Model-based X-ray CT image reconstruction using variable splitting methods with ordered subsets

Hung Nien (粘紘)

Advisor: Prof. Jeffrey A. Fessler

University of Michigan, Ann Arbor

May 19, 2014

Acknowledgements

- Supported in part by NIH grant R01–HL–098686
- Supported in part by an equipment donation from Intel Corp.
- Sinogram data are provided by GE Healthcare

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline of the thesis

1. Introduction
2. Background of X-ray CT and its reconstruction
3. Fast X-ray CT image reconstruction using VS methods with OS
   [HN & J A Fessler, Fully 3D, 2013]
   [HN & J A Fessler, SPIE MI, 2014]
   [HN & J A Fessler, CT Meeting, 2014]
   [HN & J A Fessler, arXiv:1402.4381, 2014]
4. Blind gain correction for X-ray CT image reconstruction
   [HN & J A Fessler, SPIE MI, 2013]
5. Model-based light field reconstruction
6. Conclusion and future work

Outline

1. Background
   - Model-based CT reconstruction
   - Fast (2D) CT reconstruction using ADMM
2. OS-LALM: a splitting-based OS algorithm for PWLS problems
   - Linearized AL method with OS acceleration
   - Deterministic downward continuation approach
   - Low-memory OS-LALM with additional variable splits
3. Experimental results
   - Low-dose CT with edge-preserving regularizers
   - Sparse-view CT with TV-like regularizers
4. Conclusion and future work

Outline

Outline

What is computed tomography?

What is computed tomography?

Projection radiography

An imaging technique that uses X-rays to view the internal structure of a non-uniformly composed and opaque object such as the human body.

What is computed tomography?

Projection radiography

An imaging technique that uses X-rays to view the internal structure of a non-uniformly composed and opaque object such as the human body.

Computed tomography

An imaging technique that combines a series of X-ray projections taken from many different angles and computer processing (i.e., reconstruction methods) to create cross-sectional images of the bones and soft tissues inside to the human body.

What is computed tomography?

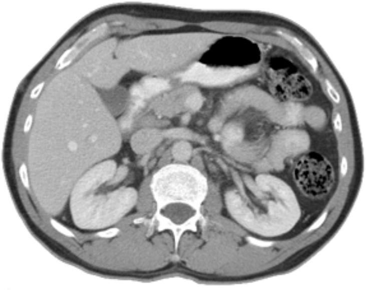
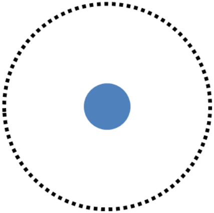


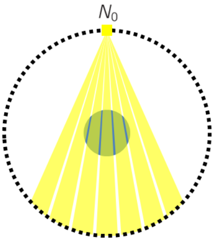
Figure: Chest X-ray image (left) and cross-sectional image of abdomen (right).

Basics of X-ray computed tomography

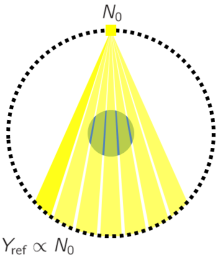
Basics of X-ray computed tomography



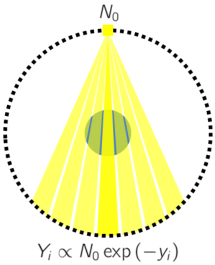
Basics of X-ray computed tomography



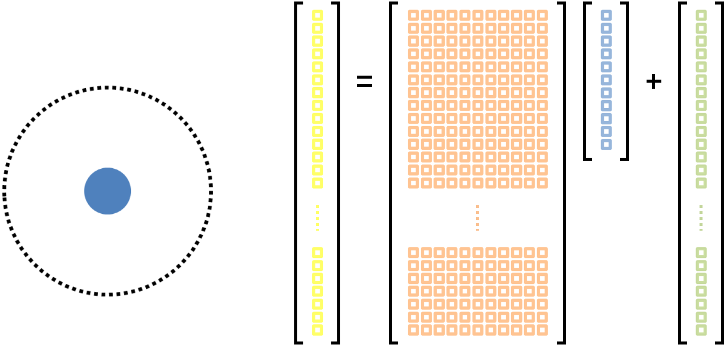
Basics of X-ray computed tomography



Basics of X-ray computed tomography

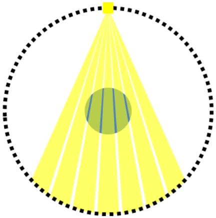


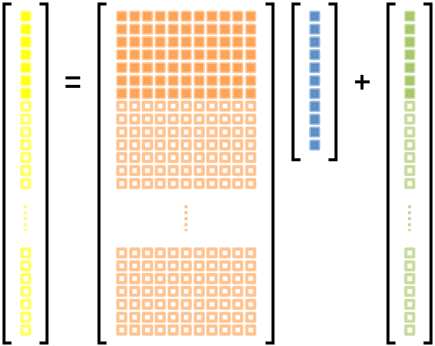
Basics of X-ray computed tomography



$$\mathbf{y} = \mathbf{A} \mathbf{x} + \varepsilon$$

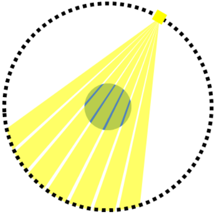
Basics of X-ray computed tomography

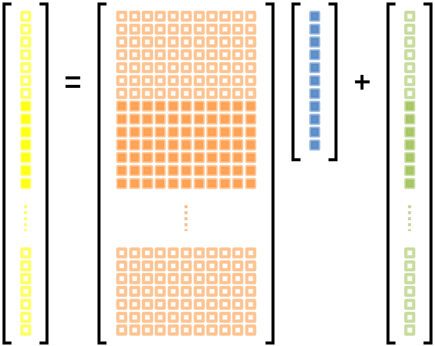




$$\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$$

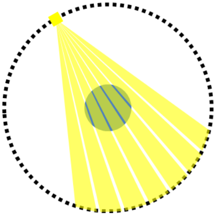
Basics of X-ray computed tomography

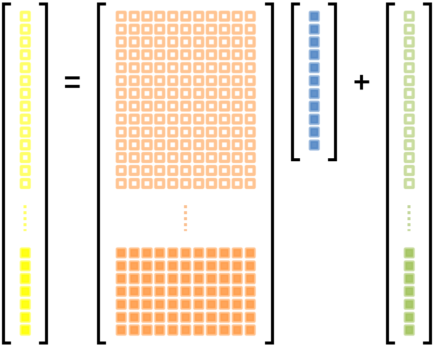




$$\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$$

Basics of X-ray computed tomography





$$\mathbf{y} = \mathbf{A}\mathbf{x} + \varepsilon$$

Image reconstruction methods

Non-iterative methods

Iterative methods

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction

Iterative methods

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method

Iterative methods

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Iterative methods

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Iterative methods

- Maximum *a posteriori* (MAP) formulation

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Iterative methods

- Maximum *a posteriori* (MAP) formulation
- Penalized weighted least-squares (PWLS) formulation [TS+07]

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \mathrm{R}(\mathbf{x}) \right\}$$

Image reconstruction methods

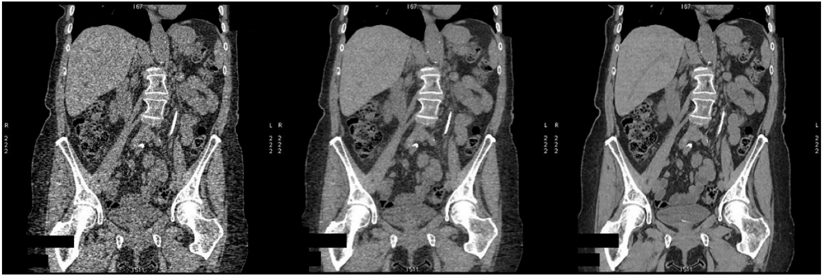


Figure: Dose reduction: FBP (left), ASiR (middle), and MBIR (right).

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Iterative methods

- Maximum *a posteriori* (MAP) formulation
- Penalized weighted least-squares (PWLS) formulation [TS+07]

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \mathrm{R}(\mathbf{x}) \right\}$$

Image reconstruction methods

Large problem size

- $\mathbf{x}$: $512 \times 512 \times 100 \approx 3 \cdot 10^7$ unknown image volume
- $\mathbf{y}$: $888 \times 32 \times 7000 \approx 2 \cdot 10^8$ measured noisy sinogram
- $\mathbf{A}$: $\left(3 \cdot 10^7\right) \times \left(2 \cdot 10^8\right)$ system matrix
- $\mathbf{A}$ is sparse but still too large to store
- Projection $\mathbf{Ax}$ and back-projection $\mathbf{A}'\mathbf{r}$ operations computed on the fly
- Computing gradient $\nabla\Psi(\mathbf{x}) = \mathbf{A}'\mathbf{W}\left(\mathbf{Ax} - \mathbf{y}\right) + \nabla \mathrm{R}(\mathbf{x})$ requires projection and back-projection operations that dominate computation

Image reconstruction methods

Large problem size

- $\mathbf{x}$: $512 \times 512 \times 100 \approx 3 \cdot 10^7$ unknown image volume
- $\mathbf{y}$: $888 \times 32 \times 7000 \approx 2 \cdot 10^8$ measured noisy sinogram
- $\mathbf{A}$: $\left(3 \cdot 10^7\right) \times \left(2 \cdot 10^8\right)$ system matrix
- $\mathbf{A}$ is sparse but still too large to store
- Projection $\mathbf{Ax}$ and back-projection $\mathbf{A}'\mathbf{r}$ operations computed on the fly
- Computing gradient $\nabla\Psi(\mathbf{x}) = \mathbf{A}'\mathbf{W}\left(\mathbf{Ax} - \mathbf{y}\right) + \nabla R(\mathbf{x})$ requires projection and back-projection operations that dominate computation

Enormous dynamic range of transmission data

- The dynamic range of weighting $\mathbf{W}$ is huge
- $\mathbf{A}'\mathbf{WA}$ is highly shift-variant, and the problem is very ill-conditioned

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Iterative methods

- Maximum *a posteriori* (MAP) formulation
- Penalized weighted least-squares (PWLS) formulation [TS+07]

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \mathrm{R}(\mathbf{x}) \right\}$$

- Very slow (hours) but noise robust (low dose)

Image reconstruction methods

Non-iterative methods

- Direct Fourier reconstruction
- Filter-backproject (FBP) method
- Very fast (seconds) but prone to noise (medium/high dose)

Fast iterative methods

- Maximum *a posteriori* (MAP) formulation
- Penalized weighted least-squares (PWLS) formulation [TS+07]

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \Psi(\mathbf{x}) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \mathsf{R}(\mathbf{x}) \right\}$$

- **Fast (minutes)** and noise robust (low dose)

Outline

1. Background
   - Model-based CT reconstruction
   - Fast (2D) CT reconstruction using ADMM
2. OS-LALM: a splitting-based OS algorithm for PWLS problems
   - Linearized AL method with OS acceleration
   - Deterministic downward continuation approach
   - Low-memory OS-LALM with additional variable splits
3. Experimental results
   - Low-dose CT with edge-preserving regularizers
   - Sparse-view CT with TV-like regularizers
4. Conclusion and future work

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method

- PWLS CT reconstruction: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{C}\mathbf{x}) \right\}$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method

- PWLS CT reconstruction: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{C}\mathbf{x}) \right\}$
- Equivalent formulation [GO09]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{v}}) \in \arg\min_{\mathbf{x},\mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{v} = \mathbf{C}\mathbf{x}$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method

- PWLS CT reconstruction: $\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{C}\mathbf{x}) \right\}$
- Equivalent formulation [GO09]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{v}}) \in \arg\min_{\mathbf{x},\mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{v} = \mathbf{C}\mathbf{x}$$

- Corresponding (scaled) augmented Lagrangian:

$$\mathcal{L}_{\mathrm{A}}(\mathbf{x}, \mathbf{v}, \mathbf{e}; \eta) \triangleq \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) + \frac{\eta}{2} \|\mathbf{C}\mathbf{x} - \mathbf{v} - \mathbf{e}\|_2^2$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\|\mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\|\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)}\right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Convergent with inexact updates [NF14a]

CT reconstruction using split Bregman method

Equivalent formulation and split Bregman method (cont'd)

- Split Bregman iterates [GO09]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{A}\mathbf{x}\|_{\mathbf{W}}^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Convergent with inexact updates [NF14a]
- Slow $\mathbf{x}$-update due to the highly shift-variant Hessian $\mathbf{A}'\mathbf{W}\mathbf{A} + \eta\mathbf{C}'\mathbf{C}$

Better conditioning with additional VS

The idea is ...

Better conditioning with additional VS

The idea is ...

- In 2D CT, $\mathbf{A}'\mathbf{W}\mathbf{A}$ is highly shift-variant, but $\mathbf{A}'\mathbf{A}$ is not

Better conditioning with additional VS

The idea is ...

- In 2D CT, $\mathbf{A}'\mathbf{W}\mathbf{A}$ is highly shift-variant, but $\mathbf{A}'\mathbf{A}$ is not
- Replacing the weighted quadratic function in the $\mathbf{x}$-update with an unweighted one removes most shift-variances of the Hessian

Better conditioning with additional VS

The idea is ...

- In 2D CT, $\mathbf{A}'\mathbf{W}\mathbf{A}$ is highly shift-variant, but $\mathbf{A}'\mathbf{A}$ is not
- Replacing the weighted quadratic function in the $\mathbf{x}$-update with an unweighted one removes most shift-variances of the Hessian

Alternative formulation and ADMM

- Alternative formulation [RF12]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg\min_{\mathbf{x},\mathbf{u},\mathbf{v}} \left\{ \frac{1}{2} \|\mathbf{y} - \mathbf{u}\|_{\mathbf{W}}^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{u} = \mathbf{A}\mathbf{x}, \mathbf{v} = \mathbf{C}\mathbf{x}$$

Better conditioning with additional VS

Alternative formulation and ADMM (cont'd)

- ADMM iterates [RF12]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{u} \right\|_{\mathbf{W}}^2 + \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

Better conditioning with additional VS

Alternative formulation and ADMM (cont'd)

- ADMM iterates [RF12]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{u} \right\|_{\mathbf{W}}^2 + \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

Better conditioning with additional VS

Alternative formulation and ADMM (cont'd)

- ADMM iterates [RF12]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{u} \right\|_{\mathbf{W}}^2 + \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

Better conditioning with additional VS

Alternative formulation and ADMM (cont'd)

- ADMM iterates [RF12]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{u} \right\|_{\mathbf{W}}^2 + \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Convergent with inexact updates [AB+11]

Better conditioning with additional VS

Alternative formulation and ADMM (cont'd)

- ADMM iterates [RF12]:

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{y} - \mathbf{u} \right\|_{\mathbf{W}}^2 + \frac{\rho}{2} \left\| \mathbf{A}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)} \right\|_2^2 \right\} \\ \mathbf{v}^{(k+1)} \in \arg\min_{\mathbf{v}} \left\{ \Phi(\mathbf{v}) + \frac{\eta}{2} \left\| \mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{v} - \mathbf{e}^{(k)} \right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \mathbf{A}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Convergent with inexact updates [AB+11]
- $\rho \mathbf{A}'\mathbf{A} + \eta \mathbf{C}'\mathbf{C}$ can be well preconditioned by an appropriate circulant preconditioner in 2D CT

Better conditioning with additional VS

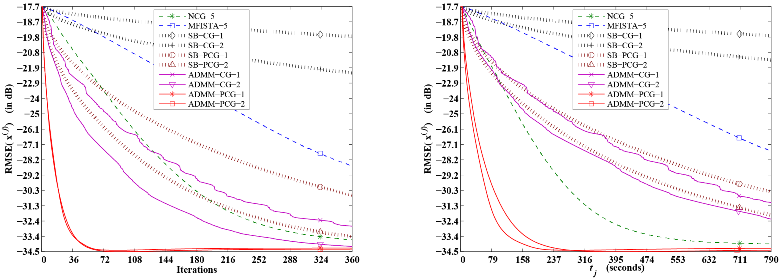


Figure: 2D NCAT: RMS errors as a function of iteration (left) and time (right).

Better conditioning with additional VS

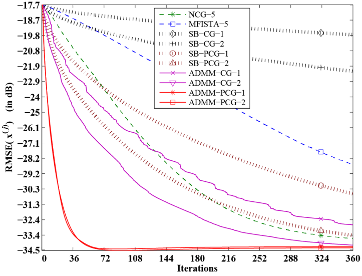


Figure: 2D NCAT: RMS errors as a function of iteration (left) and time (right).

The fact is ...

$\mathbf{A}'\mathbf{A}$ is still highly shift-variant in 3D CT due to the different geometries and scan trajectories, so this method is still slow in 3D CT

Outline

Outline

1. Background
   - Model-based CT reconstruction
   - Fast (2D) CT reconstruction using ADMM
2. OS-LALM: a splitting-based OS algorithm for PWLS problems
   - Linearized AL method with OS acceleration
   - Deterministic downward continuation approach
   - Low-memory OS-LALM with additional variable splits
3. Experimental results
   - Low-dose CT with edge-preserving regularizers
   - Sparse-view CT with TV-like regularizers
4. Conclusion and future work

Motivation

What's wrong with ADMM in CT recon?

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high
- Ordered-subsets (OS) acceleration is not applicable

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high
- Ordered-subsets (OS) acceleration is not applicable

Possible solution?

- Proposed formulation [NF13]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg\min_{\mathbf{x}\in\Omega,\mathbf{u}} \left\{ \frac{1}{2} \left\| \mathbf{W}^{1/2}\mathbf{y} - \mathbf{u} \right\|_2^2 + \mathrm{R}(\mathbf{x}) \right\} \text{ s.t. } \mathbf{u} = \mathbf{W}^{1/2}\mathbf{A}\mathbf{x}$$

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high
- Ordered-subsets (OS) acceleration is not applicable

Possible solution?

- Proposed formulation [NF13]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg\min_{\mathbf{x},\mathbf{u}} \left\{ \frac{1}{2} \|\tilde{\mathbf{y}} - \mathbf{u}\|_2^2 + h(\mathbf{x}) \right\} \text{ s.t. } \mathbf{u} = \tilde{\mathbf{A}}\mathbf{x}$$

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high
- Ordered-subsets (OS) acceleration is not applicable

Possible solution?

- Proposed formulation [NF13]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg\min_{\mathbf{x},\mathbf{u}} \left\{ \frac{1}{2} \|\tilde{\mathbf{y}} - \mathbf{u}\|_2^2 + h(\mathbf{x}) \right\} \text{ s.t. } \mathbf{u} = \tilde{\mathbf{A}}\mathbf{x}$$

- Image update:

$$\mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho}{2} \left\| \tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2 \right\}$$

Motivation

What's wrong with ADMM in CT recon?

- Image update is non-trivial
- Memory burden of difference images is high
- Ordered-subsets (OS) acceleration is not applicable

Possible solution?

- Proposed formulation [NF13]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}) \in \arg\min_{\mathbf{x},\mathbf{u}} \left\{ \frac{1}{2} \|\tilde{\mathbf{y}} - \mathbf{u}\|_2^2 + h(\mathbf{x}) \right\} \text{ s.t. } \mathbf{u} = \tilde{\mathbf{A}}\mathbf{x}$$

- Image update:

$$\mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \boxed{\frac{\rho}{2} \left\| \tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)} \right\|_2^2} \right\}$$

Inexact linearized AL method

The idea is ...

Linearized AL method and proposed variants

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates

Linearized AL method and proposed variants

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates

Linearized AL method and proposed variants

- Linearized AL method

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \breve{\theta}_k\left(\mathbf{x}; \mathbf{x}^{(k)}\right) \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}\left\|\tilde{\mathbf{y}} - \mathbf{u}\right\|_2^2 + \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \tilde{\mathbf{A}}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \end{cases}$$

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates

Linearized AL method and proposed variants

- Linearized AL method

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \breve{\theta}_k\left(\mathbf{x}; \mathbf{x}^{(k)}\right) \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}\left\|\tilde{\mathbf{y}} - \mathbf{u}\right\|_2^2 + \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \tilde{\mathbf{A}}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \end{cases}$$

$$\boxed{\breve{\theta}_k\left(\mathbf{x}; \mathbf{x}^{(k)}\right) \triangleq \theta_k\left(\mathbf{x}^{(k)}\right) + \left\langle \nabla\theta_k\left(\mathbf{x}^{(k)}\right), \mathbf{x} - \mathbf{x}^{(k)} \right\rangle + \frac{\rho L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_2^2}$$

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates

Linearized AL method and proposed variants

- Linearized AL method

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho L}{2}\left\|\mathbf{x} - \left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\nabla\theta_k\left(\mathbf{x}^{(k)}\right)\right)\right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}\left\|\tilde{\mathbf{y}} - \mathbf{u}\right\|_2^2 + \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \tilde{\mathbf{A}}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \end{cases}$$

$$\boxed{\breve{\theta}_k\left(\mathbf{x};\mathbf{x}^{(k)}\right) \triangleq \theta_k\left(\mathbf{x}^{(k)}\right) + \left\langle \nabla\theta_k\left(\mathbf{x}^{(k)}\right), \mathbf{x} - \mathbf{x}^{(k)}\right\rangle + \frac{\rho L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_2^2}$$

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates
- Quadratic data-fitting term makes the **u**-updates linear

Linearized AL method and proposed variants

- Linearized AL method

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho L}{2}\left\|\mathbf{x} - \left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\nabla\theta_k\left(\mathbf{x}^{(k)}\right)\right)\right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} \in \arg\min_{\mathbf{u}} \left\{ \frac{1}{2}\left\|\tilde{\mathbf{y}} - \mathbf{u}\right\|_2^2 + \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x}^{(k+1)} - \mathbf{u} - \mathbf{d}^{(k)}\right\|_2^2 \right\} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \tilde{\mathbf{A}}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \end{cases}$$

$$\boxed{\breve{\theta}_k\left(\mathbf{x};\mathbf{x}^{(k)}\right) \triangleq \theta_k\left(\mathbf{x}^{(k)}\right) + \left\langle \nabla\theta_k\left(\mathbf{x}^{(k)}\right), \mathbf{x} - \mathbf{x}^{(k)} \right\rangle + \frac{\rho L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_2^2}$$

Inexact linearized AL method

The idea is ...

- Majorizing $\theta_k(\mathbf{x}) \triangleq \frac{\rho}{2}\left\|\tilde{\mathbf{A}}\mathbf{x} - \mathbf{u}^{(k)} - \mathbf{d}^{(k)}\right\|_2^2$ simplifies image updates
- Quadratic data-fitting term makes the **u**-updates linear

Linearized AL method and proposed variants

- Linearized AL method

$$\begin{cases} \mathbf{x}^{(k+1)} \in \arg\min_{\mathbf{x}} \left\{ h(\mathbf{x}) + \frac{\rho L}{2}\left\|\mathbf{x} - \left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\nabla\theta_k\left(\mathbf{x}^{(k)}\right)\right)\right\|_2^2 \right\} \\ \mathbf{u}^{(k+1)} = \frac{\rho}{\rho+1}\left(\tilde{\mathbf{A}}\mathbf{x}^{(k+1)} - \mathbf{d}^{(k)}\right) + \frac{1}{\rho+1}\tilde{\mathbf{y}} \\ \mathbf{d}^{(k+1)} = \mathbf{d}^{(k)} - \tilde{\mathbf{A}}\mathbf{x}^{(k+1)} + \mathbf{u}^{(k+1)} \end{cases}$$

$$\boxed{\breve{\theta}_k\left(\mathbf{x};\mathbf{x}^{(k)}\right) \triangleq \theta_k\left(\mathbf{x}^{(k)}\right) + \left\langle \nabla\theta_k\left(\mathbf{x}^{(k)}\right), \mathbf{x} - \mathbf{x}^{(k)}\right\rangle + \frac{\rho L}{2}\left\|\mathbf{x} - \mathbf{x}^{(k)}\right\|_2^2}$$

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\Big(\mathbf{x}^{(k)} - \big(\rho^{-1}t\big)\,\mathbf{s}^{(k+1)}\Big) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla \ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\Big(\mathbf{x}^{(k)} - \big(\rho^{-1}t\big)\,\mathbf{s}^{(k+1)}\Big) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla \ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- Linearized AL method with OS acceleration (M subsets) [NF14b]:

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M \nabla \ell_m\left(\mathbf{x}^{(k,m)}\right) + (1-\rho)\,\mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k,m)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k,m+1)}\right) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1} M \nabla \ell_{m+1}\left(\mathbf{x}^{(k,m+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k,m)} \end{cases}$$

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- Linearized AL method with OS acceleration (M subsets) [NF14b]:

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M \nabla \ell_m\left(\mathbf{x}^{(k,m)}\right) + (1-\rho)\,\mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k,m)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k,m+1)}\right) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1} M \nabla \ell_{m+1}\left(\mathbf{x}^{(k,m+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k,m)} \end{cases}$$

- Inexact updates? Convergence rate? Many subsets? [NF14d]

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho\nabla\ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla\ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- Linearized AL method with OS acceleration (M subsets) [NF14b]:

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M\nabla\ell_m\left(\mathbf{x}^{(k,m)}\right) + (1-\rho)\,\mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k,m)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k,m+1)}\right) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1}M\nabla\ell_{m+1}\left(\mathbf{x}^{(k,m+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k,m)} \end{cases}$$

- Inexact updates? Convergence rate? Many subsets? [NF14d]

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho\nabla\ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla\ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- Linearized AL method with OS acceleration (M subsets) [NF14b]:

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M\nabla\ell_m\left(\mathbf{x}^{(k,m)}\right) + (1-\rho)\,\mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k,m)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k,m+1)}\right) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1}M\nabla\ell_{m+1}\left(\mathbf{x}^{(k,m+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k,m)} \end{cases}$$

- Inexact updates? Convergence rate? Many subsets? [NF14d]

Inexact linearized AL method

Linearized AL method and proposed variants (cont'd)

- Gradient-based linearized AL method [NF14b]:

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho\nabla\ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla\ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- Linearized AL method with OS acceleration (M subsets) [NF14b]:

$$\begin{cases} \mathbf{s}^{(k,m+1)} = \rho M\nabla\ell_m\left(\mathbf{x}^{(k,m)}\right) + (1-\rho)\,\mathbf{g}^{(k,m)} \\ \mathbf{x}^{(k,m+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k,m)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k,m+1)}\right) \\ \mathbf{g}^{(k,m+1)} = \frac{\rho}{\rho+1}M\nabla\ell_{m+1}\left(\mathbf{x}^{(k,m+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k,m)} \end{cases}$$

- Inexact updates? Convergence rate? Many subsets? [NF14d]

Outline

Faster convergence using continuation

An inconvenient truth ... of ADMM

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

- Yes, continuation!

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

- Yes, continuation!
- Recall the proximal-gradient image update:

$$\mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right)$$

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

- Yes, continuation!
- Recall the proximal-gradient image update:

$$\mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right)$$

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

- Yes, continuation!
- Recall the proximal-gradient image update:

$$\mathbf{x}^{(k+1)} \in \mathrm{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right)$$

- We can adjust the step size by varying the penalty parameter

Faster convergence using continuation

An inconvenient truth ... of ADMM

- ADMM is convergent for any fixed penalty parameter
- But, it is fast only if the penalty parameter is chosen appropriately
- Choosing the optimal parameter is still an open problem

Any alternative?

- Yes, continuation!
- Recall the proximal-gradient image update:

$$\mathbf{x}^{(k+1)} \in \operatorname{prox}_{(\rho^{-1}t)h}\left(\mathbf{x}^{(k)} - \left(\rho^{-1}t\right)\mathbf{s}^{(k+1)}\right)$$

- We can adjust the step size by varying the penalty parameter (How?)

Faster convergence using continuation

Deterministic downward continuation

Faster convergence using continuation

Deterministic downward continuation

- Decreasing ρ_k compensates the shrinkage of step length

Faster convergence using continuation

Deterministic downward continuation

- Decreasing ρ_k compensates the shrinkage of step length
- Decreasing ρ_k too fast could make the algorithm unstable or diverge

Faster convergence using continuation

Deterministic downward continuation

- Decreasing ρ_k compensates the shrinkage of step length
- Decreasing ρ_k too fast could make the algorithm unstable or diverge
- The designed sequence [NF14d]:

$$\rho_k = \begin{cases} 1 & \text{, if } k = 0 \\ \frac{\pi}{k+1}\sqrt{1 - \left(\frac{\pi}{2k+2}\right)^2} & \text{, otherwise} \end{cases}$$

Faster convergence using continuation

Deterministic downward continuation

- Decreasing ρ_k compensates the shrinkage of step length
- Decreasing ρ_k too fast could make the algorithm unstable or diverge
- The designed sequence [NF14d]:

$$\rho_k = \begin{cases} 1 & \text{, if } k = 0 \\ \frac{\pi}{k+1}\sqrt{1 - \left(\frac{\pi}{2k+2}\right)^2} & \text{, otherwise} \end{cases}$$

- Inspired by a second-order recursive system analysis Derivation

Faster convergence using continuation

Deterministic downward continuation

- Decreasing ρ_k compensates the shrinkage of step length
- Decreasing ρ_k too fast could make the algorithm unstable or diverge
- The designed sequence [NF14d]:

$$\rho_k = \begin{cases} 1 & \text{, if } k = 0 \\ \frac{\pi}{k+1}\sqrt{1 - \left(\frac{\pi}{2k+2}\right)^2} & \text{, otherwise} \end{cases}$$

- Inspired by a second-order recursive system analysis Derivation
- An adaptive restart condition takes care of the dependence on $\tilde{\mathbf{A}}$

Outline

OS-LALM with an additional VS

Is additional VS really beneficial?

OS-LALM with an additional VS

Is additional VS really beneficial?

- ℓ_1-regularization (compressed sensing)

OS-LALM with an additional VS

Is additional VS really beneficial?

- ℓ_1-regularization (compressed sensing)
- TV-regularization (sparse-view CT)

OS-LALM with an additional VS

Is additional VS really beneficial?

- ℓ_1-regularization (compressed sensing)
- TV-regularization (sparse-view CT)
- Smooth regularizer with very high curvature (corner-rounding)

OS-LALM with an additional VS

Is additional VS really beneficial?

- ℓ_1-regularization (compressed sensing)
- TV-regularization (sparse-view CT)
- Smooth regularizer with very high curvature (corner-rounding)

CT reconstruction with "high-memory" VS

- PWLS formulation:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \frac{1}{2} \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{A}}\mathbf{x} \right\|_2^2 + \Phi(\mathbf{C}\mathbf{x}) \right\}$$

OS-LALM with an additional VS

Is additional VS really beneficial?

- ℓ_1-regularization (compressed sensing)
- TV-regularization (sparse-view CT)
- Smooth regularizer with very high curvature (corner-rounding)

CT reconstruction with “high-memory” VS

- PWLS formulation:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}\in\Omega} \left\{ \frac{1}{2} \left\| \tilde{\mathbf{y}} - \tilde{\mathbf{A}}\mathbf{x} \right\|_2^2 + \Phi(\mathbf{C}\mathbf{x}) \right\}$$

- Equivalent formulation [NF14c]:

$$(\hat{\mathbf{x}}, \hat{\mathbf{u}}, \hat{\mathbf{v}}) \in \arg\min_{\mathbf{x}\in\Omega,\mathbf{u},\mathbf{v}} \left\{ \frac{1}{2} \left\| \tilde{\mathbf{y}} - \mathbf{u} \right\|_2^2 + \Phi(\mathbf{v}) \right\} \text{ s.t. } \mathbf{u} = \tilde{\mathbf{A}}\mathbf{x}, \mathbf{v} = \mathbf{C}\mathbf{x}$$

OS-LALM with an additional VS

"High-memory" OS-LALM

$$
\begin{cases}
\mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\
\boldsymbol{\sigma}^{(k+1)} = \bar{\eta}\,\mathbf{C}'\left(\mathbf{C}\mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\right) \\
\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \bar{\eta} L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_{\Omega} \\
\mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\
\mathbf{v}^{(k+1)} \in \operatorname{prox}_{\bar{\eta}^{-1}\Phi}\left(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}\right) \\
\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}
\end{cases}
$$

OS-LALM with an additional VS

"High-memory" OS-LALM

$$
\begin{cases}
\mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\
\boldsymbol{\sigma}^{(k+1)} = \bar{\eta}\,\mathbf{C}'\left(\mathbf{C}\mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\right) \\
\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \bar{\eta} L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_{\Omega} \\
\mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\
\mathbf{v}^{(k+1)} \in \operatorname{prox}_{\bar{\eta}^{-1}\Phi}\left(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}\right) \\
\mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)}
\end{cases}
$$

- Gradient descent-like algorithm with adjustable step sizes

OS-LALM with an additional VS

"High-memory" OS-LALM

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho\nabla\ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\ \boldsymbol{\sigma}^{(k+1)} = \bar{\eta}\,\mathbf{C}'\big(\mathbf{C}\mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\big) \\ \mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \bar{\eta} L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_\Omega \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla\ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\ \mathbf{v}^{(k+1)} \in \operatorname{prox}_{\bar{\eta}^{-1}\Phi}\big(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}\big) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Gradient descent-like algorithm with adjustable step sizes
- Φ can be either smooth or non-smooth (with efficient prox_Φ)

OS-LALM with an additional VS

"High-memory" OS-LALM

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \boldsymbol{\sigma}^{(k+1)} = \bar{\eta}\,\mathbf{C}'\left(\mathbf{C}\mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\right) \\ \mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \bar{\eta} L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_{\Omega} \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\ \mathbf{v}^{(k+1)} \in \operatorname{prox}_{\bar{\eta}^{-1}\Phi}\left(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}\right) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Gradient descent-like algorithm with adjustable step sizes
- Φ can be either smooth or non-smooth (with efficient $\operatorname{prox}_{\Phi}$)
- Requires two extra image volumes for each direction

OS-LALM with an additional VS

"High-memory" OS-LALM

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\ \boldsymbol{\sigma}^{(k+1)} = \bar{\eta}\,\mathbf{C}'\big(\mathbf{C}\mathbf{x}^{(k)} - \mathbf{v}^{(k)} - \mathbf{e}^{(k)}\big) \\ \mathbf{x}^{(k+1)} = \Big[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \bar{\eta} L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\Big]_{\Omega} \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\ \mathbf{v}^{(k+1)} \in \operatorname{prox}_{\bar{\eta}^{-1}\Phi}\big(\mathbf{C}\mathbf{x}^{(k+1)} - \mathbf{e}^{(k)}\big) \\ \mathbf{e}^{(k+1)} = \mathbf{e}^{(k)} - \mathbf{C}\mathbf{x}^{(k+1)} + \mathbf{v}^{(k+1)} \end{cases}$$

- Gradient descent-like algorithm with adjustable step sizes
- Φ can be either smooth or non-smooth (with efficient $\operatorname{prox}_{\Phi}$)
- Requires two extra image volumes for each direction
- Remarkable memory and computational overhead

Low-memory OS-LALM with “compressed” VS

The idea is ...

Low-memory OS-LALM with “compressed” VS

The idea is ...

- The auxiliary variables $\mathbf{v}$ and $\mathbf{e}$ can be large, but $\mathbf{C}'\mathbf{v}$ and $\mathbf{C}'\mathbf{e}$ are not

Low-memory OS-LALM with "compressed" VS

The idea is ...

- The auxiliary variables $\mathbf{v}$ and $\mathbf{e}$ can be large, but $\mathbf{C}'\mathbf{v}$ and $\mathbf{C}'\mathbf{e}$ are not
- Majoirzing Φ makes all $\mathbf{v}$- and $\mathbf{e}$-related updates linear!

Low-memory OS-LALM with “compressed” VS

The idea is ...

- The auxiliary variables $\mathbf{v}$ and $\mathbf{e}$ can be large, but $\mathbf{C}'\mathbf{v}$ and $\mathbf{C}'\mathbf{e}$ are not
- Majoirzing Φ makes all $\mathbf{v}$- and $\mathbf{e}$-related updates linear!

“Low-memory” OS-LALM

- At the kth iteration, replace Φ by

$$\breve{\Phi}\left(\mathbf{v}; \mathbf{C}\mathbf{x}^{(k+1)}\right) \propto \mathbf{v}'\nabla\Phi\left(\mathbf{C}\mathbf{x}^{(k+1)}\right) + \tfrac{L_{\Phi}}{2}\left\|\mathbf{v} - \mathbf{C}\mathbf{x}^{(k+1)}\right\|_2^2$$

Low-memory OS-LALM with “compressed” VS

The idea is ...

- The auxiliary variables $\mathbf{v}$ and $\mathbf{e}$ can be large, but $\mathbf{C}'\mathbf{v}$ and $\mathbf{C}'\mathbf{e}$ are not
- Majoirzing Φ makes all $\mathbf{v}$- and $\mathbf{e}$-related updates linear!

“Low-memory” OS-LALM

- At the kth iteration, replace Φ by

$$\breve{\Phi}\left(\mathbf{v}; \mathbf{C}\mathbf{x}^{(k+1)}\right) \propto \mathbf{v}'\nabla\Phi\left(\mathbf{C}\mathbf{x}^{(k+1)}\right) + \tfrac{L_\Phi}{2}\left\|\mathbf{v} - \mathbf{C}\mathbf{x}^{(k+1)}\right\|_2^2$$

- The $\mathbf{v}$-update has a linear approximate solution:

$$\mathbf{v}^{(k+1)} \approx \mathbf{C}\mathbf{x}^{(k+1)} - \left(\tfrac{\bar{\eta}}{\bar{\eta}+L_\Phi}\mathbf{e}^{(k)} + \tfrac{L_\Phi}{\bar{\eta}+L_\Phi} L_\Phi^{-1}\nabla\Phi\left(\mathbf{C}x^{(k+1)}\right)\right)$$

Low-memory OS-LALM with "compressed" VS

"Low-memory" OS-LALM (cont'd)

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \boldsymbol{\sigma}^{(k+1)} = \eta\left(\bar{\mathbf{v}}^{(k)} - \tilde{\mathbf{e}}^{(k)}\right) \\ \mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \eta L_\Phi L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_\Omega \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\ \bar{\mathbf{v}}^{(k+1)} = \frac{\eta}{\eta+1}\tilde{\mathbf{e}}^{(k)} + \frac{1}{\eta+1}\nabla \mathrm{R}\left(\mathbf{x}^{(k+1)}\right) \\ \tilde{\mathbf{e}}^{(k+1)} = \tilde{\mathbf{e}}^{(k)} - \bar{\mathbf{v}}^{(k+1)} \end{cases}$$

Low-memory OS-LALM with "compressed" VS

"Low-memory" OS-LALM (cont'd)

$$
\begin{cases}
\mathbf{s}^{(k+1)} = \rho \nabla \ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\
\boldsymbol{\sigma}^{(k+1)} = \eta\left(\bar{\mathbf{v}}^{(k)} - \tilde{\mathbf{e}}^{(k)}\right) \\
\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \eta L_\Phi L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_\Omega \\
\mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1} \nabla \ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\
\bar{\mathbf{v}}^{(k+1)} = \frac{\eta}{\eta+1}\tilde{\mathbf{e}}^{(k)} + \frac{1}{\eta+1}\nabla \mathrm{R}\big(\mathbf{x}^{(k+1)}\big) \\
\tilde{\mathbf{e}}^{(k+1)} = \tilde{\mathbf{e}}^{(k)} - \bar{\mathbf{v}}^{(k+1)}
\end{cases}
$$

- Suppose Φ is smooth, and $\nabla\Phi$ is L_Φ-Lipschitz

Low-memory OS-LALM with "compressed" VS

"Low-memory" OS-LALM (cont'd)

$$\begin{cases} \mathbf{s}^{(k+1)} = \rho \nabla \ell\left(\mathbf{x}^{(k)}\right) + (1-\rho)\,\mathbf{g}^{(k)} \\ \boldsymbol{\sigma}^{(k+1)} = \eta\left(\bar{\mathbf{v}}^{(k)} - \tilde{\mathbf{e}}^{(k)}\right) \\ \mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \eta L_\Phi L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_\Omega \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla \ell\left(\mathbf{x}^{(k+1)}\right) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\ \bar{\mathbf{v}}^{(k+1)} = \frac{\eta}{\eta+1}\tilde{\mathbf{e}}^{(k)} + \frac{1}{\eta+1}\nabla \mathrm{R}\left(\mathbf{x}^{(k+1)}\right) \\ \tilde{\mathbf{e}}^{(k+1)} = \tilde{\mathbf{e}}^{(k)} - \bar{\mathbf{v}}^{(k+1)} \end{cases}$$

- Suppose Φ is smooth, and $\nabla\Phi$ is L_Φ-Lipschitz
- No explicit proximal mapping is in the updates

Low-memory OS-LALM with "compressed" VS

"Low-memory" OS-LALM (cont'd)

$$
\begin{cases}
\mathbf{s}^{(k+1)} = \rho \nabla \ell\big(\mathbf{x}^{(k)}\big) + (1-\rho)\,\mathbf{g}^{(k)} \\
\boldsymbol{\sigma}^{(k+1)} = \eta\left(\bar{\mathbf{v}}^{(k)} - \tilde{\mathbf{e}}^{(k)}\right) \\
\mathbf{x}^{(k+1)} = \left[\mathbf{x}^{(k)} - \frac{1}{\rho L_1 + \eta L_\Phi L_2}\left(\mathbf{s}^{(k+1)} + \boldsymbol{\sigma}^{(k+1)}\right)\right]_\Omega \\
\mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\nabla \ell\big(\mathbf{x}^{(k+1)}\big) + \frac{1}{\rho+1}\mathbf{g}^{(k)} \\
\bar{\mathbf{v}}^{(k+1)} = \frac{\eta}{\eta+1}\tilde{\mathbf{e}}^{(k)} + \frac{1}{\eta+1}\nabla \mathsf{R}\big(\mathbf{x}^{(k+1)}\big) \\
\tilde{\mathbf{e}}^{(k+1)} = \tilde{\mathbf{e}}^{(k)} - \bar{\mathbf{v}}^{(k+1)}
\end{cases}
$$

- Suppose Φ is smooth, and $\nabla\Phi$ is L_Φ-Lipschitz
- No explicit proximal mapping is in the updates
- Requires only two extra image volumes for all directions

Outline

1. Background
   - Model-based CT reconstruction
   - Fast (2D) CT reconstruction using ADMM
2. OS-LALM: a splitting-based OS algorithm for PWLS problems
   - Linearized AL method with OS acceleration
   - Deterministic downward continuation approach
   - Low-memory OS-LALM with additional variable splits
3. Experimental results
   - Low-dose CT with edge-preserving regularizers
   - Sparse-view CT with TV-like regularizers
4. Conclusion and future work

Setup and notation

Setup and notation

OS configuration

- OS with the bit-reversal order
- Separable quadratic surrogate with Hessian $\mathbf{D}_{\mathsf{L}} \triangleq \mathrm{diag}\{\mathbf{A}'\mathbf{W}\mathbf{A}\mathbf{1}\}$

Setup and notation

OS configuration

- OS with the bit-reversal order
- Separable quadratic surrogate with Hessian $\mathbf{D}_{\mathrm{L}} \triangleq \mathrm{diag}\{\mathbf{A}'\mathbf{W}\mathbf{A}\mathbf{1}\}$

Naming conventions

- OS-SQS-M: the standard OS algorithm [EF99]
- OS-Nes05-M: the state-of-the-art OS+momentum algorithm [KR+13]
- OS-LALM-M-ρ-n: the proposed one-split algorithm [NF14b; NF14d]
- OS-LALM-M-c-η (low-mem): the proposed two-split algorithm

Outline

Shoulder region helical scan

Specification

- Image size: $512 \times 512 \times 109$
- Sinogram size: $888 \times 32 \times 7146$ (about 7 turns, pitch $= 0.5$)

Shoulder region helical scan

Specification

- Image size: $512 \times 512 \times 109$
- Sinogram size: $888 \times 32 \times 7146$ (about 7 turns, pitch $= 0.5$)

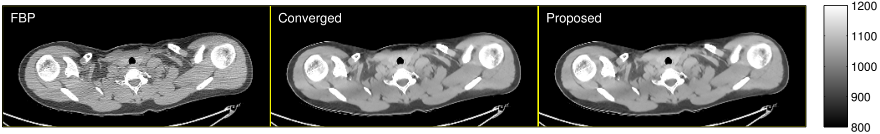


Figure: Shoulder: the initial FBP image (left), the reference reconstruction (middle), and the reconstructed image using OS-LALM after 30 iterations (right).

Shoulder region helical scan

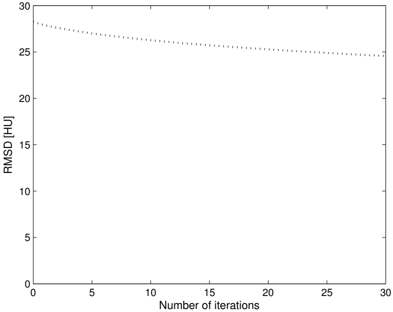


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

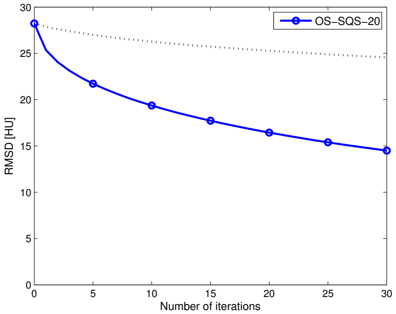


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

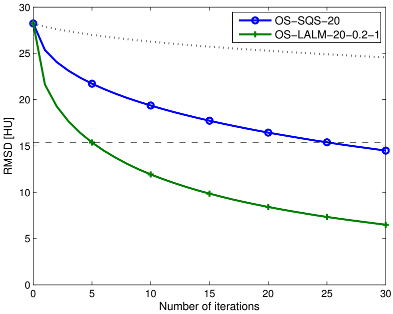


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

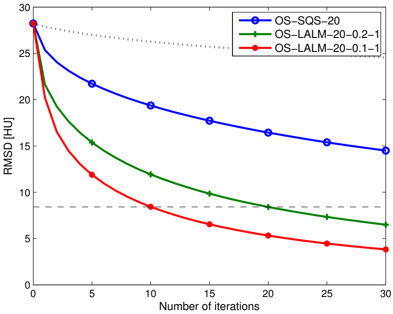


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

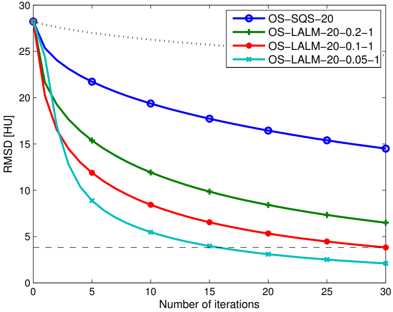


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

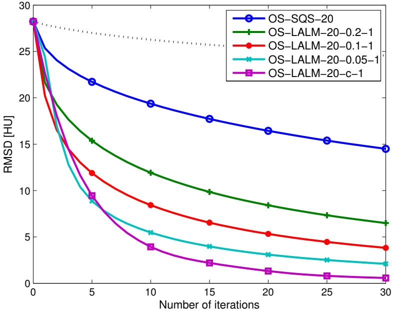


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

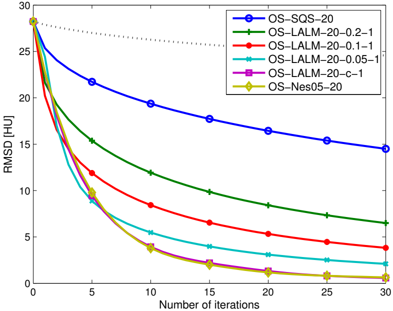


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

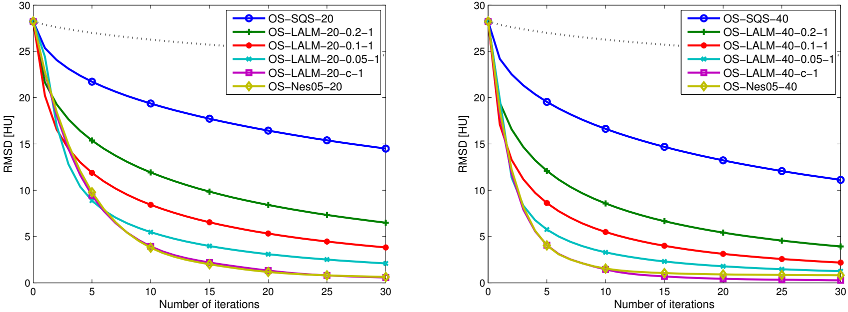


Figure: Shoulder: RMS differences as a function of iteration using different OS-based algorithms with 20 subsets (left) and 40 subsets (right).

Shoulder region helical scan

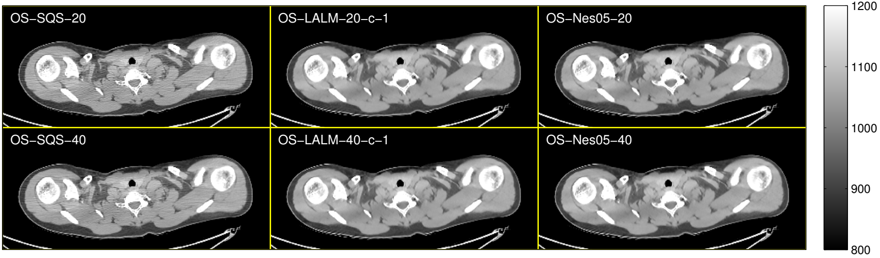


Figure: Shoulder: reconstructed images using different OS-based algorithms after 30 iterations.

Shoulder region helical scan

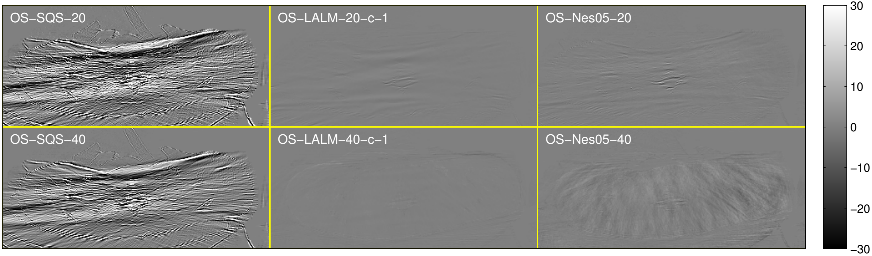


Figure: Shoulder: difference images of the reconstructed images using different OS-based algorithms after 30 iterations.

GE performance phantom axial scan

Specification

- Image size: $1024 \times 1024 \times 90$
- Sinogram size: $888 \times 64 \times 984$ (less view redundancy, cf. helical scan)

GE performance phantom axial scan

Specification

- Image size: $1024 \times 1024 \times 90$
- Sinogram size: $888 \times 64 \times 984$ (less view redundancy, cf. helical scan)

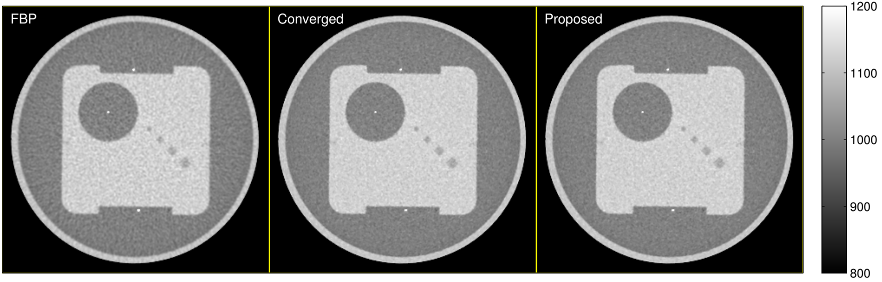


Figure: GEPP: the initial FBP image (left), the reference reconstruction (middle), and the reconstructed image using OS-LALM after 30 iterations (right).

GE performance phantom axial scan

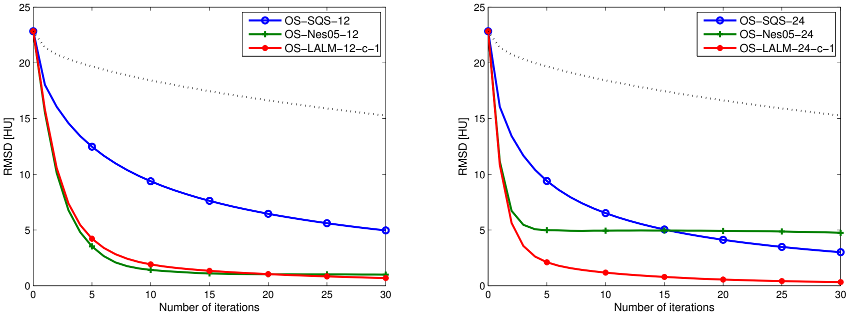


Figure: GEPP: RMS differences as a function of iteration using different OS-based algorithms with 12 subsets (left) and 24 subsets (right).

GE performance phantom axial scan

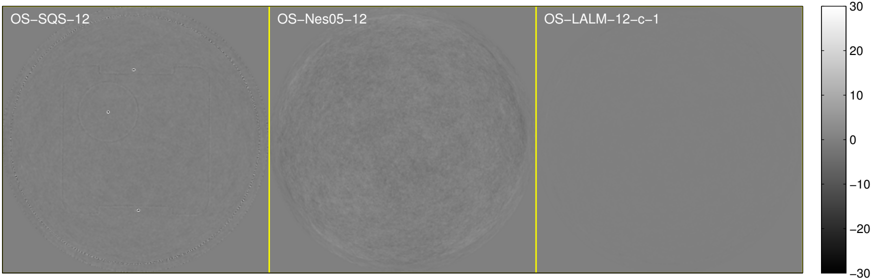


Figure: GEPP: difference images of the reconstructed images using different OS-based algorithms with 12 subsets after 30 iterations.

GE performance phantom axial scan

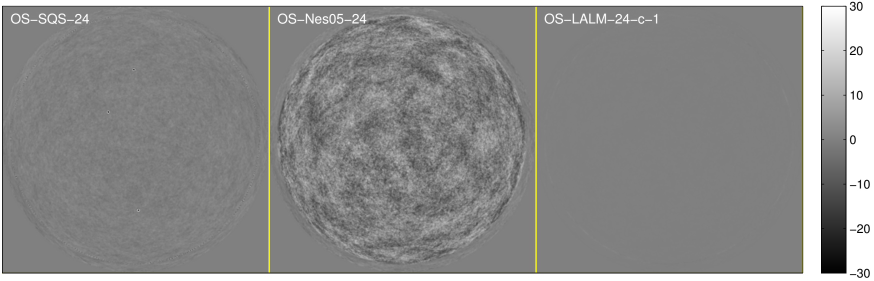


Figure: GEPP: difference images of the reconstructed images using different OS-based algorithms with 24 subsets after 30 iterations.

Outline

Chest region half scan

Specification

- Image size: $718 \times 718 \times 122$
- Sinogram size: $888 \times 64 \times 81$ (about 12.6% views are used)

Chest region half scan

Specification

- Image size: $718 \times 718 \times 122$
- Sinogram size: $888 \times 64 \times 81$ (about 12.6% views are used)

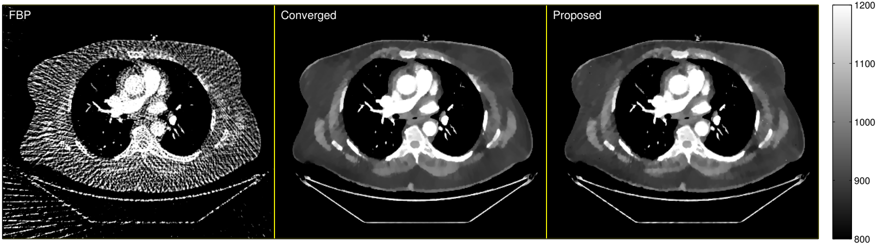


Figure: Chest: the initial FBP image (left), the reference reconstruction (middle), and the reconstructed image using OS-LALM after 100 iterations (right).

Chest region half scan

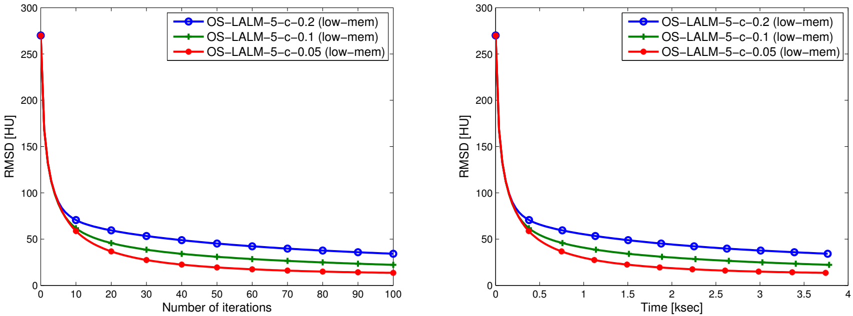


Figure: Chest: RMS differences as a function of iteration (left) and time (right) using OS-LALM with $M = 5$ and different values of η.

Chest region half scan

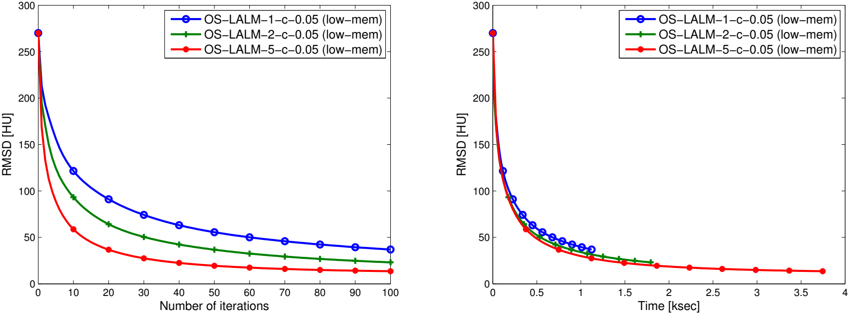


Figure: Chest: RMS differences as a function of iteration (left) and time (right) using OS-LALM with different values of M and $\eta = 0.05$.

Chest region half scan

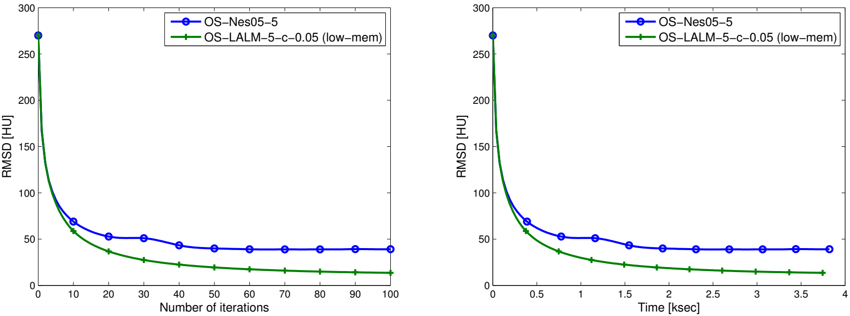


Figure: Chest: RMS differences as a function of iteration (left) and time (right) using different OS-based algorithms with $M = 5$.

Chest region half scan

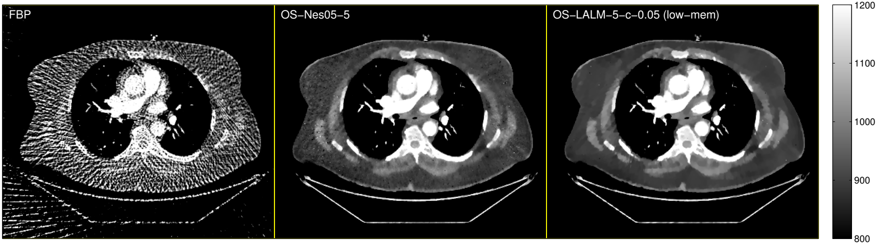


Figure: Chest: reconstructed images using different OS-based algorithms with $M = 5$ after 100 iterations.

Outline

Conclusion

Conclusion

We proposed ...

- A splitting-based OS algorithm, OS-LALM, for solving PWLS X-ray CT image reconstruction problems
- A deterministic downward continuation approach for accelerating the proposed algorithm
- A low-memory variant of the proposed algorithm when considering additional variable splits

Conclusion

We proposed ...

- A splitting-based OS algorithm, OS-LALM, for solving PWLS X-ray CT image reconstruction problems
- A deterministic downward continuation approach for accelerating the proposed algorithm
- A low-memory variant of the proposed algorithm when considering additional variable splits

Experimental results showed that ...

- The proposed algorithm significantly accelerates the convergence of X-ray CT image reconstruction with negligible overhead
- The proposed algorithm is stable when using many subsets for OS acceleration

Future work

Future work

Theory

- Convergence analysis of OS-LALM when $M > 1$
- Convergence analysis of OS-LALM with downward continuation
- Optimal downward continuation and restart condition
- Convergence analysis of low-mem OS-LALM
- Parameter selection for low-mem OS-LALM

Future work

Theory

- Convergence analysis of OS-LALM when $M > 1$
- Convergence analysis of OS-LALM with downward continuation
- Optimal downward continuation and restart condition
- Convergence analysis of low-mem OS-LALM
- Parameter selection for low-mem OS-LALM

Extension

- Non-quadratic data-fitting term
- Low-mem OS-LALM with non-smooth potential functions

References I

[AB$^+$11] M. V. Afonso, J. M. Bioucas-Dias, and M. A. T. Figueiredo. "An augmented Lagrangian approach to the constrained optimization formulation of imaging inverse problems". In: *IEEE Trans. Im. Proc.* 20.3 (Mar. 2011), pp. 681–95.

[EF99] H. Erdoğan and J. A. Fessler. "Ordered subsets algorithms for transmission tomography". In: *Phys. Med. Biol.* 44.11 (Nov. 1999), pp. 2835–51.

[GO09] T. Goldstein and S. Osher. "The split Bregman method for L1-regularized problems". In: *SIAM J. Imaging Sci.* 2.2 (2009), pp. 323–43.

[KR$^+$13] D. Kim, S. Ramani, and J. A. Fessler. "Accelerating X-ray CT ordered subsets image reconstruction with Nesterov's first-order methods". In: *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.* 2013, pp. 22–5.

References II

[NF13] H. Nien and J. A. Fessler. "Combining augmented Lagrangian method with ordered subsets for X-ray CT reconstruction". In: *Proc. Intl. Mtg. on Fully 3D Image Recon. in Rad. and Nuc. Med.* 2013, pp. 280–3.

[NF14a] H. Nien and J. A. Fessler. "A convergence proof of the split Bregman method for regularized least-squares problems". In: *arXiv: 1402.4371* (2014).

[NF14b] H. Nien and J. A. Fessler. "Accelerating ordered-subsets X-ray CT image reconstruction using the linearized augmented Lagrangian framework". In: *Proc. SPIE 9033 Medical Imaging 2014: Phys. Med. Im.* To appear. 2014.

[NF14c] H. Nien and J. A. Fessler. "Fast splitting-based ordered-subsets X-ray CT image reconstruction". In: *Proc. 3rd Intl. Mtg. on image formation in X-ray CT*. Submitted. 2014.

References III

[NF14d] H. Nien and J. A. Fessler. “Fast X-ray CT image reconstruction using the linearized augmented Lagrangian method with ordered subsets”. In: *arXiv: 1402.4381* (2014).

[RF12] S. Ramani and J. A. Fessler. “A splitting-based iterative algorithm for accelerated statistical X-ray CT reconstruction”. In: *IEEE Trans. Med. Imag.* 31.3 (Mar. 2012), pp. 677–88.

[TS+07] J-B. Thibault et al. “A three-dimensional statistical approach to improved image quality for multi-slice helical CT”. In: *Med. Phys.* 34.11 (Nov. 2007), pp. 4526–44.

THANK YOU!

any question?

Second-order recursive system analysis

Minimize a quadratic function

Second-order recursive system analysis

Minimize a quadratic function

- Simple quadratic programming:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x}\|_2^2$$

Second-order recursive system analysis

Minimize a quadratic function

- Simple quadratic programming:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x}\|_2^2$$

- Let $\mathbf{V}\mathbf{\Lambda}\mathbf{V}'$ be the EVD of $\mathbf{A}'\mathbf{A}$, where $0 < \mu = \lambda_1 \leq \ldots \leq \lambda_n = L$

Second-order recursive system analysis

Minimize a quadratic function

- Simple quadratic programming:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \frac{1}{2} \|\mathbf{A}\mathbf{x}\|_2^2$$

- Let $\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'$ be the EVD of $\mathbf{A}'\mathbf{A}$, where $0 < \mu = \lambda_1 \leq \ldots \leq \lambda_n = L$
- LALM iterates (with a fixed ρ):

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (1/L)\left(\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'\mathbf{x}^{(k)} + \left(\rho^{-1} - 1\right)\mathbf{g}^{(k)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{V}\boldsymbol{\Lambda}\mathbf{V}'\mathbf{x}^{(k+1)} + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

Second-order recursive system analysis

Minimize a quadratic function

- Simple quadratic programming:

$$\hat{\mathbf{x}} \in \arg\min_{\mathbf{x}} \tfrac{1}{2} \|\mathbf{A}\mathbf{x}\|_2^2$$

- Let $\mathbf{V}\mathbf{\Lambda}\mathbf{V}'$ be the EVD of $\mathbf{A}'\mathbf{A}$, where $0 < \mu = \lambda_1 \leq \ldots \leq \lambda_n = L$
- LALM iterates (with a fixed ρ):

$$\begin{cases} \mathbf{x}^{(k+1)} = \mathbf{x}^{(k)} - (1/L)\left(\mathbf{V}\mathbf{\Lambda}\mathbf{V}'\mathbf{x}^{(k)} + \left(\rho^{-1} - 1\right)\mathbf{g}^{(k)}\right) \\ \mathbf{g}^{(k+1)} = \frac{\rho}{\rho+1}\mathbf{V}\mathbf{\Lambda}\mathbf{V}'\mathbf{x}^{(k+1)} + \frac{1}{\rho+1}\mathbf{g}^{(k)} \end{cases}$$

- The diagonalized system $\bar{\mathbf{x}} \triangleq \mathbf{V}'\mathbf{x}$ and $\bar{\mathbf{g}} \triangleq \mathbf{V}'\mathbf{g}$ satisfies a 2nd-order recursive system determined by the characteristic polynomial:

$$(1+\rho)\, r^2 - 2\left(1 - \lambda_i/L + \rho/2\right) r + \left(1 - \lambda_i/L\right)$$

Second-order recursive system analysis

System behaviors based on the value of ρ

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1-\frac{\lambda_i}{L}\right)}$$

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1 - \frac{\lambda_i}{L}\right)}$$

- $\rho = \rho_i^\star$: critically damped

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1 - \frac{\lambda_i}{L}\right)}$$

- $\rho = \rho_i^\star$: critically damped
- $\rho > \rho_i^\star$: over-damped

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1-\frac{\lambda_i}{L}\right)}$$

- $\rho = \rho_i^\star$: critically damped
- $\rho > \rho_i^\star$: over-damped
- $\rho < \rho_i^\star$: under-damped, oscillates with frequency $\psi_i \approx \sqrt{\lambda_i/L}$

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1-\frac{\lambda_i}{L}\right)}$$

- $\rho = \rho_i^\star$: critically damped
- $\rho > \rho_i^\star$: over-damped
- $\rho < \rho_i^\star$: under-damped, oscillates with frequency $\psi_i \approx \sqrt{\lambda_i / L}$

We also observed that

- In practice, the asymp. convergence rate of the system is determined by the eigencomponent with the smallest eigenvalue $\lambda_1 = \mu$

Second-order recursive system analysis

System behaviors based on the value of ρ

Let

$$\rho_i^\star \triangleq 2\sqrt{\frac{\lambda_i}{L}\left(1-\frac{\lambda_i}{L}\right)}$$

- $\rho = \rho_i^\star$: critically damped
- $\rho > \rho_i^\star$: over-damped
- $\rho < \rho_i^\star$: under-damped, oscillates with frequency $\psi_i \approx \sqrt{\lambda_i/L}$

We also observed that

- In practice, the asymp. convergence rate of the system is determined by the eigencomponent with the smallest eigenvalue $\lambda_1 = \mu$
- For $\lambda_i < L/2$, the critically damped system converges fastest

Second-order recursive system analysis

System behaviors based on the value of ρ (cont'd)

To attain the fastest asymp. convergence rate, we would like to choose

$$\rho^\star = \rho_1^\star = 2\sqrt{\frac{\mu}{L}\left(1 - \frac{\mu}{L}\right)}$$

Second-order recursive system analysis

System behaviors based on the value of ρ (cont'd)

To attain the fastest asymp. convergence rate, we would like to choose

$$\rho^\star = \rho_1^\star = 2\sqrt{\frac{\mu}{L}\left(1 - \frac{\mu}{L}\right)}$$

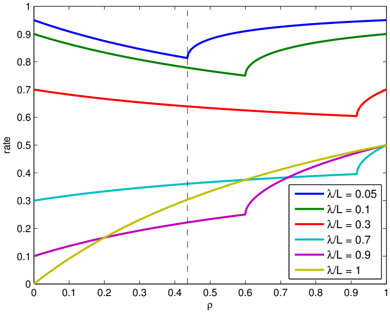


Figure: Asymptotic convergence rate of a system with 6 distinct eigenvalues.

Second-order recursive system analysis

Design a decreasing sequence $\rho_k \to \rho^\star$

Second-order recursive system analysis

Design a decreasing sequence $\rho_k \to \rho^\star$

We know that

- As the algorithm proceeds, only the component oscillating at the freqency $\psi_1 \approx \sqrt{\mu/L}$ persists

Second-order recursive system analysis

Design a decreasing sequence $\rho_k \to \rho^\star$

We know that

- As the algorithm proceeds, only the component oscillating at the freqency $\psi_1 \approx \sqrt{\mu/L}$ persists
- In this case, $\xi(k) \triangleq \left(\mathbf{g}^{(k)} - \nabla\ell\left(\mathbf{x}^{(k+1)}\right)\right)' \left(\nabla\ell\left(\mathbf{x}^{(k+1)}\right) - \nabla\ell\left(\mathbf{x}^{(k)}\right)\right)$ oscillates at the frequency $2\sqrt{\mu/L}$

Second-order recursive system analysis

Design a decreasing sequence $\rho_k \to \rho^\star$

We know that

- As the algorithm proceeds, only the component oscillating at the freqency $\psi_1 \approx \sqrt{\mu/L}$ persists
- In this case, $\xi(k) \triangleq \left(\mathbf{g}^{(k)} - \nabla\ell\left(\mathbf{x}^{(k+1)}\right)\right)' \left(\nabla\ell\left(\mathbf{x}^{(k+1)}\right) - \nabla\ell\left(\mathbf{x}^{(k)}\right)\right)$ oscillates at the frequency $2\sqrt{\mu/L}$
- If the algorithm is restarted when $\xi(k) > 0$, we shall observe the next restart signal after a further $(\pi/2)\sqrt{L/\mu}$ iterations

Second-order recursive system analysis

Design a decreasing sequence $\rho_k \to \rho^\star$

We know that

- As the algorithm proceeds, only the component oscillating at the freqency $\psi_1 \approx \sqrt{\mu/L}$ persists
- In this case, $\xi(k) \triangleq \left(\mathbf{g}^{(k)} - \nabla\ell\left(\mathbf{x}^{(k+1)}\right)\right)' \left(\nabla\ell\left(\mathbf{x}^{(k+1)}\right) - \nabla\ell\left(\mathbf{x}^{(k)}\right)\right)$ oscillates at the frequency $2\sqrt{\mu/L}$
- If the algorithm is restarted when $\xi(k) > 0$, we shall observe the next restart signal after a further $(\pi/2)\sqrt{L/\mu}$ iterations
- We can easily design a decreasing sequence ρ_k that reaches $\rho^\star$ every time we restart the algorithm! Back