Generalizing Nesterov's Scheme and Magical High-order Methods

Tao Hong

working with Prof. Jeffrey A. Fessler and Prof. Luis Hernandez Garcia

fMRI Lab & EECS
University of Michigan, Ann Arbor
Email: tahong@umich.edu

Jeff

Luis

Outline

Generalizing Nesterov's Scheme

Motivation

General Nest. Acc.

Numerical Tests

Magical High-order Methods $\rightarrow$ CS MRI Reco.

Problem Formulation

Our Suggestion

Numerical Results

Optimization Problems

$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Optimization Problems

$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Optimization Problems

$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

Optimization Problems

$\min_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP)

Optimization Problems

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

Optimization Problems

$\min\limits_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

$\min\limits_{\boldsymbol{x} \in \mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x})$: g nonsmooth & convex & L Lip. Const. of $f(\boldsymbol{x})$

Optimization Problems

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x})$: g nonsmooth & convex & L Lip. Const. of $f(\boldsymbol{x})$

Define $\mathrm{Prox}_g(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{x}\|_2^2 + g(\boldsymbol{v})$

Solver: plain $\rightarrow$ proximal gradient method

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_g(\boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

Optimization Problems

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x})$: g nonsmooth & convex & L Lip. Const. of $f(\boldsymbol{x})$

Define $\mathrm{Prox}_g(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{x}\|_2^2 + g(\boldsymbol{v})$

Solver: plain $\rightarrow$ proximal gradient method

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_g(\boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

Nest. Acc. (APM or FISTA)

$$\boldsymbol{x}_{x+1} = \mathrm{Prox}_g(\boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)),\ \boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$$

Optimization Problems

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x})$: g nonsmooth & convex & L Lip. Const. of $f(\boldsymbol{x})$

Define $\mathrm{Prox}_g(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{x}\|_2^2 + g(\boldsymbol{v})$

Solver: plain $\rightarrow$ proximal gradient method

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_g(\boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

Nest. Acc. (APM or FISTA)

$$\boldsymbol{x}_{x+1} = \mathrm{Prox}_g(\boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)),\ \boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$$

$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots} f(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots)$: block coordinate descent and acc. version

Optimization Problems

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x})$: f smooth & convex & L Lip. Const.

Solver: plain $\rightarrow$ gradient descent $\mathcal{O}(\frac{1}{k})$ $\boldsymbol{x}_{x+1} = \boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$

Nest. Acc. $\mathcal{O}(\frac{1}{k^2})$ $\boldsymbol{x}_{x+1} = \boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)$, $\boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$

$\rightarrow$ Cubic reg. (Nest. 06 MP) $\rightarrow$ Acc. version (Nest. 08 MP)

$\min_{\boldsymbol{x}\in\mathbb{R}^N} f(\boldsymbol{x}) + g(\boldsymbol{x})$: g nonsmooth & convex & L Lip. Const. of $f(\boldsymbol{x})$

Define $\mathrm{Prox}_g(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{v}} \frac{1}{2}\|\boldsymbol{v} - \boldsymbol{x}\|_2^2 + g(\boldsymbol{v})$

Solver: plain $\rightarrow$ proximal gradient method

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_g(\boldsymbol{x}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

Nest. Acc. (APM or FISTA)

$$\boldsymbol{x}_{x+1} = \mathrm{Prox}_g(\boldsymbol{v}_k - \frac{1}{L}\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k)),\ \boldsymbol{v}_{k+1} = t_k^1 \boldsymbol{x}_k + t_k^2 \boldsymbol{x}_{k+1}$$

$\min_{\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots} f(\boldsymbol{x}_1, \boldsymbol{x}_2, \cdots)$: block coordinate descent and acc. version

Many others...

Generalizing Nesterov's Acceleration

Can we accelerate an abstract solver by Nesterov's scheme?

Generalizing Nesterov's Acceleration

Can we accelerate an abstract solver by Nesterov's scheme?

Looks possible and we have some answers

Generalizing Nesterov's Acceleration

Can we accelerate an abstract solver by Nesterov's scheme?

Looks possible and we have some answers

DOI: 10.1002/nla.2417

RESEARCH ARTICLE

WILEY

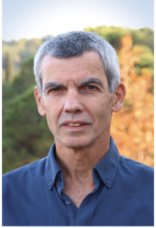
On adapting Nesterov's scheme to accelerate iterative methods for linear problems

Tao Hong | **Irad Yavneh**

Department of Computer Science,
Technion - Israel Institute of Technology,
Haifa, Israel

Abstract
Nesterov's well-known scheme for accelerating gradient descent in convex opti-

T. Hong and I. Yavneh, NLAA, 2022.



Generalizing Nesterov's Scheme

- Motivation
- General Nest. Acc.
- Numerical Tests

Magical High-order Methods $\rightarrow$ CS MRI Reco.

- Problem Formulation
- Our Suggestion
- Numerical Results

Problem Formulation

We consider ($\boldsymbol{A} \succeq 0$)

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \boldsymbol{x}^{\mathcal{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{f}^{\mathcal{T}} \boldsymbol{x} \Leftrightarrow \boldsymbol{A} \boldsymbol{x} = \boldsymbol{f}$$

Problem Formulation

We consider ($\boldsymbol{A} \succeq 0$)

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \boldsymbol{x}^{\mathcal{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{f}^{\mathcal{T}} \boldsymbol{x} \Leftrightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$$

Nest. formulation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{B}\boldsymbol{v}_k + \text{Constant} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{x}_{k+1} + c_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

Problem Formulation

We consider ($\boldsymbol{A} \succeq 0$)

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \boldsymbol{x}^{\mathcal{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{f}^{\mathcal{T}} \boldsymbol{x} \Leftrightarrow \boldsymbol{A} \boldsymbol{x} = \boldsymbol{f}$$

Nest. formulation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{B} \boldsymbol{v}_k + \text{Constant} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{x}_{k+1} + c_k (\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

$\boldsymbol{B}$ iteration matrix, e.g., $\boldsymbol{B} = \boldsymbol{I} - a_k \boldsymbol{A}$ (gradient descent)

Problem Formulation

We consider ($\boldsymbol{A} \succeq 0$)

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \boldsymbol{x}^{\mathcal{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{f}^{\mathcal{T}} \boldsymbol{x} \Leftrightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$$

Nest. formulation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{B}\boldsymbol{v}_k + \text{Constant} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{x}_{k+1} + c_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

$\boldsymbol{B}$ iteration matrix, e.g., $\boldsymbol{B} = \boldsymbol{I} - a_k \boldsymbol{A}$ (gradient descent)

Next question: how to choose c_k & can we get acceleration?

Problem Formulation

We consider ($\boldsymbol{A} \succeq 0$)

$$\min_{\boldsymbol{x} \in \mathbb{R}^N} \frac{1}{2} \boldsymbol{x}^{\mathcal{T}} \boldsymbol{A} \boldsymbol{x} - \boldsymbol{f}^{\mathcal{T}} \boldsymbol{x} \Leftrightarrow \boldsymbol{A}\boldsymbol{x} = \boldsymbol{f}$$

Nest. formulation:

$$\begin{aligned} \boldsymbol{x}_{k+1} &= \boldsymbol{B}\boldsymbol{v}_k + \text{Constant} \\ \boldsymbol{v}_{k+1} &= \boldsymbol{x}_{k+1} + c_k(\boldsymbol{x}_{k+1} - \boldsymbol{x}_k) \end{aligned}$$

$\boldsymbol{B}$ iteration matrix, e.g., $\boldsymbol{B} = \boldsymbol{I} - a_k \boldsymbol{A}$ (gradient descent)

Next question: how to choose c_k & can we get acceleration?

The answer is positive at least for some $\boldsymbol{B}$

UNIVERSITY OF MICHIGAN

Optimal Acceleration $\rightarrow$ The Choice of c_k

Assumption: $\boldsymbol{B}$ only has real eigenvalues.

Optimal Acceleration $\to$ The Choice of c_k

Assumption: $\boldsymbol{B}$ only has real eigenvalues.

$-1 < b_1 \leq \cdots \leq b_N < 1$ (eigenvalue $\boldsymbol{B}$) & $c_{cr}(b) = \frac{1-\sqrt{1-b}}{1+\sqrt{1-b}}$

Theorem

Let $-1 < b_1 \leq b_N < 1 \to c^* = c_{cr}(g(b_1, b_N))$

$$g(b_1, b_N) = \begin{cases} b_N, & b_N \geq -3b_1, \\ -\frac{8b_N b_1 (b_1 + b_N)}{(b_1 - b_N)^2}, & -\frac{1}{3}b_1 < b_N < -3b_1, \\ b_1, & b_N \leq -\frac{1}{3}b_1, \end{cases}$$

yielding conv. factor

$$r^* = \begin{cases} 1 - \sqrt{1 - b_N}, & b_N \geq -3b_1, \\ r(c^*, b_1) = r(c^*, b_N), & -\frac{1}{3}b_1 < b_N < -3b_1, \\ \sqrt{1 - b_1} - 1, & b_N \leq -\frac{1}{3}b_1. \end{cases}$$

$$r(c, b) = \frac{1}{2}\left|(1+c)b + sgn(b)\sqrt{(1+c)^2 b^2 - 4cb}\right|$$

Remarks

- $c_k \to c$

Remarks

- $c_k \to c$
- Nest. can converge even for some divergent $\boldsymbol{B}$ (whose spectral radii are larger than 1). Relax assumption from $-1 < b_1 \leq b_N < 1$ to $-3 < b_1 \leq b_N < 1$.

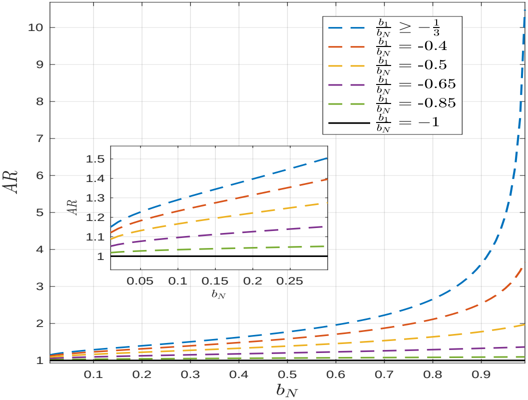
Remarks

- $c_k \to c$
- Nest. can converge even for some divergent $\boldsymbol{B}$ (whose spectral radii are larger than 1). Relax assumption from $-1 < b_1 \leq b_N < 1$ to $-3 < b_1 \leq b_N < 1$.
- Optimal step-size $\frac{4}{3L}$

UNIVERSITY OF MICHIGAN

Remarks

- $c_k \to c$
- Nest. can converge even for some divergent $\boldsymbol{B}$ (whose spectral radii are larger than 1). Relax assumption from $-1 < b_1 \leq b_N < 1$ to $-3 < b_1 \leq b_N < 1$.
- Optimal step-size $\frac{4}{3L}$
- $AR = \frac{\log r^*}{\log b_N}$ b_N conv. factor (plain)



$\boldsymbol{B}$ Complex Eigenvalues

Denote by $b^c = \bar{b}^c e^{j\theta}$
j: imaginary unit; $\bar{b}^c$: modulus; $\theta \in (-\pi, \pi]$: argument

$\boldsymbol{B}$ Complex Eigenvalues

Denote by $b^c = \bar{b}^c e^{j\theta}$

j: imaginary unit; $\bar{b}^c$: modulus; $\theta \in (-\pi, \pi]$: argument

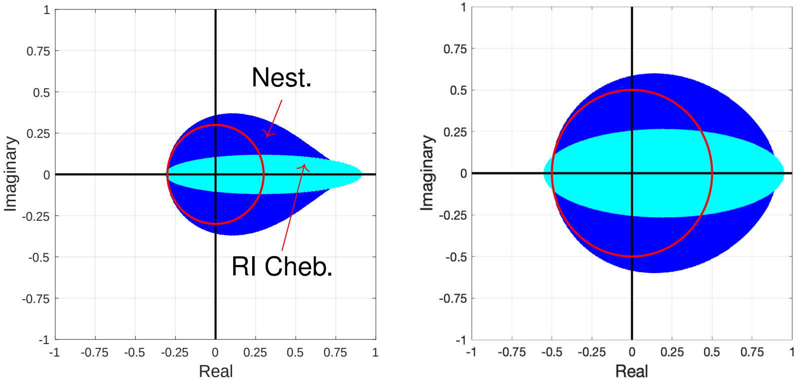
Theorem

In addition to $-1 < b_1 \leq \cdots \leq b_N < 1$ of $\boldsymbol{B}$,
$\boldsymbol{B}$ also has complex eigenvalues.
Then, c^* and r^* remain valid if the modulus of all complex eigenvalues satisfies

$$\bar{b}^c \leq \begin{cases} \frac{1}{3} b_N & b_N \geq -3b_1 \\ \min(|b_1|, |b_N|) & -\frac{1}{3} b_1 < b_N < -3b_1 \\ -\frac{1}{3} b_1 & b_N \leq -\frac{1}{3} b_1 \end{cases}.$$

Compare with RI Chebyshev Acc.

Left: $b_1 = -0.3$ and $b_N = 0.9$; Right: $b_1 = -0.5$ and $b_N = 0.9$.

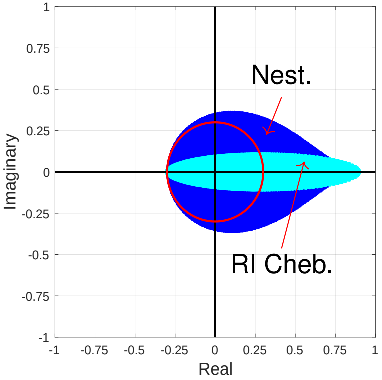


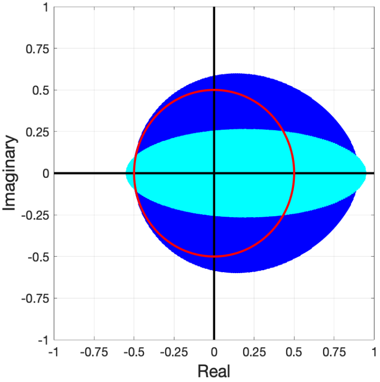
Cheb. Acc.

$$
\begin{aligned}
\boldsymbol{x}_1 &= \gamma(\boldsymbol{B}\boldsymbol{x}_0 + \text{Constant}) + (1-\gamma)\boldsymbol{x}_0, \\
\boldsymbol{x}_{k+1} &= \beta_{k+1}\{\gamma(\boldsymbol{B}\boldsymbol{x}_k + \text{Constant}) + (1-\gamma)\,\boldsymbol{x}_k\} + (1-\beta_{k+1})\,\boldsymbol{x}_{k-1}
\end{aligned}
$$

Compare with RI Chebyshev Acc.

Left: $b_1 = -0.3$ and $b_N = 0.9$; Right: $b_1 = -0.5$ and $b_N = 0.9$.





Nest. and RI Cheb. are different

Generalizing Nesterov's Scheme

- Motivation
- General Nest. Acc.
- Numerical Tests

Magical High-order Methods $\rightarrow$ CS MRI Reco.

- Problem Formulation
- Our Suggestion
- Numerical Results

Numerical Tests

Consider a diffusion equation:

$$-\nabla(\sigma(x,y)\nabla u(x,y)) = f(x,y)$$

Numerical Tests

Consider a diffusion equation:

$$-\nabla(\sigma(x,y)\nabla u(x,y)) = f(x,y)$$

Test on three examples:

1) $\sigma(x,y) = 1$;

2) $\sigma(x,y)$ sampled from log-normal distribution;

3) $\sigma(x,y)$ sampled from uniform distribution;

Numerical Tests

Consider a diffusion equation:

$$-\nabla(\sigma(x,y)\nabla u(x,y)) = f(x,y)$$

Test on three examples:

1) $\sigma(x,y) = 1$;

2) $\sigma(x,y)$ sampled from log-normal distribution;

3) $\sigma(x,y)$ sampled from uniform distribution;

Discretized on a 1024×1024 uniform grid $\rightarrow \boldsymbol{A}\boldsymbol{u} = \boldsymbol{f}$

Residual vector: $\boldsymbol{r}_k = \boldsymbol{f} - \boldsymbol{A}\boldsymbol{u}_k$; Residual norm: $\|\boldsymbol{r}_k\|_2$;

Conv. factor: $\lim_{k\to\infty} \frac{\|\boldsymbol{r}_{k+1}\|_2}{\|\boldsymbol{r}_k\|_2}$

Numerical Tests

Consider a diffusion equation:

$$-\nabla(\sigma(x,y)\nabla u(x,y)) = f(x,y)$$

Test on three examples:

1) $\sigma(x,y) = 1$;

2) $\sigma(x,y)$ sampled from log-normal distribution;

3) $\sigma(x,y)$ sampled from uniform distribution;

Discretized on a 1024×1024 uniform grid $\rightarrow \boldsymbol{Au} = \boldsymbol{f}$

Residual vector: $\boldsymbol{r}_k = \boldsymbol{f} - \boldsymbol{Au}_k$; Residual norm: $\|\boldsymbol{r}_k\|_2$;

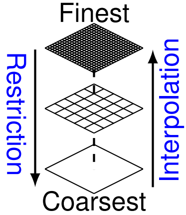
Conv. factor: $\lim_{k\to\infty} \frac{\|\boldsymbol{r}_{k+1}\|_2}{\|\boldsymbol{r}_k\|_2}$

$\boldsymbol{B}$: multigrid methods

Local relaxation

Restriction

Interpolation



Test 1: $\sigma(x, y) = 1$

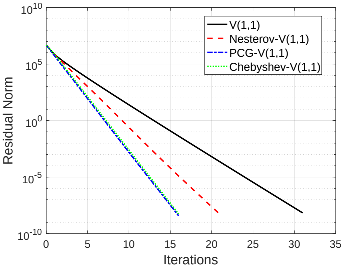
The Poisson problem

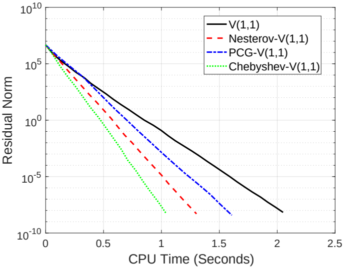
Damped Jacobi relaxation $\rightarrow$ $\boldsymbol{B}$ only has real eiganvalues

Test 1: $\sigma(x, y) = 1$

The Poisson problem

Damped Jacobi relaxation $\rightarrow$ ***B*** only has real eiganvalues





PCG: preconditioned conjugate gradient

$V(1,1)$: multigrid method

Test 1: $\sigma(x, y) = 1$

The Poisson problem

Red-Black relaxation $\rightarrow$ ***B*** contains complex & real eiganvalues

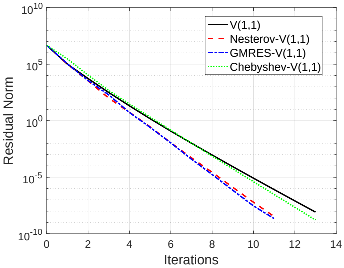
PCG: invalid

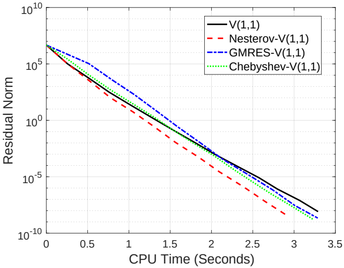
Test 1: $\sigma(x,y) = 1$

The Poisson problem

Red-Black relaxation $\rightarrow$ ***B*** contains complex & real eiganvalues

PCG: invalid





Test 2: $\sigma(x, y)$ log-normal distribution

B: Black Box multigrid method (complex)

Cheb. Acc.: invalid

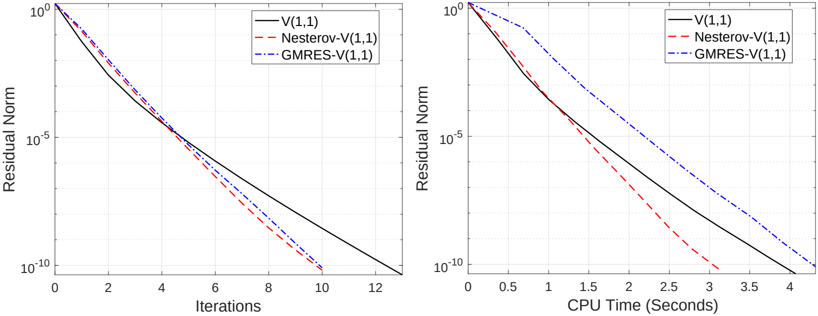
$V(1,1)$: multigrid method

Test 2: $\sigma(x, y)$ log-normal distribution

B: Black Box multigrid method (complex)

Cheb. Acc.: invalid

$V(1,1)$: multigrid method



Test 3: $\sigma(x, y)$ normal distribution

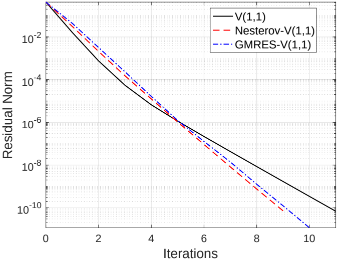
B: Black Box multigrid method (complex)

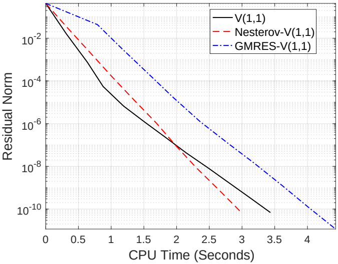
Cheb. Acc.: invalid

Test 3: $\sigma(x, y)$ normal distribution

B: Black Box multigrid method (complex)

Cheb. Acc.: invalid





Generalizing Nesterov's Scheme

Motivation

General Nest. Acc.

Numerical Tests

Magical High-order Methods $\rightarrow$ CS MRI Reco.

Problem Formulation

Our Suggestion

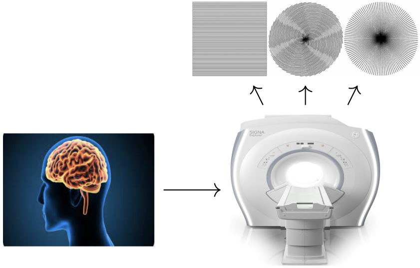
Numerical Results

MRI: Magnetic Resonance Imaging

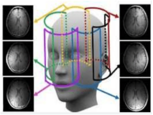
MRI: Magnetic Resonance Imaging

$\boldsymbol{x}^{\text{true}}$

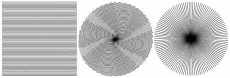
MRI: Magnetic Resonance Imaging

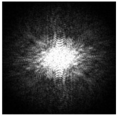


$\boldsymbol{x}^{\text{true}}$ forward process $\boldsymbol{A}$

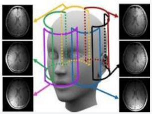


MRI: Magnetic Resonance Imaging

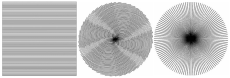


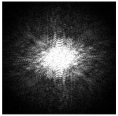


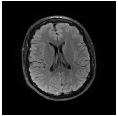
$\boldsymbol{x}^{\text{true}}$ forward process $\boldsymbol{A}$ k-space $\boldsymbol{y}$



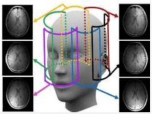
MRI: Magnetic Resonance Imaging



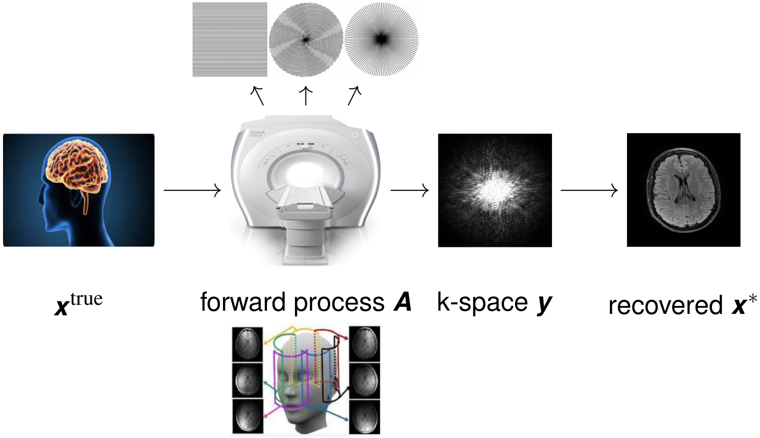




$\boldsymbol{x}^{\text{true}}$ forward process $\boldsymbol{A}$ k-space $\boldsymbol{y}$ recovered $\boldsymbol{x}^*$



MRI: Magnetic Resonance Imaging



$$\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$$

Problem Formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$

Fully sample $\boldsymbol{x}^* = \text{IFFT}(\boldsymbol{y})$

Problem Formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$

Fully sample $\boldsymbol{x}^* = \text{IFFT}(\boldsymbol{y})$

Reco. in CS MRI $\rightarrow$ Composite Optimization Problem

$$\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$$

Problem Formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$

Fully sample $\boldsymbol{x}^* = \text{IFFT}(\boldsymbol{y})$

Reco. in CS MRI $\rightarrow$ Composite Optimization Problem

$$\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$$

$$\boldsymbol{A} = [\boldsymbol{A}_i, \cdots], \boldsymbol{A}_i = \boldsymbol{P}\boldsymbol{F}\boldsymbol{S}_i$$

Problem Formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$

Fully sample $\boldsymbol{x}^* = \text{IFFT}(\boldsymbol{y})$

Reco. in CS MRI $\rightarrow$ Composite Optimization Problem

$$\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$$

$$\boldsymbol{A} = [\boldsymbol{A}_i, \cdots], \boldsymbol{A}_i = \boldsymbol{P}\boldsymbol{F}\boldsymbol{S}_i$$

$\boldsymbol{P}$: downsample; $\boldsymbol{F}$: (nonuniform) FFT; $\boldsymbol{S}_i$: sensitivity mapping

Problem Formulation $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}^{\text{true}} + \text{noise}$

Fully sample $\boldsymbol{x}^* = \mathrm{IFFT}(\boldsymbol{y})$

Reco. in CS MRI $\rightarrow$ Composite Optimization Problem

$$\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$$

$$\boldsymbol{A} = [\boldsymbol{A}_i, \cdots], \boldsymbol{A}_i = \boldsymbol{P}\boldsymbol{F}\boldsymbol{S}_i$$

$\boldsymbol{P}$: downsample; $\boldsymbol{F}$: (nonuniform) FFT; $\boldsymbol{S}_i$: sensitivity mapping

$R(\boldsymbol{x})$: regularizer

We consider wavelet, TV, or both.

Wavelet and TV

Wavelet Reco. $\lambda > 0$

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^M} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

$\boldsymbol{T}$: wavelet transform, image $\boldsymbol{T}^{-1}\boldsymbol{x}^*$

Wavelet and TV

Wavelet Reco. $\lambda > 0$

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^M} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

$\boldsymbol{T}$: wavelet transform, image $\boldsymbol{T}^{-1}\boldsymbol{x}^*$

TV Reco.

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \mathrm{TV}(\boldsymbol{x})$$

image $\boldsymbol{x}^*$

Wavelet and TV

Wavelet Reco. $\lambda > 0$

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^M} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \|\boldsymbol{x}\|_1$$

$\boldsymbol{T}$: wavelet transform, image $\boldsymbol{T}^{-1}\boldsymbol{x}^*$

TV Reco.

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda \mathrm{TV}(\boldsymbol{x})$$

image $\boldsymbol{x}^*$

Wavelet and TV Reco.

$$\boldsymbol{x}^* = \operatorname*{argmin}_{\boldsymbol{x} \in \mathbb{C}^N} \frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\alpha \|\boldsymbol{T}\boldsymbol{x}\|_1 + \lambda(1-\alpha)\mathrm{TV}(\boldsymbol{x}), \ \alpha \in (0,1)$$

image $\boldsymbol{x}^*$

Solvers $\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Solvers $\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Wavelet Reco. $\min_{\boldsymbol{x} \in \mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \rightarrow$ FISTA (APM)

Solvers $\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Wavelet Reco. $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \to$ FISTA (APM)

TV Reco.
$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\underbrace{\mathrm{TV}(\boldsymbol{x})}_{\|\boldsymbol{D}\boldsymbol{x}\|_1} \to$ APM iter. proximal mapping

Solvers $\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Wavelet Reco. $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \rightarrow$ FISTA (APM)

TV Reco.
$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\underbrace{\mathrm{TV}(\boldsymbol{x})}_{\|\boldsymbol{D}\boldsymbol{x}\|_1} \rightarrow$ APM iter. proximal mapping

Wavelet and TV Reco.
$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\alpha\|\boldsymbol{T}\boldsymbol{x}\|_1 + \lambda(1-\alpha)\mathrm{TV}(\boldsymbol{x}) \rightarrow$ APM&ADMM

Solvers $\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Wavelet Reco. $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \to$ FISTA (APM)

TV Reco.

$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\underbrace{\mathrm{TV}(\boldsymbol{x})}_{\|\boldsymbol{D}\boldsymbol{x}\|_1} \to$ APM iter. proximal mapping

Wavelet and TV Reco.

$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\alpha\|\boldsymbol{T}\boldsymbol{x}\|_1 + \lambda(1-\alpha)\mathrm{TV}(\boldsymbol{x}) \to$ APM&ADMM

Proximal gradient descent:

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}\|\boldsymbol{u}-\boldsymbol{x}_k\|_2^2 + R(\boldsymbol{u})}_{\mathrm{Prox}_{a_k R}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}}\, a_k R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_2^2}$$

Solvers $\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + R(\boldsymbol{x})$

Wavelet Reco. $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1 \to$ FISTA (APM)

TV Reco.
$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\underbrace{\mathrm{TV}(\boldsymbol{x})}_{\|\boldsymbol{D}\boldsymbol{x}\|_1} \to$ APM iter. proximal mapping

Wavelet and TV Reco.
$\min_{\boldsymbol{x}\in\mathbb{C}^N} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2 + \lambda\alpha\|\boldsymbol{T}\boldsymbol{x}\|_1 + \lambda(1-\alpha)\mathrm{TV}(\boldsymbol{x}) \to$ APM&ADMM

Proximal gradient descent:

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}\|\boldsymbol{u}-\boldsymbol{x}_k\|_2^2 + R(\boldsymbol{u})}_{\mathrm{Prox}_{a_k R}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}}\, a_k R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_2^2}$$

Nest. Acc.

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_{a_k R}(\boldsymbol{v}_k - a_k\nabla_{\boldsymbol{x}} f(\boldsymbol{v}_k))$$

$$\boldsymbol{v}_{k+1} = t_k^1\boldsymbol{x}_k + t_k^2\boldsymbol{x}_{k+1}$$

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + \lambda R(\boldsymbol{x})$$

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + \lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k) + R(\boldsymbol{u})}_{\mathrm{Prox}^{\boldsymbol{W}}_{a_k R}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}}\, a_k R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|^2_{\boldsymbol{W}},\ \boldsymbol{W}=\boldsymbol{B}_k}$$

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})} + \lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k) + R(\boldsymbol{u})}_{\mathrm{Prox}_{a_k R}^{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}}\, a_k R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{W}}^2,\ \boldsymbol{W}=\boldsymbol{B}_k}$$

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_{a_k R}^{\boldsymbol{B}_k}(\boldsymbol{x}_k - a_k\boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

$\boldsymbol{B}_k \approx \boldsymbol{A}^{\mathcal{H}}\boldsymbol{A}$, we use the SR1 method

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})}+\lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k) + R(\boldsymbol{u})}_{\mathrm{Prox}^{\boldsymbol{W}}_{a_k R}(\boldsymbol{x})=\operatorname{argmin}_{\boldsymbol{u}} a_k R(\boldsymbol{u})+\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|^2_{\boldsymbol{W}},\ \boldsymbol{W}=\boldsymbol{B}_k}$$

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}^{\boldsymbol{B}_k}_{a_k R}(\boldsymbol{x}_k - a_k \boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

$\boldsymbol{B}_k \approx \boldsymbol{A}^{\mathcal{H}}\boldsymbol{A}$, we use the SR1 method

CQNPMs converge faster than APMs $\rightarrow$ iterations

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})}+\lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k) + R(\boldsymbol{u})}_{\mathrm{Prox}_{a_k R}^{\boldsymbol{W}}(\boldsymbol{x})=\mathrm{argmin}_{\boldsymbol{u}}\, a_k R(\boldsymbol{u})+\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{W}}^2,\ \boldsymbol{W}=\boldsymbol{B}_k}$$

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_{a_k R}^{\boldsymbol{B}_k}(\boldsymbol{x}_k - a_k\boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

$\boldsymbol{B}_k \approx \boldsymbol{A}^{\mathcal{H}}\boldsymbol{A}$, we use the SR1 method

CQNPMs converge faster than APMs $\rightarrow$ iterations

Wall (CPU or GPU) time?

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})}+\lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k) + R(\boldsymbol{u})}_{\mathrm{Prox}_{a_kR}^{\boldsymbol{W}}(\boldsymbol{x})=\mathrm{argmin}_{\boldsymbol{u}}\, a_kR(\boldsymbol{u})+\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{W}}^2,\ \boldsymbol{W}=\boldsymbol{B}_k}$$

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}_{a_kR}^{\boldsymbol{B}_k}(\boldsymbol{x}_k - a_k\boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

$\boldsymbol{B}_k \approx \boldsymbol{A}^{\mathcal{H}}\boldsymbol{A}$, we use the SR1 method

CQNPMs converge faster than APMs $\rightarrow$ iterations

Wall (CPU or GPU) time?

Slow because of $\mathrm{Prox}_{a_kR}^{\boldsymbol{B}_k}$

Our Suggestion – Complex Quasi-Newton Proximal Methods (CQNPMs)

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x}-\boldsymbol{y}\|_2^2}_{f(\boldsymbol{x})}+\lambda R(\boldsymbol{x})$$

$$\boldsymbol{x}_{k+1} = \underbrace{\operatorname*{argmin}_{\boldsymbol{u}} f(\boldsymbol{x}_k) + \langle \nabla f(\boldsymbol{x}_k), \boldsymbol{u}-\boldsymbol{x}_k\rangle + \frac{1}{2a_k}(\boldsymbol{u}-\boldsymbol{x}_k)^{\mathcal{H}}\boldsymbol{B}_k(\boldsymbol{u}-\boldsymbol{x}_k)+R(\boldsymbol{u})}_{\mathrm{Prox}^{\boldsymbol{W}}_{a_kR}(\boldsymbol{x})=\mathrm{argmin}_{\boldsymbol{u}}\, a_kR(\boldsymbol{u})+\frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|^2_{\boldsymbol{W}},\ \boldsymbol{W}=\boldsymbol{B}_k}$$

$$\boldsymbol{x}_{k+1} = \mathrm{Prox}^{\boldsymbol{B}_k}_{a_kR}(\boldsymbol{x}_k - a_k\boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k))$$

$\boldsymbol{B}_k \approx \boldsymbol{A}^{\mathcal{H}}\boldsymbol{A}$, we use the SR1 method

CQNPMs converge faster than APMs $\rightarrow$ iterations

Wall (CPU or GPU) time?

Slow because of $\mathrm{Prox}^{\boldsymbol{B}_k}_{a_kR}$

A Complex Quasi-Newton Proximal Method for Image Reconstruction in Compressed Sensing MRI

Tao Hong, Luis Hernandez-Garcia, and Jeffrey A. Fessler, *Fellow, IEEE*

arXiv:2303.02586

Challenging Issues – Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

$$\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}}\, R(\boldsymbol{u}) + \tfrac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_{\boldsymbol{W}}^2$$

$$R(\boldsymbol{u}) = \|\cdot\|_1, \mathrm{TV}, \alpha\|\cdot\|_1 + (1-\alpha)\mathrm{TV}$$

Challenging Issues – Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

$$\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{argmin}_{\boldsymbol{u}} R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_{\boldsymbol{W}}^2$$

$$R(\boldsymbol{u}) = \|\cdot\|_1, \mathrm{TV}, \alpha\|\cdot\|_1 + (1-\alpha)\mathrm{TV}$$

Define $\mathcal{L} : \mathbb{C}^{(I-1)\times J} \times \mathbb{C}^{I\times(J-1)} \to \mathbb{C}^{I\times J}$ that

$$\mathcal{L}(\boldsymbol{P}, \boldsymbol{Q})_{i,j} = \boldsymbol{P}_{i,j} + \boldsymbol{Q}_{i,j} - \boldsymbol{P}_{i-1,j} - \boldsymbol{Q}_{i,j-1}, \forall i, j,$$

The adjoint operator of $\mathcal{L} : \mathbb{C}^{I\times J} \to \mathbb{C}^{(I-1)\times J} \times \mathbb{C}^{I\times(J-1)}$ is

$$\mathcal{L}^{\mathcal{T}}(\boldsymbol{X}) = (\boldsymbol{P}, \boldsymbol{Q}),$$

that $\boldsymbol{P}_{i,j} = \boldsymbol{X}_{i,j} - \boldsymbol{X}_{i+1,j}, \boldsymbol{Q}_{i,j} = \boldsymbol{X}_{i,j} - \boldsymbol{X}_{i,j+1}, \forall i, j$.

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

For complex x, y, we have

$$\sqrt{|x|^2+|y|^2} = \max_{p_1,p_2\in\mathbb{C}} \left\{\Re\left(p_1^* x + p_2^* y\right) : |p_1|^2 + |p_2|^2 \leq 1\right\}$$

$$|x| = \max_{p\in\mathbb{C}} \{\Re\left(p^* x\right) : |p| \leq 1\}$$

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

For complex x, y, we have

$$\sqrt{|x|^2 + |y|^2} = \max_{p_1, p_2 \in \mathbb{C}} \left\{ \Re\left(p_1^* x + p_2^* y\right) : |p_1|^2 + |p_2|^2 \leq 1 \right\}$$
$$|x| = \max_{p \in \mathbb{C}} \left\{ \Re\left(p^* x\right) : |p| \leq 1 \right\}$$

Then

$$\mathrm{TV}(\boldsymbol{x}) = \max_{(\boldsymbol{P}, \boldsymbol{Q}) \in \mathcal{P}} \Re\left\{ \mathrm{vec}\left(\mathcal{L}\left(\boldsymbol{P}, \boldsymbol{Q}\right)\right)^{\mathcal{H}} \boldsymbol{x} \right\},$$

$$\|\boldsymbol{T}\boldsymbol{x}\|_1 = \max_{\boldsymbol{z} \in \mathcal{Z}} \Re\left\{ \boldsymbol{z}^{\mathcal{H}} \boldsymbol{T}\boldsymbol{x} \right\}$$

$\mathcal{P}$, $\mathcal{Z}$: convex sets

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

Consider wavelet and TV: $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u} - \boldsymbol{x}\|_{\boldsymbol{W}}^2$

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \max_{\substack{\boldsymbol{z}\in\mathcal{Z} \\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \|\boldsymbol{x} - \boldsymbol{v}_k\|_{\boldsymbol{B}_k}^2 + 2\lambda g(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

$\boldsymbol{v}_k = \boldsymbol{x}_k - a_k \boldsymbol{B}_k^{-1} \nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$ and

$$g(\boldsymbol{x}, \boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q}) = \Re\left\{\alpha\langle \boldsymbol{T}\boldsymbol{x}, \boldsymbol{z}\rangle + (1-\alpha)\mathrm{vec}\left(\mathcal{L}\left(\boldsymbol{P}, \boldsymbol{Q}\right)\right)^{\mathcal{H}} \boldsymbol{x}\right\}$$

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

Consider wavelet and TV: $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x}) = \arg\min_{\boldsymbol{u}} R(\boldsymbol{u}) + \frac{1}{2}\|\boldsymbol{u}-\boldsymbol{x}\|_{\boldsymbol{W}}^2$

$$\min_{\boldsymbol{x}\in\mathbb{C}^N} \max_{\substack{\boldsymbol{z}\in\mathcal{Z} \\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \|\boldsymbol{x}-\boldsymbol{v}_k\|_{\boldsymbol{B}_k}^2 + 2\lambda g(\boldsymbol{x},\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})$$

$\boldsymbol{v}_k = \boldsymbol{x}_k - a_k \boldsymbol{B}_k^{-1}\nabla_{\boldsymbol{x}} f(\boldsymbol{x}_k)$ and

$$g(\boldsymbol{x},\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \Re\left\{\alpha\langle \boldsymbol{T}\boldsymbol{x},\boldsymbol{z}\rangle + (1-\alpha)\mathrm{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)^{\mathcal{H}}\boldsymbol{x}\right\}$$

$$\max_{\substack{\boldsymbol{z}\in\mathcal{Z}, \\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \min_{\boldsymbol{x}\in\mathbb{C}^N} \quad \|\boldsymbol{x}-\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 - \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2$$

$$\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \boldsymbol{v}_k - \lambda\boldsymbol{B}_k^{-1}\left(\alpha\boldsymbol{T}^{\mathcal{H}}\boldsymbol{z} + (1-\alpha)\mathrm{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)\right)$$

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

$$\max_{\substack{\boldsymbol{z}\in\mathcal{Z},\\(\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \min_{\boldsymbol{x}\in\mathbb{C}^N} \quad \|\boldsymbol{x}-\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 - \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2$$

$$\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \boldsymbol{v}_k - \lambda\boldsymbol{B}_k^{-1}\left(\alpha\boldsymbol{T}^{\mathcal{H}}\boldsymbol{z} + (1-\alpha)\mathrm{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)\right)$$

$$\boldsymbol{x}^* = \boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})$$

$$(\boldsymbol{z}^*,\boldsymbol{P}^*,\boldsymbol{Q}^*) = \operatorname*{argmin}_{\substack{\boldsymbol{z}\in\mathcal{Z},\\(\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \quad \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 .$$

Compute $\mathrm{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

$$\max_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \min_{\boldsymbol{x}\in\mathbb{C}^N} \quad \|\boldsymbol{x}-\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 - \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2$$

$$\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \boldsymbol{v}_k - \lambda\boldsymbol{B}_k^{-1}\left(\alpha\boldsymbol{T}^{\mathcal{H}}\boldsymbol{z} + (1-\alpha)\mathrm{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)\right)$$

$$\boldsymbol{x}^* = \boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})$$

$$(\boldsymbol{z}^*,\boldsymbol{P}^*,\boldsymbol{Q}^*) = \operatorname*{argmin}_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \quad \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 .$$

Gradient

$$-2\lambda\begin{bmatrix}\alpha\boldsymbol{T}\\ (1-\alpha)\mathcal{L}^{\mathcal{T}}\end{bmatrix}\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) \quad \text{APM, iter.}$$

Compute $\operatorname{Prox}_R^{\boldsymbol{W}}(\boldsymbol{x})$

$$\max_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \min_{\boldsymbol{x}\in\mathbb{C}^N} \quad \|\boldsymbol{x}-\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 - \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2$$

$$\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \boldsymbol{v}_k - \lambda\boldsymbol{B}_k^{-1}\left(\alpha\boldsymbol{T}^{\mathcal{H}}\boldsymbol{z} + (1-\alpha)\operatorname{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)\right)$$

$$\boldsymbol{x}^* = \boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})$$

$$(\boldsymbol{z}^*,\boldsymbol{P}^*,\boldsymbol{Q}^*) = \operatorname*{argmin}_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \quad \|\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 .$$

Gradient

$$-2\lambda\begin{bmatrix}\alpha\boldsymbol{T}\\ (1-\alpha)\mathcal{L}^{\mathcal{T}}\end{bmatrix}\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) \quad \text{APM, iter.}$$

But

$$\boldsymbol{w}_k(\boldsymbol{z},\boldsymbol{P},\boldsymbol{Q}) = \boldsymbol{v}_k - \lambda\boldsymbol{B}_k^{-1}\left(\alpha\boldsymbol{T}^{\mathcal{H}}\boldsymbol{z} + (1-\alpha)\operatorname{vec}\left(\mathcal{L}(\boldsymbol{P},\boldsymbol{Q})\right)\right)$$

Structure of $\boldsymbol{B}_k = \boldsymbol{D} + \sigma \boldsymbol{u}\boldsymbol{u}^{\mathcal{H}}$

Define $\sigma = 1/\left\langle \boldsymbol{m}_k - \boldsymbol{H}_0 \boldsymbol{s}_k, \boldsymbol{s}_k \right\rangle$ & $\boldsymbol{D} = \boldsymbol{H}_0$

Algorithm 1 SR1

Initialization: $\gamma > 1$, $\delta = 10^{-8}$, $\Xi > 0$ a fixed real scalar, $\boldsymbol{x}_k$, $\boldsymbol{x}_{k-1}$, $\nabla f(\boldsymbol{x}_k)$, and $\nabla f(\boldsymbol{x}_{k-1})$.

$\vdots$

Set $\boldsymbol{s}_k \leftarrow \boldsymbol{x}_k - \boldsymbol{x}_{k-1}$ and $\boldsymbol{m}_k \leftarrow \nabla f(\boldsymbol{x}_k) - \nabla f(\boldsymbol{x}_{k-1})$.

Compute $\tau_k \leftarrow \gamma \frac{\|\boldsymbol{m}_k\|_2^2}{\langle \boldsymbol{s}_k, \boldsymbol{m}_k \rangle}$. % $\langle \boldsymbol{a}, \boldsymbol{b} \rangle = \boldsymbol{b}^{\mathcal{H}} \boldsymbol{a}$

$\vdots$

$\boldsymbol{H}_0 \leftarrow \tau_k \boldsymbol{I}$.

$\boldsymbol{u}_k \leftarrow \boldsymbol{m}_k - \boldsymbol{H}_0 \boldsymbol{s}_k$.

$\boldsymbol{B}_k \leftarrow \boldsymbol{H}_0 + \frac{\boldsymbol{u}_k \boldsymbol{u}_k^{\mathcal{H}}}{\langle \boldsymbol{m}_k - \boldsymbol{H}_0 \boldsymbol{s}_k, \boldsymbol{s}_k \rangle}$.

Only Wavelet

$$(\boldsymbol{z}^*, \boldsymbol{P}^*, \boldsymbol{Q}^*) = \operatorname*{argmin}_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \quad \|\boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})\|^2_{\boldsymbol{B}_k}.$$

Gradient

$$-2\lambda \begin{bmatrix} 1\,\boldsymbol{T} \\ (1-1)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

Only Wavelet

$$(\boldsymbol{z}^*, \boldsymbol{P}^*, \boldsymbol{Q}^*) = \underset{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}}{\operatorname{argmin}} \quad \|\boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 .$$

Gradient

$$-2\lambda \begin{bmatrix} 1\,\boldsymbol{T} \\ (1-1)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

Solve $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$

Only Wavelet

$$(\boldsymbol{z}^*, \boldsymbol{P}^*, \boldsymbol{Q}^*) = \operatorname*{argmin}_{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}} \quad \|\boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})\|^2_{\boldsymbol{B}_k}.$$

Gradient

$$-2\lambda \begin{bmatrix} 1\,\boldsymbol{T} \\ (1-1)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

Solve $\min_{\boldsymbol{x}\in\mathbb{C}^M} \frac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$

Theorem (Becker19 SIAMOPT: real $\rightarrow$ complex)

Let $\boldsymbol{W} = \boldsymbol{D} \pm \boldsymbol{u}\boldsymbol{u}^{\mathcal{H}}$. *Then,*

$$\mathrm{Prox}_{\lambda R}^{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{Prox}_{\lambda R}^{\boldsymbol{D}}(\boldsymbol{x} \mp \boldsymbol{D}^{-1}\boldsymbol{u}\beta^*),$$

where $\beta^* \in \mathbb{C}$ *is the unique zero of the following nonlinear equation*

$$\mathbb{J}(\beta) : \boldsymbol{u}^{\mathcal{H}}\left(\boldsymbol{x} - \mathrm{Prox}_{\lambda R}^{\boldsymbol{D}}(\boldsymbol{x} \mp \boldsymbol{D}^{-1}\boldsymbol{u}\beta)\right) + \beta.$$

Only Wavelet

$$(\boldsymbol{z}^*, \boldsymbol{P}^*, \boldsymbol{Q}^*) = \underset{\substack{\boldsymbol{z}\in\mathcal{Z},\\ (\boldsymbol{P},\boldsymbol{Q})\in\mathcal{P}}}{\operatorname{argmin}} \quad \|\boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})\|_{\boldsymbol{B}_k}^2 .$$

Gradient

$$-2\lambda \begin{bmatrix} 1\,\boldsymbol{T} \\ (1-1)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

$$\text{Solve } \min_{\boldsymbol{x}\in\mathbb{C}^M} \tfrac{1}{2}\|\boldsymbol{A}\boldsymbol{T}^{-1}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\|\boldsymbol{x}\|_1$$

Theorem (Becker19 SIAMOPT: real $\to$ complex)

Let $\boldsymbol{W} = \boldsymbol{D} \pm \boldsymbol{u}\boldsymbol{u}^{\mathcal{H}}$*. Then,*

$$\mathrm{Prox}_{\lambda R}^{\boldsymbol{W}}(\boldsymbol{x}) = \mathrm{Prox}_{\lambda R}^{\boldsymbol{D}}(\boldsymbol{x} \mp \boldsymbol{D}^{-1}\boldsymbol{u}\beta^*),$$

where $\beta^* \in \mathbb{C}$ *is the unique zero of the following nonlinear equation*

$$\mathbb{J}(\beta) : \boldsymbol{u}^{\mathcal{H}}\left(\boldsymbol{x} - \mathrm{Prox}_{\lambda R}^{\boldsymbol{D}}(\boldsymbol{x} \mp \boldsymbol{D}^{-1}\boldsymbol{u}\beta)\right) + \beta.$$

$$\boldsymbol{B}_k = \boldsymbol{D} + \sigma\boldsymbol{u}\boldsymbol{u}^{\mathcal{H}} \text{ and } \sigma = 1/\langle \boldsymbol{m}_k - \boldsymbol{H}_0\boldsymbol{s}_k, \boldsymbol{s}_k\rangle \text{ is real}$$

What is More: TV + Wavelet?

Wavelet and TV:

Gradient

$$-2\lambda \begin{bmatrix} \alpha \boldsymbol{T} \\ (1-\alpha)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

What is More: TV + Wavelet?

Wavelet and TV:

Gradient

$$-2\lambda \begin{bmatrix} \alpha \boldsymbol{T} \\ (1-\alpha)\mathcal{L}^{\mathcal{T}} \end{bmatrix} \boldsymbol{w}_k(\boldsymbol{z}, \boldsymbol{P}, \boldsymbol{Q})$$

Partially smooth:

$$\min_{\boldsymbol{x} \in \mathbb{C}^N} \underbrace{\frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{y}\|_2^2 + \lambda\alpha \cdot \mathrm{S}^{\eta}\left(\|\boldsymbol{T}\boldsymbol{x}\|_1\right)}_{f(\boldsymbol{x})} + \underbrace{\lambda(1-\alpha)\mathrm{TV}(\boldsymbol{x})}_{R(\boldsymbol{x})}$$

$\mathrm{S}^{\eta}(\|\boldsymbol{x}\|_1) = \sum_{n=1}^{N} \sqrt{\boldsymbol{x}_n^2 + \eta}$

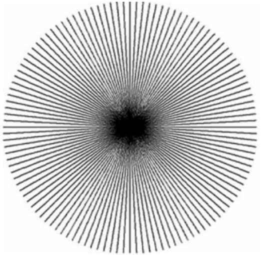
Experimental Settings

- Took k-space data from NYU fastMRI dataset
- Applied the ESPIRiT algorithm to recover the complex images
- Cropped the images to size 256×256 with maximum magnitude scaled to one
- Formulated the simulated k-space data with a given trajectory
- Added Gaussian noise with mean zero and variance 10^{-2} to all coils to form the final measurements

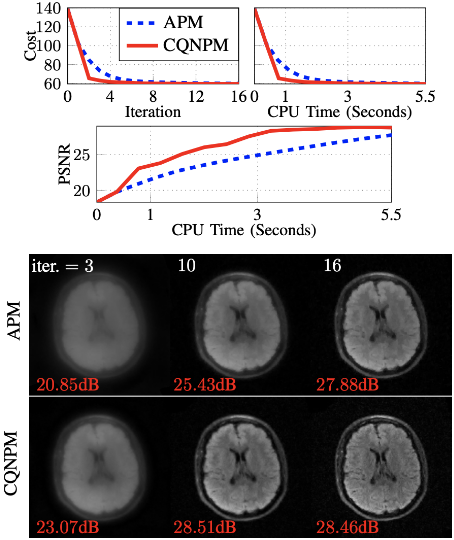
Experimental Settings

- Took k-space data from NYU fastMRI dataset
- Applied the ESPIRiT algorithm to recover the complex images
- Cropped the images to size 256×256 with maximum magnitude scaled to one
- Formulated the simulated k-space data with a given trajectory
- Added Gaussian noise with mean zero and variance 10^{-2} to all coils to form the final measurements

96 radial projections, 512 readout points, and 12 coils



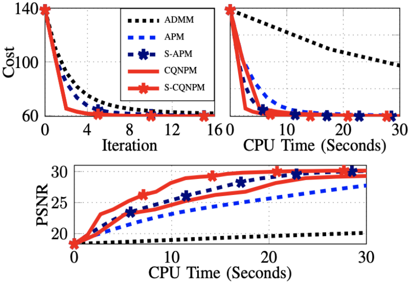
Wavelet

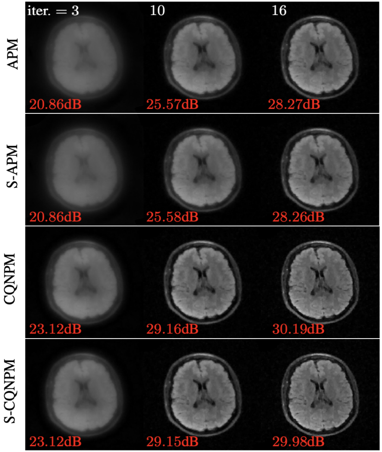


APM
CQNPM
Cost
140
120
100
80
60
0
4
8
12
16
Iteration
0
1
3
5.5
CPU Time (Seconds)
PSNR
25
20
0
1
3
5.5
CPU Time (Seconds)
iter. = 3
10
16
APM
20.85dB
25.43dB
27.88dB
CQNPM
23.07dB
28.51dB
28.46dB

Wavelet+TV

Ours: CQNPM & S-CQNPM





😀

😀 Thanks & Questions? 😀