteration 50</td>
<td>28.7933</td>
<td>27.9617</td>
<td>29.0062</td>
</tr>
<tr>
<td>Iteration 100</td>
<td>29.0510</td>
<td>27.9887</td>
<td>29.0517</td>
</tr>
<tr>
<td>Best Overall</td>
<td><strong>29.5327</strong></td>
<td>28.4065</td>
<td>29.3563</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>PnP-ADMM, $\alpha = 0.01$</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Average PSNR</td>
<td>BM3D</td>
<td>RealSN-DnCNN</td>
<td>RealSN-SimpleCNN</td>
</tr>
<tr>
<td>Iteration 50</td>
<td>30.0034</td>
<td>31.0032</td>
<td>29.2154</td>
</tr>
<tr>
<td>Iteration 100</td>
<td>30.0014</td>
<td>31.0032</td>
<td>29.2151</td>
</tr>
<tr>
<td>Best Overall</td>
<td>30.0474</td>
<td><strong>31.0431</strong></td>
<td>29.2155</td>
</tr>
</tbody>
</table>

Experimental validation
Compressed sensing MRI

PnP is useful in medical imaging when we do not have enough data for end-to-end training: train the denoiser $H_\sigma$ on natural images, and “plug” it into the PnP framework to be applied to medical images.

Given a true image $x_{\text{true}} \in \mathbb{C}^d$, CS-MRI measures

$$y = \mathcal{F}_p x_{\text{true}} + \varepsilon_e,$$

where $\mathcal{F}_p$ is the Fourier k-domain subsampling (partial Fourier operator), and $\varepsilon_e \sim N(0, \sigma_e I_k)$ is measurement noise. We use the objective function

$$f(x) = (1/2) \| y - \mathcal{F}_p x \|^2.$$

For further details of the experimental setup, see the main paper or 13.

---

Compressed sensing MRI

Radial sampling $k$-space

Recovery 19.09dB

$k$-space measurement is complex-valued so we plot the absolute value.

Experimental validation
Compressed sensing MRI

PSNR (in dB) for 30% sampling with additive Gaussian noise $\sigma_e = 15$. RealSN generally improves the performance.

<table>
<thead>
<tr>
<th>Sampling approach</th>
<th>Random</th>
<th>Radial</th>
<th>Cartesian</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Brain</td>
<td>Bust</td>
<td>Brain</td>
</tr>
<tr>
<td>Image</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Zero-filling</td>
<td>9.58</td>
<td>7.00</td>
<td>9.29</td>
</tr>
<tr>
<td>TV$^{14}$</td>
<td>16.92</td>
<td>15.31</td>
<td>15.61</td>
</tr>
<tr>
<td>RecRF$^{15}$</td>
<td>16.98</td>
<td>15.37</td>
<td>16.04</td>
</tr>
<tr>
<td>BM3D-MRI$^{16}$</td>
<td>17.31</td>
<td>13.90</td>
<td>16.95</td>
</tr>
<tr>
<td>RealSN-DnCNN</td>
<td>19.82</td>
<td>16.60</td>
<td>18.96</td>
</tr>
<tr>
<td>SimpleCNN</td>
<td>15.58</td>
<td>12.19</td>
<td>15.06</td>
</tr>
<tr>
<td>RealSN-SimpleCNN</td>
<td>17.65</td>
<td>14.98</td>
<td>16.52</td>
</tr>
<tr>
<td>PnP-ADMM BM3D</td>
<td>19.61</td>
<td>17.23</td>
<td>18.94</td>
</tr>
<tr>
<td>DnCNN</td>
<td>19.86</td>
<td>17.05</td>
<td>19.00</td>
</tr>
<tr>
<td>RealSN-DnCNN</td>
<td>19.91</td>
<td>17.09</td>
<td>19.08</td>
</tr>
<tr>
<td>SimpleCNN</td>
<td>16.68</td>
<td>12.56</td>
<td>16.83</td>
</tr>
<tr>
<td>RealSN-SimpleCNN</td>
<td>17.77</td>
<td>14.89</td>
<td>17.00</td>
</tr>
</tbody>
</table>

$^{14}$Lustig, Santos, Lee, Donoho, and Pauly, SPARS, 2005.
$^{15}$Yang, Zhang, and Yin, IEEE JSTSP, 2010.
Scaled relative graph (proof technique)
Scaled relative graph (SRG) a 2D geometric tool to analyze (monotone, contractive, nonexpansive, ...) operators, which are building blocks of many first-order optimization algorithms.

- Captures core ideas in algebraic proofs.
- Serves as rigorous proofs.
An example

**Proposition:** Suppose $f$ is $\mu$-strongly convex and $L$-Lipschitz differentiable. Then, $x^{k+1} = x^k - \alpha \nabla f(x^k)$ converges linearly at sharp rate:

$$R = \max\{|1 - \alpha \mu|, |1 - \alpha L|\}.$$ 

**Proof by diagrams:**

(definitions and steps will be explained)
SRG of operator $A$ (single- or multi-valued)

Pick $x \neq y$, $u \in Ax$, and $v \in Ay$. Plot a complex $z = re^{\phi i}$ with

$$r := \frac{\|u - v\|}{\|x - y\|}, \quad \phi := \pm \angle(u - v, x - y),$$

For example, if $A = I$, $z \equiv (1, 0)$.

In general, $z$ depends on $(x, u), (y, v)$. So, we let

$$G(A) := \{z : x \neq y, u \in Ax, v \in Ay\} \cup \{\infty\} \text{ if } A \text{ is multi-valued}$$
Examples of SRGs

In $\mathbb{R}^2$, projection to any line:

$$A \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix} = \begin{bmatrix} \alpha \\ 2\beta \\ 3\gamma \end{bmatrix}$$

subdifferential of $\| \cdot \|_2$ in $\mathbb{R}^2$: $\bigcup \{ \infty \}$
Recall: Proposition: Suppose $f$ is $\mu$-strongly convex and $L$-Lipschitz differentiable. Then, $x^{k+1} = x^k - \alpha \nabla f(x^k)$ converges linearly at sharp rate:

$$R = \max\{|1 - \alpha \mu|, |1 - \alpha L|\}.$$ 

Rephrased using sets: Define sets

$$\partial F_{\mu, L} := \{\nabla f : f \text{ is a } \mu\text{-strongly convex, } L\text{-Lipschitz differentiable function}\}$$

$$\mathcal{L}_R := \{R\text{-Lipschitz operators}\}.$$ 

Then,

$$I - \alpha \partial F_{\mu, L} \subseteq \mathcal{L}_R.$$ 

holds for $R = \max\{|1 - \alpha \mu|, |1 - \alpha L|\}$, but not for any smaller $R$.

Results are stated with set relations
SRG of operator class \( \mathcal{A} \): \( \mathcal{G}(\mathcal{A}) = \bigcup_{A \in \mathcal{A}} \mathcal{G}(A) \)

\[ M_{\mu}: \mu\text{-strongly monotone operator} \]
\[ \partial F_{\mu, \infty}: \text{gradient of } \mu\text{-strgly-cvx function} \]

\[ C_{\beta}: \beta\text{-cocoercive operator} \]
\[ \partial F_{0, \frac{1}{\beta}}: \text{gradient of } \frac{1}{\beta}\text{-Lip.diff.cvx function} \]

\[ \mathcal{L}_{\mathcal{L}}: L\text{-Lipschitz operator} \]

\[ \mathcal{N}_{\theta}: \theta\text{-averaged operator} \]
Algebraic inclusion vs visual inclusion

By definition, $\mathcal{A} \subseteq \mathcal{B} \Rightarrow G(\mathcal{A}) \subseteq G(\mathcal{B})$.

Do we also have $\mathcal{A} \subseteq \mathcal{B} \Leftrightarrow G(\mathcal{A}) \subseteq G(\mathcal{B})$?

Not always!
SRG-full classes of operators

**Definition:** an class $\mathcal{A}$ of operators $\mathcal{H} \Rightarrow \mathcal{H}$ is **SRG-full** if

$$\mathcal{G}(\mathcal{A}') \subseteq \mathcal{G}(\mathcal{A}) \Rightarrow \mathcal{A}' \subseteq \mathcal{A}.$$ 

**Theorem:** if an operator class is defined by a 1-homogeneous equation of $\|u - v\|^2, \|x - y\|^2, \langle u - v, x - y \rangle$, then it is SRG-full.

In particular, classes $\mathcal{M}_{\mu}, \mathcal{C}_\beta, \mathcal{L}_L$, and $\mathcal{N}_\theta$ are SRG-full.

**Theorem:** If $\mathcal{B}$ is SRG-full, then

- $\mathcal{G}(\mathcal{A}) \subseteq \mathcal{G}(\mathcal{B}) \iff \mathcal{A} \subseteq \mathcal{B}$;
- $\mathcal{G}(\mathcal{A} \cap \mathcal{B}) = \mathcal{G}(\mathcal{A}) \cap \mathcal{G}(\mathcal{B})$. 
Transformation: scaling and translation

**Theorem:** for $\alpha, \beta \in \mathbb{R}$,

$$G(\beta I + \alpha A) = \beta + \alpha G(A).$$

**Proposition:** Suppose $f$ is $\mu$-strongly convex and $L$-Lipschitz differentiable. Then, $x^{k+1} = x^k - \alpha \nabla f(x^k)$ converges linearly at sharp rate:

$$R = \max\{|1 - \alpha \mu|, |1 - \alpha L|\}.$$ **Proof by diagrams:**

Since $\mathcal{L}_R$ is SRG-full, the last diagram implies $I - \alpha \partial F_{\mu,L} \subseteq \mathcal{L}_R$. 

(uses scaling, translation)
Geometric inversion

also know as *reflection in the unit circle*: $z \mapsto z^{-1}$.

A line is a generalized circle with infinite radius.
Including this generalization, the inversion of a circle is a circle.
Transformation: inversion

**Theorem:** $G(A^{-1}) = (G(A))^{-1}$ (operator inversion = geometric inversion)

**Proposition:** if $A$ is monotone and $\alpha > 0$, $J_{\alpha A} := (I + \alpha A)^{-1}$ is 1/2-cocoercive (firmly nonexpansive). Iteration $x^{k+1} = J_{\alpha A}(x^k)$ has sublinear convergence.

**Proof by diagrams:**

![Proof by diagrams](image_url)
Proposition: if $A$ is $\mu$-strongly monotone and $\alpha > 0$, $J_{\alpha A}$ is $1/(1 + \alpha \mu)$-Lipschitz. Iteration $x^{k+1} = J_{\alpha A}(x^k)$ has linear convergence.

Proof by diagrams:
Proposition 1: If $f$ is a $\mu$-strongly convex $L$-Lipschitz differentiable function and $\alpha > 0$, we have

- $\text{prox}_{\alpha f}$ is $\frac{1}{1+\alpha \mu}$-Lipschitz;
- $2\text{prox}_{\alpha f} - I$ is $R$-Lipschitz for $R = \max \left\{ \left| \frac{1-\alpha \mu}{1+\alpha \mu} \right|, \left| \frac{1-\alpha L}{1+\alpha L} \right| \right\}$, tight.

Proof by diagrams:
Composition of operators

**Theorem:** If $\mathcal{A}, \mathcal{B}$ are SRG-full, then excluding $\infty \cdot \emptyset$ cases

$$\mathcal{G}(\mathcal{A}\mathcal{B}) \supseteq \mathcal{G}(\mathcal{A})\mathcal{G}(\mathcal{B}).$$

In addition, if $\mathcal{A}$ satisfies *one of the two “arc properties”* or if $\mathcal{B}$ does, then

$$\mathcal{G}(\mathcal{A}\mathcal{B}) = \mathcal{G}(\mathcal{B}\mathcal{A}) = \mathcal{G}(\mathcal{A})\mathcal{G}(\mathcal{B}).$$
Composition of firmly nonexpansive operators

**Theorem**\(^2\): Let \( \mathcal{N}_\theta \) be the class of \( \theta \)-averaged operators (firmly nonexpansive operators if \( \theta = 1/2 \)). Then,

\[
\mathcal{N}_{1/2} \cap \mathcal{N}_{1/2} \subset \mathcal{N}_{2/3} \quad \text{strictly, } 2/3 \text{ is sharp.}
\]

**Corollary:** Douglas-Rachford Method or ADMM has sublinear convergence.

---

\(^2\)[Combettes and Yamada, 2015]
Convergence theory for PnP

Assume denoising operator

\[ H : \text{noisy image} \mapsto \text{less noisy image} \]

is close to \( I \), specifically,

\[ \|(H - I)x - (H - I)y\|^2 \leq \epsilon^2 \|x - y\|^2, \quad \forall x, y. \]
Assume $f$ is $\mu$-strongly convex and $L$-Lipschitz differentiable.

\[ R = \max\{|1 - \alpha \mu|, |1 - \alpha L|\} \]

\[ \mathcal{G}(I - \alpha \nabla f) \]

**Theorem:** The Forward-backward PnP operator

\[ T = H(I - \alpha \nabla f) \]

is contractive for proper $\epsilon \in (0, 2\mu/(L - \mu))$ and $\alpha$. 

\[ -R(1 + \epsilon) \quad R(1 + \epsilon) \]

\[ \mathcal{G}(T) \]
Conclusion

1. PnP-FBS and PnP-ADMM converge under a Lipschitz condition on \((I - H)\)
2. Real spectral normalization enforces the Lipschitz condition
3. Experiments validate the theory

References:


Code is available at Github.

Thank you!