# The Proximal-proximal Gradient Algorithm

Ting Kei Pong
PIMS Postdoctoral Fellow
Department of Computer Science
University of British Columbia
Vancouver

SINBAD Consortium Meeting, Fall
Dec 2013

# Analysis vs Synthesis

$$\begin{aligned} \min_{m} \quad & \frac{1}{2}\|\mathcal{A}m - b\|^2 \\ \text{s.t.} \quad & \|\mathcal{C}m\|_1 \leq \tau. \end{aligned}$$

- $\mathcal{A}$ is a linear operator (RTM), $\mathcal{C}$ is curvelet operator.
- Various applications: full waveform inversion, RTM imaging...

# Analysis vs Synthesis

$$\begin{aligned} \min_{m} \quad & \frac{1}{2}\|\mathcal{A}m - b\|^2 \\ \text{s.t.} \quad & \|\mathcal{C}m\|_1 \leq \tau. \end{aligned}$$

- $\mathcal{A}$ is a linear operator (RTM), $\mathcal{C}$ is curvelet operator.
- Various applications: full waveform inversion, RTM imaging...
- Currently used subproblem in modified Gauss-Newton full waveform inversion (Li '12)

$$\begin{aligned} \min_{x} \quad & \frac{1}{2}\|\mathcal{A}\mathcal{C}^*x - b\|^2 \\ \text{s.t.} \quad & \|x\|_1 \leq \tau, \end{aligned}$$

  solved using SPGL1 (van den Berg, Friedlander '08).

# Proximal gradient algorithm

Building block algorithm for SPGL1.

# Proximal gradient algorithm

Building block algorithm for SPGL1.

Set $L \geq \lambda_{\max}(\mathcal{A}^*\mathcal{A})$. For $t = 0, 1, 2, \ldots$, update

$$m^{t+1} = \underset{\|\mathcal{C}m\|_1 \leq \tau}{\arg\min} \left\{ \left\| m - \left( m^t - \frac{1}{L}\mathcal{A}^*(\mathcal{A}m^t - b) \right) \right\|^2 \right\}.$$

# Proximal gradient algorithm

Building block algorithm for SPGL1.

Set $L \geq \lambda_{\max}(\mathcal{A}^*\mathcal{A})$. For $t = 0, 1, 2, \ldots$, update

$$m^{t+1} = \operatorname*{arg\,min}_{\|\mathcal{C}m\|_1 \leq \tau} \left\{ \left\| m - \left( m^t - \frac{1}{L}\mathcal{A}^*(\mathcal{A}m^t - b) \right) \right\|^2 \right\}.$$

If $\mathcal{C}$ were absent, this would be projection onto $\ell_1$ norm ball — easy: implemented in SPGL1

# Proximal gradient algorithm

Building block algorithm for SPGL1.

Set $L \geq \lambda_{\max}(\mathcal{A}^*\mathcal{A})$. For $t = 0, 1, 2, \ldots$, update

$$m^{t+1} = \underset{\|\mathcal{C}m\|_1 \leq \tau}{\arg\min} \left\{ \left\| m - \left( m^t - \frac{1}{L}\mathcal{A}^*(\mathcal{A}m^t - b) \right) \right\|^2 \right\}.$$

If $\mathcal{C}$ were absent, this would be projection onto $\ell_1$ norm ball — easy: implemented in SPGL1

Otherwise, no simple formula for the update!

# Inexact Proximal Gradient

One solution: solve the subproblem iteratively — introduce inner loop.

# Inexact Proximal Gradient

One solution: solve the subproblem iteratively — introduce inner loop.

- Initialize $m^0$, $y^0$. Set $\beta = \frac{1}{L}$.
- For $t = 0, 1, 2, \ldots$
  - ⋆ For $s = 0, 1, 2, \ldots$, starting with $u^0 = y^t$, (warm start)

$$w^s = u^s + \beta^{-1}\mathcal{C}\left[m^t - \beta\mathcal{A}^*(\mathcal{A}m^t - b) - \beta\mathcal{C}^*u^s\right].$$

$$u^{s+1} = w^s - \beta^{-1}\mathrm{Proj}_{\|\cdot\|_1 \le \tau}\left(\beta w^s\right).$$

  - ⋆ Get approximate solution $y^{t+1} = u^{s+1}$.

  Update $m^{t+1} = m^t - \beta\mathcal{A}^*(\mathcal{A}m^t - b) - \beta\mathcal{C}^*y^{t+1}$.

# Proximal-proximal Gradient Algorithm

- Initialize $m^0$, $y^0$. Set $\beta \in (0, \frac{2}{L})$, $\gamma \in (0, 1 + \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\})$.

- For $t = 0, 1, 2, \ldots$

$$\begin{cases} w^t & = y^t + \beta^{-1}\mathcal{C}\left[m^t - \beta\mathcal{A}^*(\mathcal{A}m^t - b) - \beta\mathcal{C}^* y^t\right], \\ y^{t+1} & = w^t - \beta^{-1}\text{Proj}_{\|\cdot\|_1 \leq \tau}\left(\beta w^t\right), \\ m^{t+1} & = m^t - \gamma\beta\mathcal{A}^*(\mathcal{A}m^t - b) - \gamma\beta\mathcal{C}^* y^{t+1}. \end{cases}$$

# Proximal-proximal Gradient Algorithm

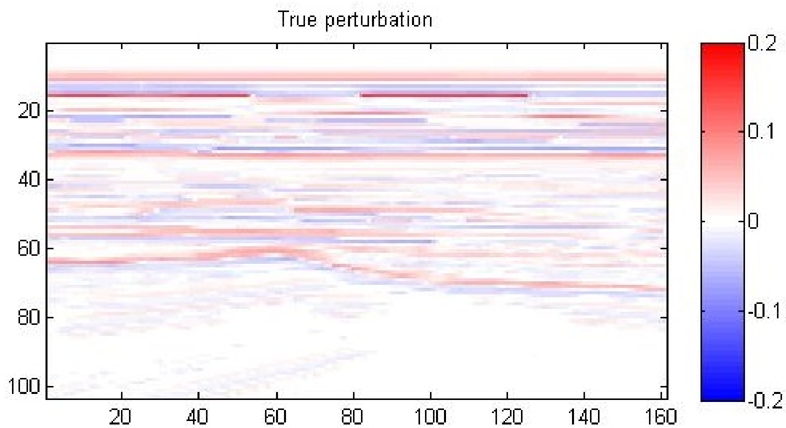
- Initialize $m^0$, $y^0$. Set $\beta \in (0, \frac{2}{L})$, $\gamma \in (0, 1 + \min\{\frac{1}{2}, \frac{1}{\beta L} - \frac{1}{2}\})$.
- For $t = 0, 1, 2, \ldots$

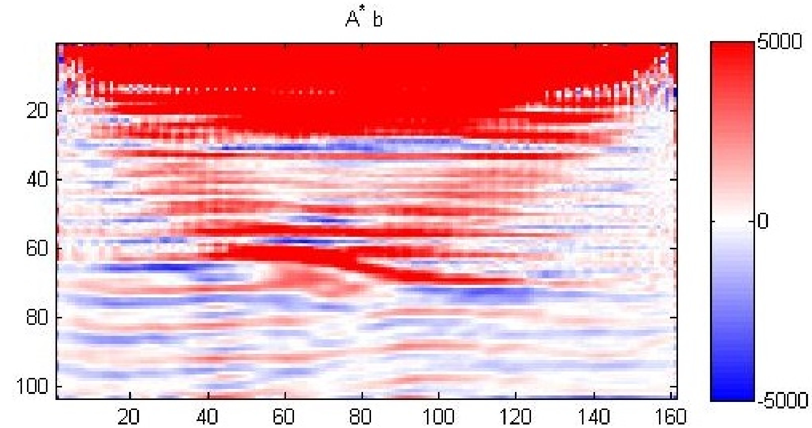
$$\begin{cases} w^t & = y^t + \beta^{-1}\mathcal{C}\left[m^t - \beta\mathcal{A}^*(\mathcal{A}m^t - b) - \beta\mathcal{C}^* y^t\right], \\ y^{t+1} & = w^t - \beta^{-1}\mathrm{Proj}_{\|\cdot\|_1 \le \tau}\left(\beta w^t\right), \\ m^{t+1} & = m^t - \gamma\beta\mathcal{A}^*(\mathcal{A}m^t - b) - \gamma\beta\mathcal{C}^* y^{t+1}. \end{cases}$$

Remark:

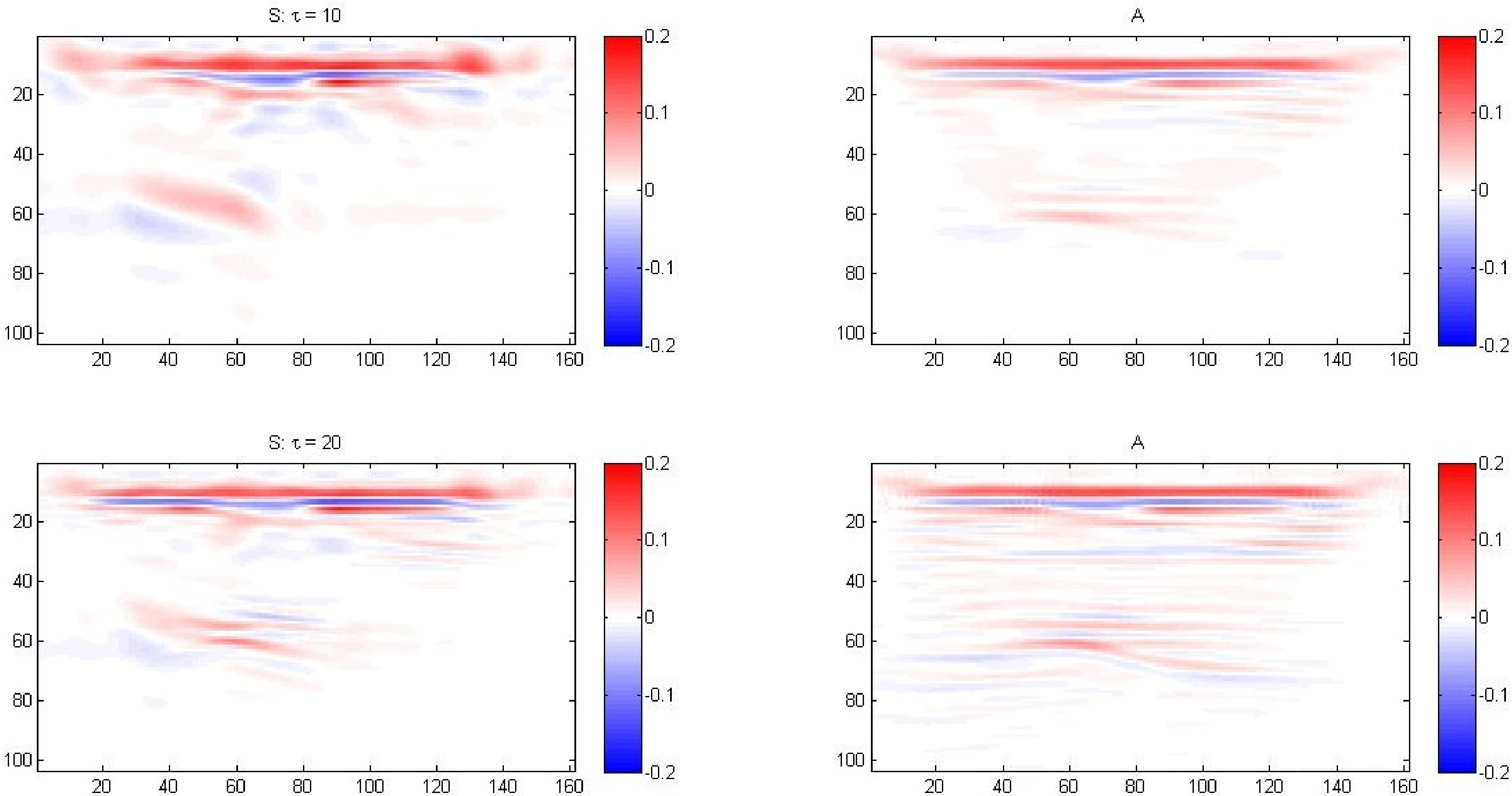
- This is a very inexact proximal gradient algorithm: Take away the inner loop!
- $\{m^t\}$ converges to an optimal solution and $\{y^t, \mathcal{A}^*(\mathcal{A}m^t - b)\}$ converges to a dual optimal solution. (P '13)

# RTM Imaging

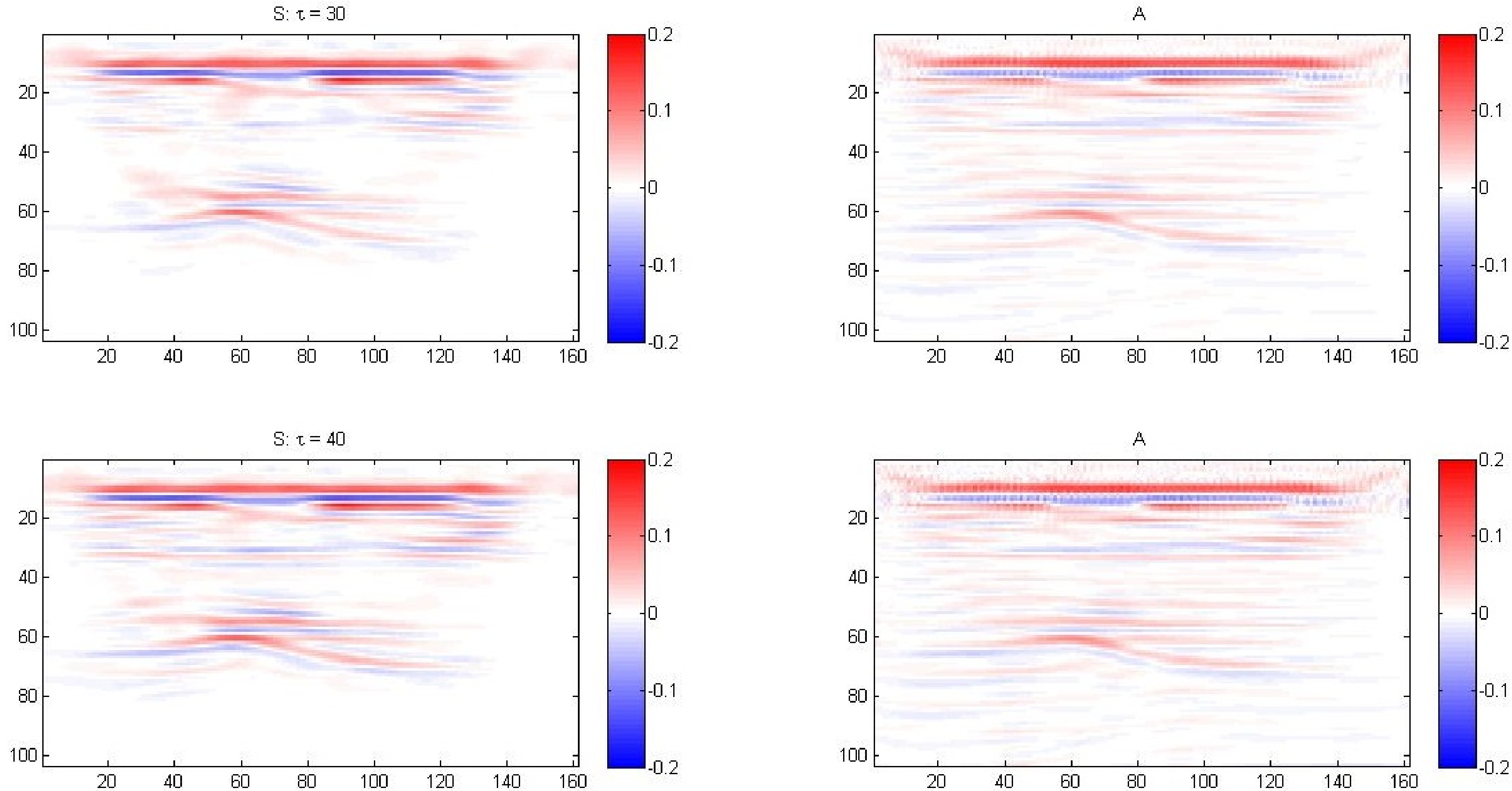

True perturbation
20
40
60
80
100
20 40 60 80 100 120 140 160
0.2
0.1
0
-0.1
-0.2


A* b
20
40
60
80
100
20 40 60 80 100 120 140 160
5000
0
-5000

# After 100 Iterations



# After 100 Iterations



# Conclusion and Extension

- The proximal-proximal gradient algorithm, which is a variant of a primitive version of the SPGL1, is a promising approach for solving the analysis model.
- Extension to allow adaptive stepsize?
- Extension to allow more general objective?

Thanks for coming! ⌣̈