Sparsity-promoting least-square migration with linearized Bregman and compressive Sensing

Ning Tu, Ernie Esser, Philipp Witte*, Mengmeng Yang, Mathias Louboutin and Felix J. Herrmann

FWI workshop Natal, September 2015

SLIM
University of British Columbia

Motivation: From processing to inversion

Shift from seismic processing to inversion exposes challenges:

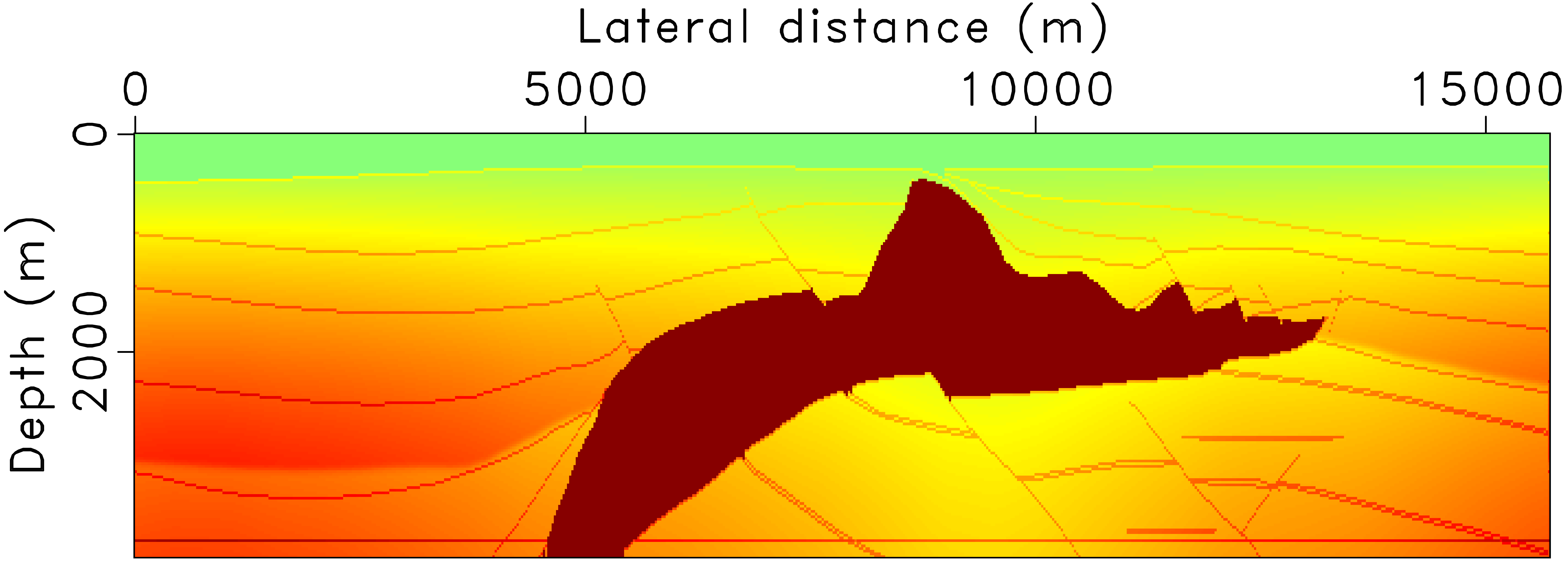
- handling I/O of large data volumes
- high computational demand of wave-equation based inversions

Sparsity promoting inversions:

- produce high-resolution results
- are computationally expensive & require many passes through data
- are algorithmically complex

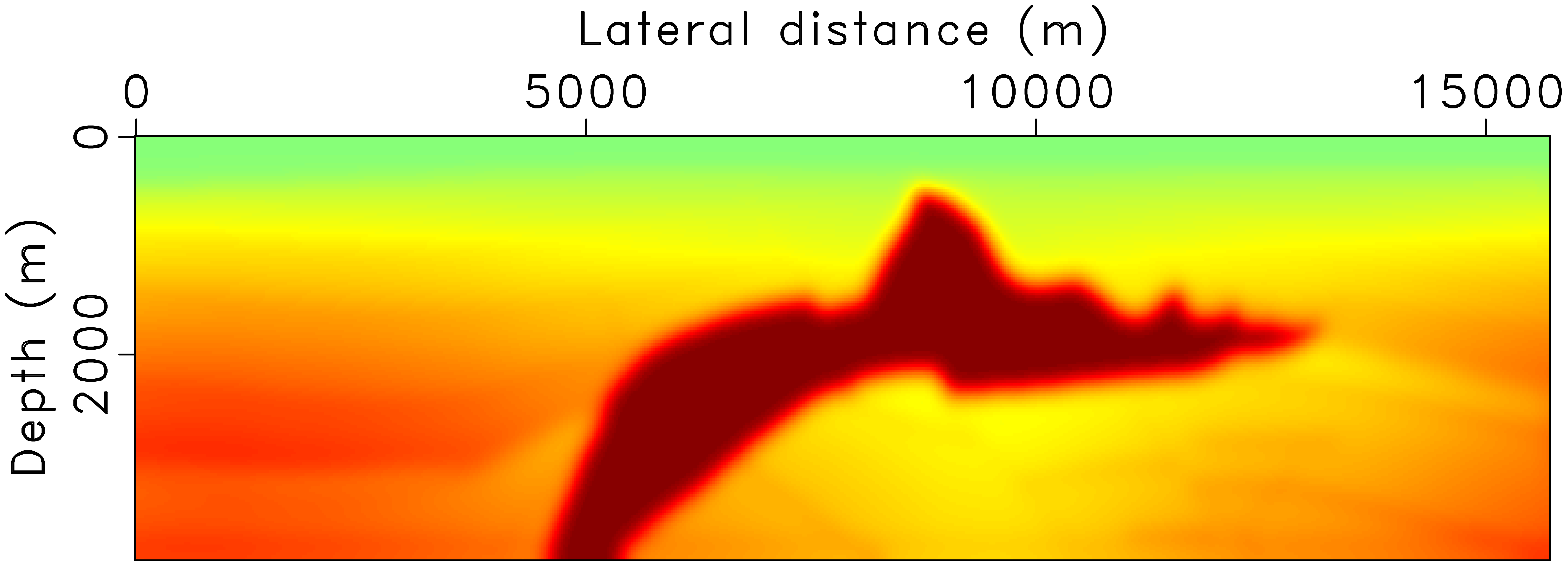
Problematic for adaption by industry

From processing to inversion

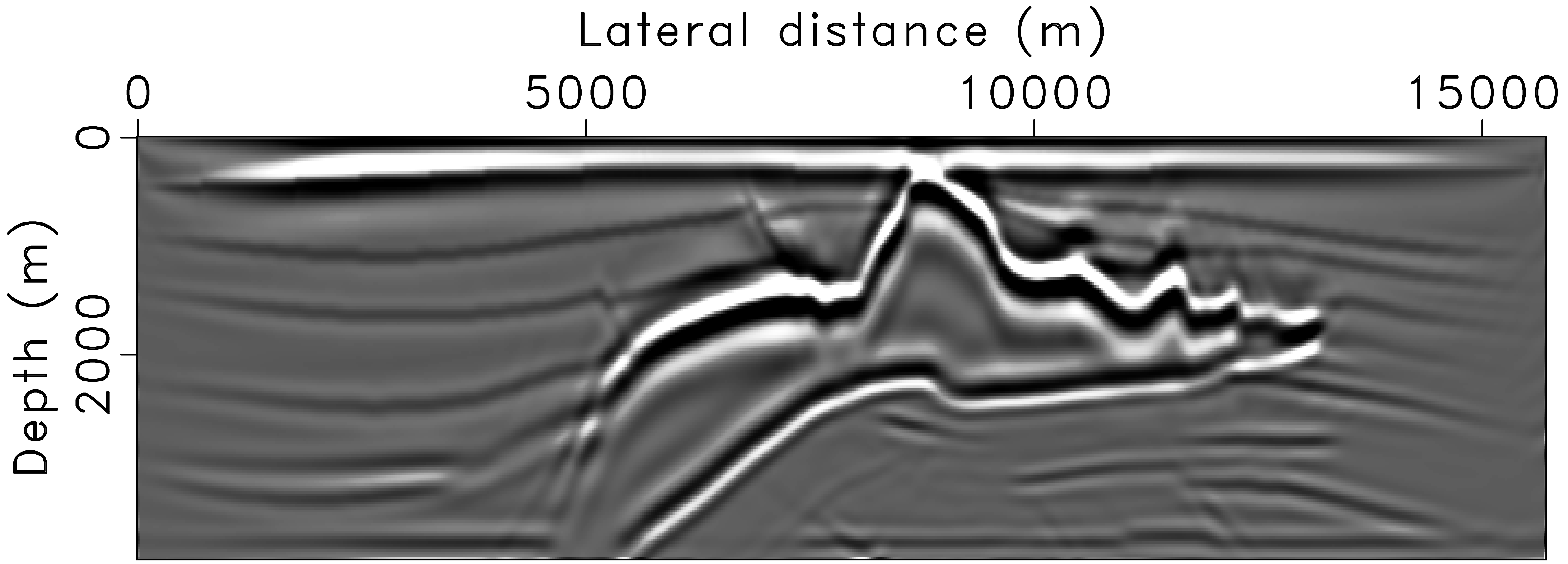


SLIM

From processing to inversion

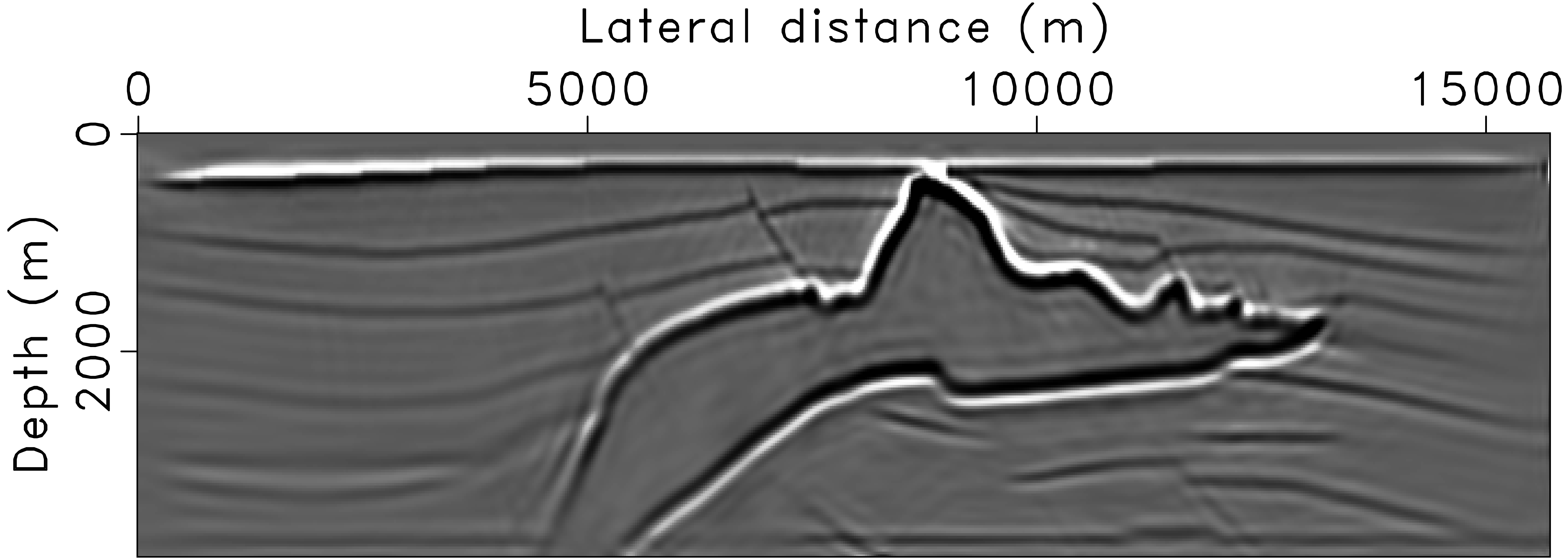


From processing to inversion



RTM imaging via **adjoint**, high-pass filtered to remove low-wavenumber RTM artifacts

From processing to inversion

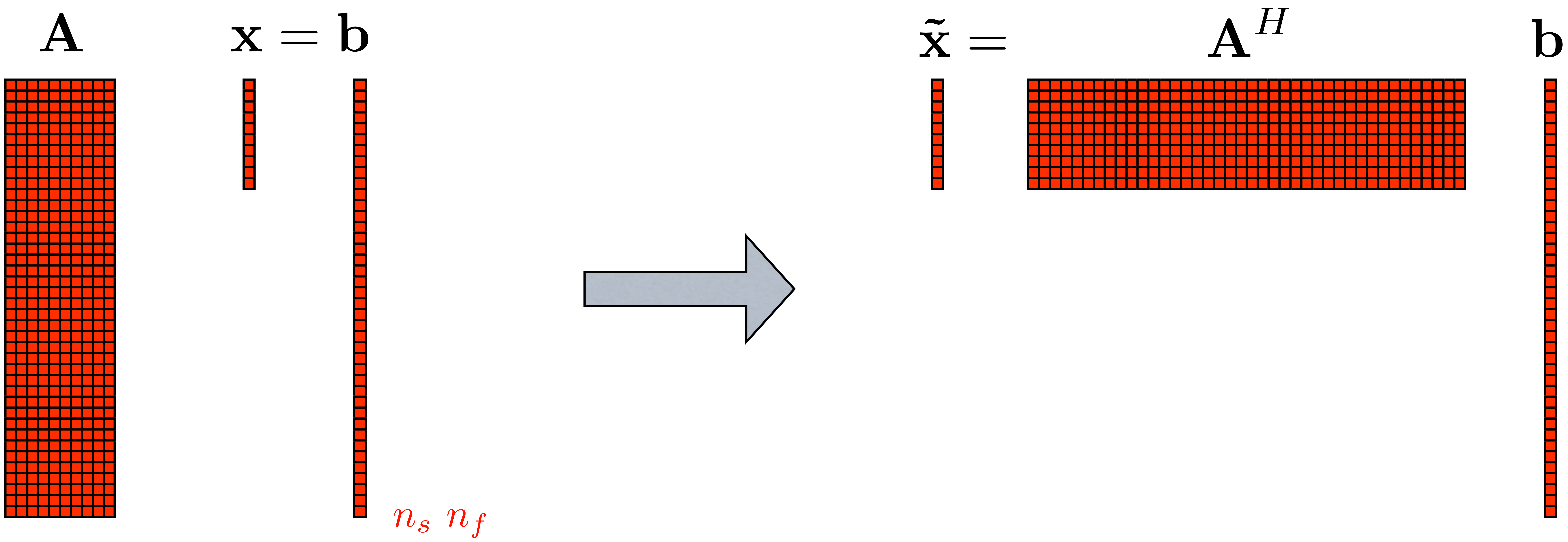


SPLSM image via **inversion**, # of wave-equation solves roughly equals 1 RTM w/ all data

Migration

Seismic problems are

- often overdetermined
- so far: often solved by applying the (scaled) adjoint



Migration

Solving $\mathbf{A}\mathbf{x} = \mathbf{b}$ with $\tilde{\mathbf{x}} = \mathbf{A}^H\mathbf{b}$ ➡ RTM image

Alternatively: least squares solution by solving

$$\underset{\mathbf{x}}{\text{minimize}} \;\; \frac{1}{2}\|\mathbf{A}\mathbf{x} - \mathbf{b}\|_2^2$$

with solution $\mathbf{x} = (\underbrace{\mathbf{A}^H\mathbf{A}}_{\text{GN Hessian}})^{-1}\mathbf{A}^H\mathbf{b}$

Inverting GN Hessian expensive ➡ solve with linear optimization

Migration with sparsity promotion

Normal least squares solution:

- does not exploit structure in $\mathbf{x}$
- requires many iterations (data passes)

Sparsity promoting inversion

- "classic" CS problem:

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{x}\|_1 \quad \text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}$$

- Designed for underdetermined systems
(but we will later see it works for overdetermined systems too!)

Sparsity promoting inversion

Various algorithms/solvers to obtain solution of basis pursuit (BP) problem

$$\underset{\mathbf{x}}{\text{minimize}} \quad \|\mathbf{x}\|_1$$

$$\text{subject to } \mathbf{A}\mathbf{x} = \mathbf{b}$$

Iterative shrinkage thresholding algorithm (ISTA)

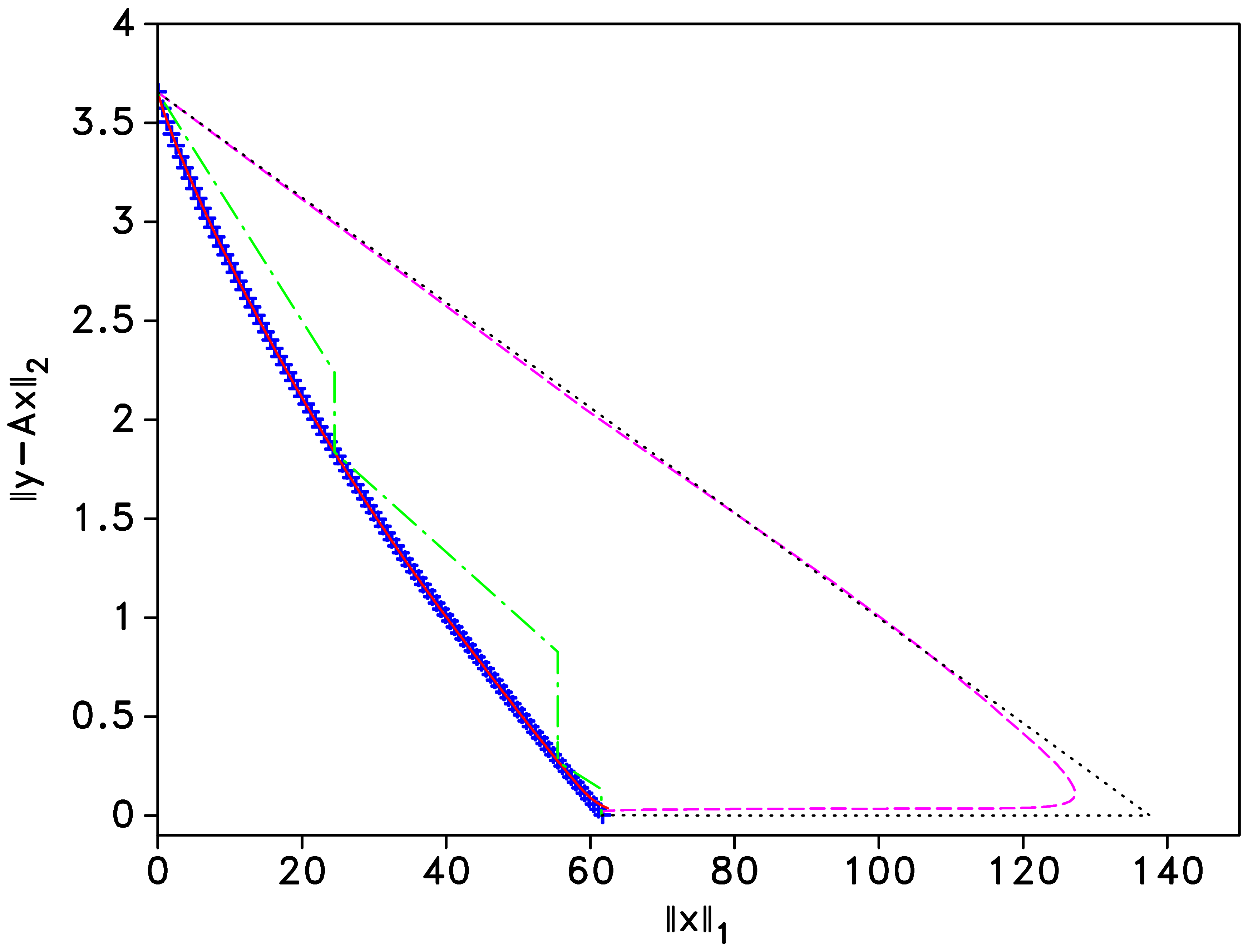
1. **for** $k = 0, 1, \cdots$
2. $\quad \mathbf{z}_{k+1} = \mathbf{x}_k - t_k \mathbf{A}^*(\mathbf{A}\mathbf{x}_k - \mathbf{b}_k)$
3. $\quad \mathbf{x}_{k+1} = S_\lambda(\mathbf{z}_{k+1})$
4. **end for**

*where $S_\lambda(x) = \text{sign}(x) \cdot \max(|x| - \lambda, 0)$ is soft thresholding and t_k are step lengths

Iterative soft shrinkage algorithm

Simple algorithm, but

- slow convergence, especially for λ small
- to solve BP problem, non-trivial limit for $\lambda \rightarrow 0^+$ (requires complicated continuation strategies for λ)



Other L1 solvers

Alternative algorithm: SPGL1

- solves BP problem as a series of Lasso problems

$$\begin{aligned}\text{minimize} \quad & \|\mathbf{Ax} - \mathbf{b}\|_2 \\ \text{subject to} \quad & \|\mathbf{x}\|_1 \leq \tau\end{aligned}$$

- very fast due to continuation methods that relax constraint
- black box solver with state-of-the-art “tricks”

However:

- slow convergence for realistic seismic problems with expensive I/O
- complicated implementation
- not designed for overdetermined problems (like LSRTM)

SLIM

Other L1 solvers

Linearized Bregman method (derived from “Bregman Iterative Regularization”)

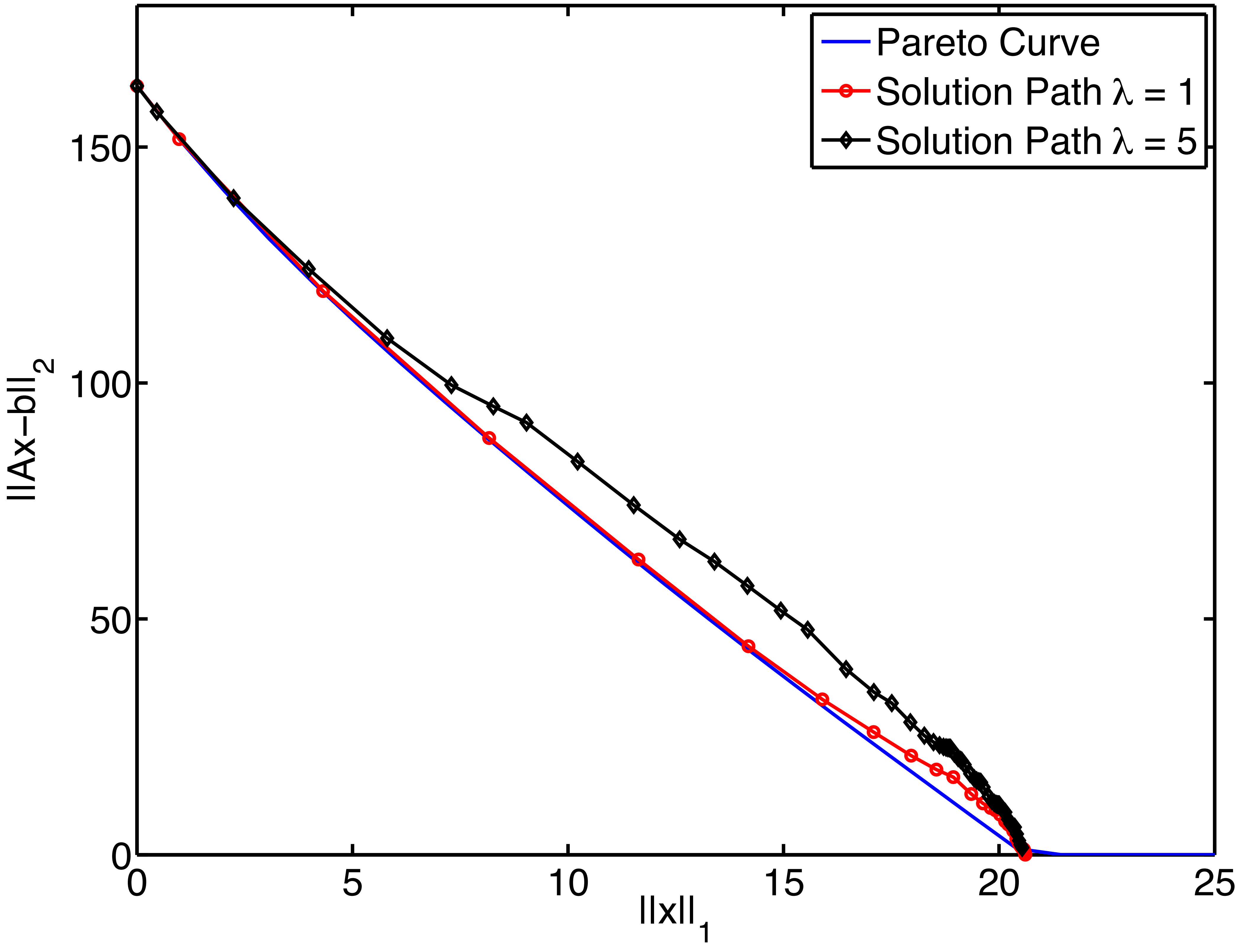
$$\begin{aligned} \underset{\mathbf{x}}{\text{minimize}} \quad & \lambda \|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|^2 \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \end{aligned}$$

- strongly convex objective function known as “elastic net” in ML
- for $\lambda \to \infty$ solves BP problem
- simple algorithm with iterations

$$\begin{aligned} \mathbf{z}_{k+1} &= \mathbf{z}_k - t_k \mathbf{A}^* (\mathbf{A}\mathbf{x}_k - \mathbf{b}) \\ \mathbf{x}_{k+1} &= S_\lambda(\mathbf{z}_{k+1}) \end{aligned}$$

Linearized Bregman solution paths

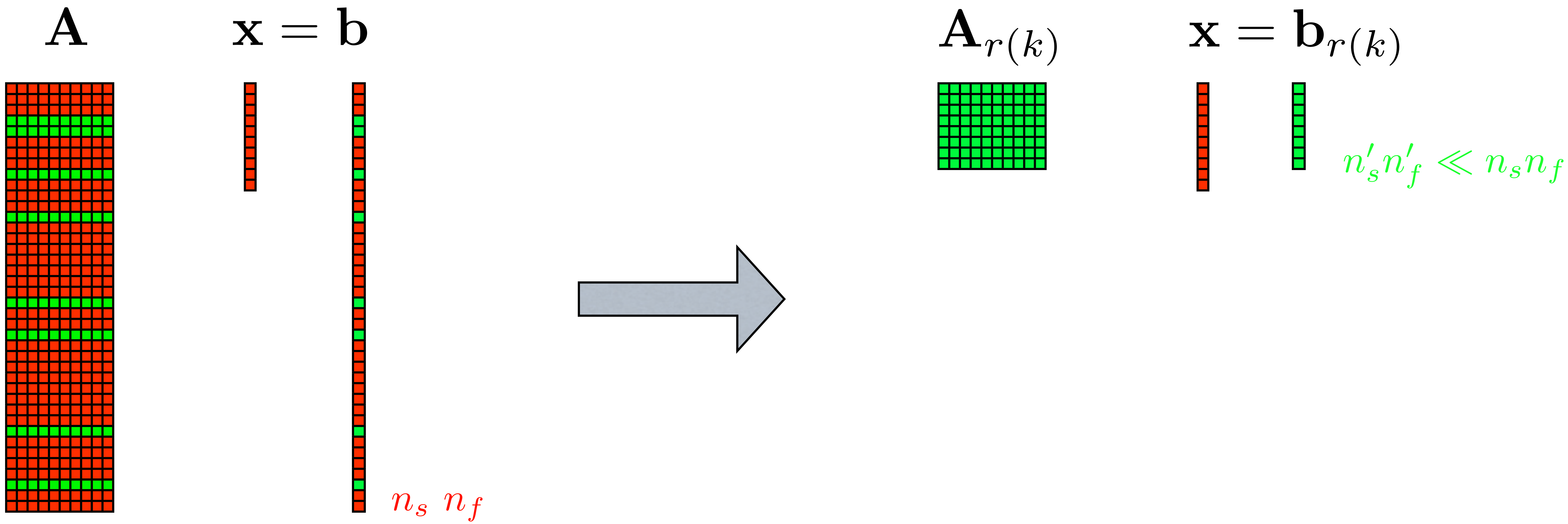
Follows pareto curve closer than ISTA + less sensitive to λ



Randomized L1 solvers

For large scale seismic problems we are interested in:

- reducing time consuming I/O
- working on (random) subsets of data and rows of $\mathbf{A}$



Felix J. Herrmann and Xiang Li, "Efficient least-squares imaging with sparsity promotion and compressive sensing", *Geophysical Prospecting*, vol. 60, p. 696-712, 2012
Felix J. Herrmann, "Accelerated large-scale inversion with message passing", in *SEG Technical Program Expanded Abstracts*, 2012, vol. 31, p. 1-6.

SLIM

Randomized L1 solvers

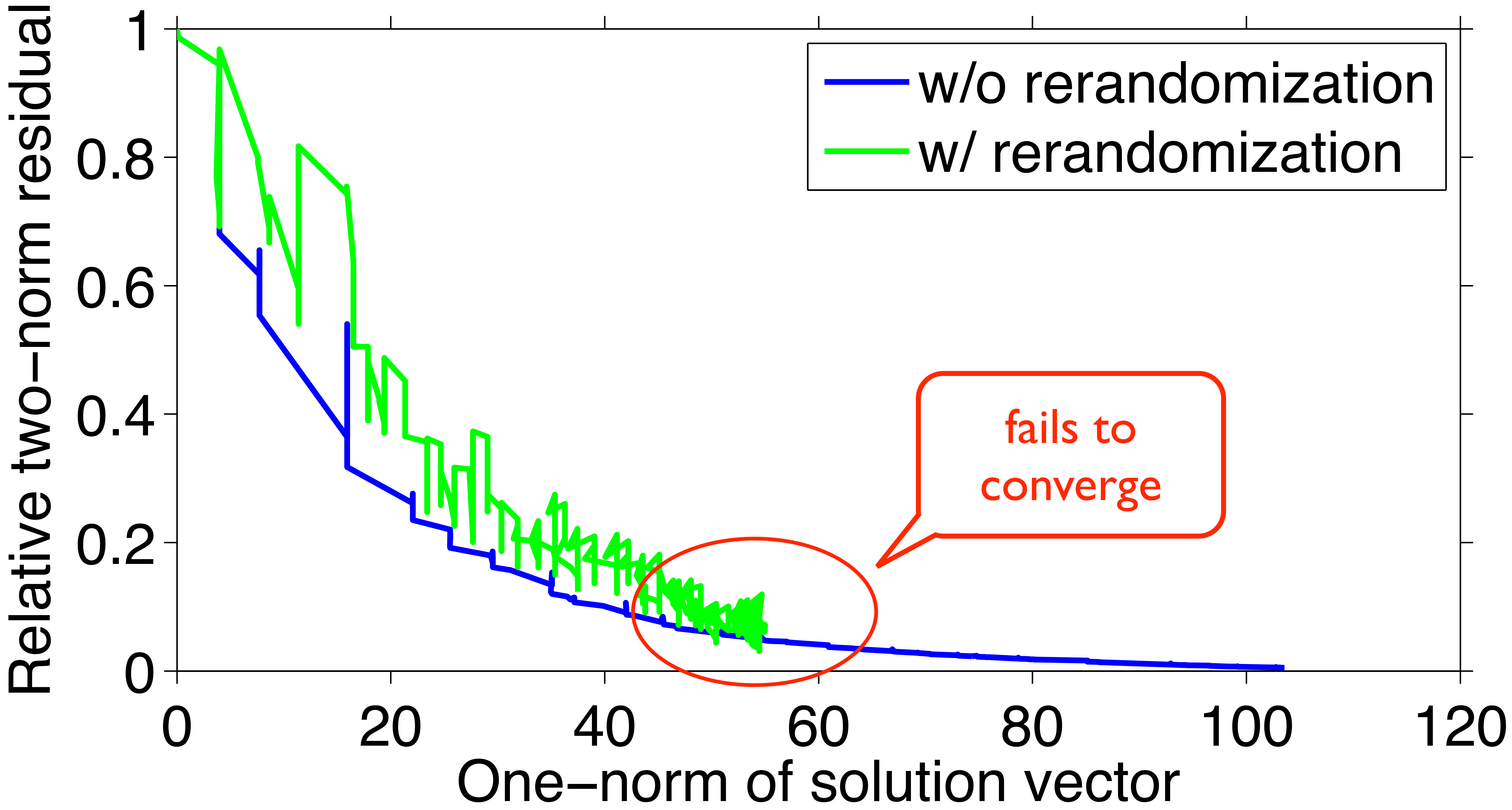
Randomized iterative soft thresholding algorithm (RISTA):

1. **for** $k = 0, 1, \cdots$
2. $\mathbf{z}_{k+1} = \mathbf{x}_k - t_k \mathbf{A}^*_{r(k)}(\mathbf{A}_{r(k)}\mathbf{x}_k - \mathbf{b}_k)$
3. $\mathbf{x}_{k+1} = S_{\lambda_k}(\mathbf{z}_{k+1})$
4. **end for**

*where $S_\lambda(x) = \text{sign}(x) \cdot \max(|x| - \lambda, 0)$ is soft thresholding and t_k are step lengths

- relates to "approximate" message passing theory (Montanari, '09)
- reduces I/O, works on small subset of data
- only converges for special $\mathbf{A}^*, \mathbf{A}$ and tuned λ s

RISTA solution path



SLIM

Randomized L1 solvers

- Randomized version that works on L subsets of $\mathbf{A}, \mathbf{b}$ also exists for the linearized Bregman method:

$$
\begin{array}{ll}
1. & \textbf{for } k = 0, 1, \cdots \\
2. & \quad \mathbf{z}_{k+1} = \mathbf{z}_k - t_k \mathbf{A}^*_{r(k)} (\mathbf{A}_{r(k)} \mathbf{x}_k - \mathbf{b}_{r(k)}) \\
3. & \quad \mathbf{x}_{k+1} = S_\lambda(\mathbf{z}_{k+1}) \\
4. & \textbf{end for}
\end{array}
$$

- For L=1 $\rightarrow$ Linearized Bregman
- For L=m $\rightarrow$ Sparse Kaczmarz

SLIM

Linearized Bregman

Extension to handle noisy data

$$\underset{\mathbf{x}}{\text{minimize}} \quad \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|^2$$
$$\text{subject to} \quad \|\mathbf{A}\mathbf{x} - \mathbf{b}\| \leq \sigma$$

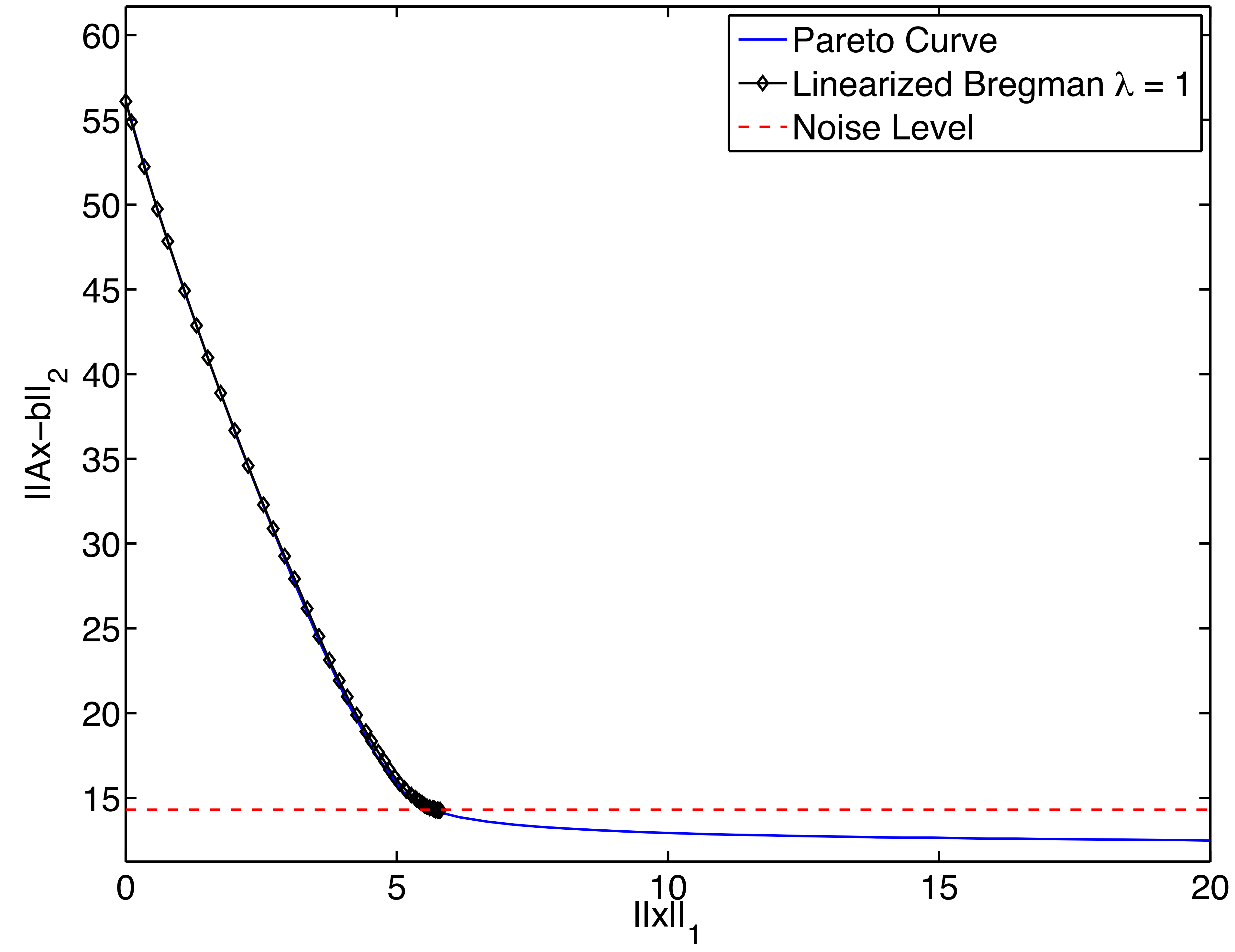
via projections onto norm balls

1. **for** $k = 0, 1, \cdots$
2. $\mathbf{z}_{k+1} = \mathbf{z}_k - t_k \mathbf{A}^*_{r(k)} \mathcal{P}_\sigma(\mathbf{A}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)})$
3. $\mathbf{x}_{k+1} = S_\lambda(\mathbf{z}_{k+1})$
4. **end for**

*where $\mathcal{P}_\sigma(\mathbf{A}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)}) = \max\{0, 1 - \frac{\sigma}{\|\mathbf{A}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)}\|}\} \cdot (\mathbf{A}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)})$

Linearized Bregman solution path

Converges (up to noise level), even when working on randomized subsets of data + without difficult strategies for λ



Preliminary conclusions

Various solvers to iteratively solve sparsity promoting least squares problems:

- Iterative soft thresholding
- SPGL1
- Linearized Bregman

Among these, the linearized Bregman methods stands out:

- easy to implement (unlike SPGL1)
- can work on subsets of rows and data (unlike SPGL1) and still converge (unlike RISTA)!
- no cooling of λ, just needs to be bigger than some threshold

→ Great candidate to use for least squares migration

Least squares migration

Switch to seismic notation: $\delta \mathbf{d}_{ij} = \nabla \mathbf{F}_{ij}(\mathbf{m}_0, \mathbf{q}_{ij})\delta \mathbf{m}$

$\delta \mathbf{m}$: Model perturbation
$\delta \mathbf{d}$: data residual
$\nabla \mathbf{F}$: Born modeling operator
i, j : frequency, source index

Instead of applying $\nabla \mathbf{F}^H$ (RTM)

$$\delta \mathbf{m} = \sum_{ij} \nabla \mathbf{F}_{ij}^H(\mathbf{m}_0, \mathbf{q}_{ij})\delta \mathbf{d}_{ij}$$

➡ Solve least squares problem with sparsity constraint (SPLSM) using the linearized Bregman method

(frequency domain)

Fast SPLSM w/ CS

– w/ randomized source subsets

$$\underset{\mathbf{x}}{\text{minimize}} \quad \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|^2$$

$$\text{subject to} \quad \sum_{ij} \|\nabla \mathbf{F}_{ij}(\mathbf{m}_0, \mathbf{q}_{ij})\mathbf{C}^*\mathbf{x} - \delta\mathbf{d}_{ij}\| \leq \sigma$$

By iterating

$\mathbf{C}$: Curvelet transform

1. **for** $k = 0, 1, \cdots$
2. $\Omega \in [1 \cdots n_f]$, $\Sigma \in [1 \cdots n_s]$ for $\#\Omega \ll n_f$, $\#\Sigma \ll n_s$
3. $\mathbf{A}_k = \{\nabla \mathbf{F}_{ij}(\mathbf{m}_0, \bar{\mathbf{q}}_{ij})\mathbf{C}^*\}_{i\in\Omega, j\in\Sigma}$ with $\bar{\mathbf{q}}_{ij} = \sum_{l=1}^{n_s} w_l \mathbf{q}_{i,l}$
4. $\mathbf{b}_k = \{\delta\bar{\mathbf{d}}_{ij}\}_{i\in\Omega, j\in\Sigma}$ with $\delta\bar{\mathbf{d}}_{ij} = \sum_{l=1}^{n_s} w_l \delta\mathbf{d}_{i,l}$
5. $\mathbf{z}_{k+1} = \mathbf{z}_k - t_k \mathbf{A}_k^* \mathcal{P}_\sigma(\mathbf{A}_k\mathbf{x}_k - \mathbf{b}_k)$
5. $\mathbf{x}_{k+1} = S_\lambda(\mathbf{z}_{k+1})$
6. **end for**

Fast SPLSM w/ CS
– experimental setup

Data:

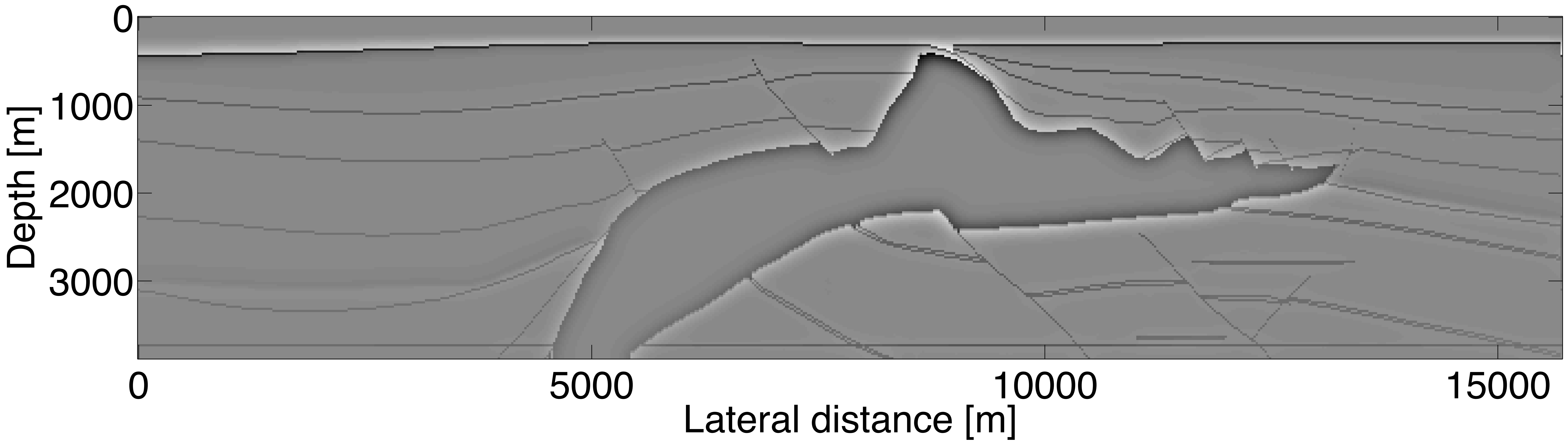
- 320 sources and receivers
- 72 frequency slices ranging from 3 – 12 Hz
- $\delta\mathbf{d} = \mathbf{F}(\mathbf{m}) - \mathbf{F}(\mathbf{m}_0)$, generated with separate modeling engine

Experiments:

- one pass through the data with different batch/block sizes
- simultaneous vs sequential shots
- choose λ according to $\max\left(t_1 \cdot \mathbf{A}_1^{*}\mathbf{b}_1\right)$ and number of iterations
- no source estimation – use correct source for linearized inversions

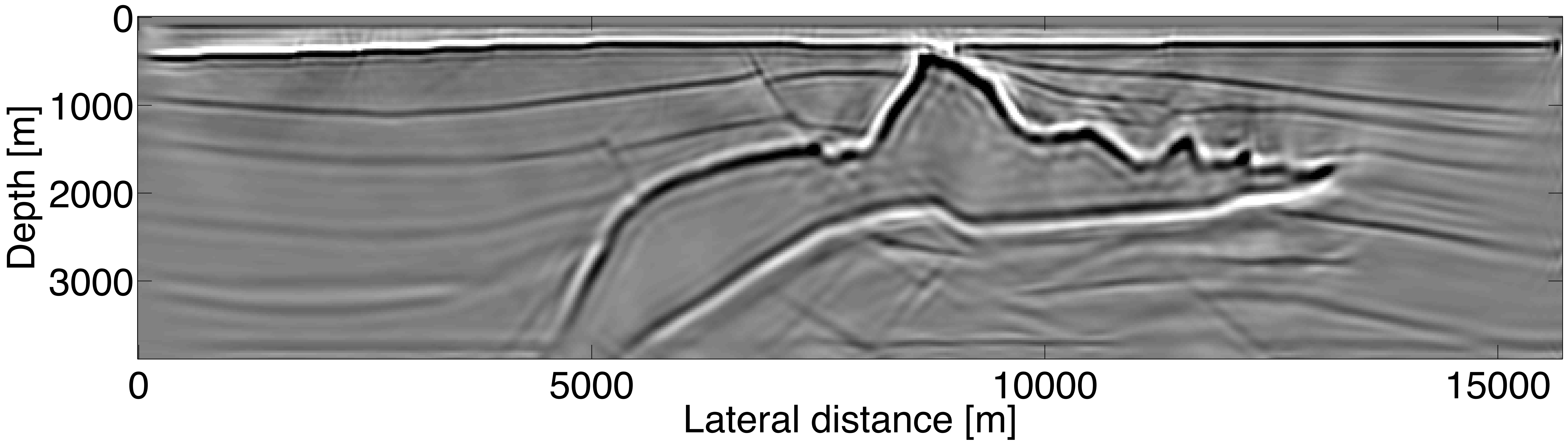
Fast SPLSM w/ CS

– True model perturbation

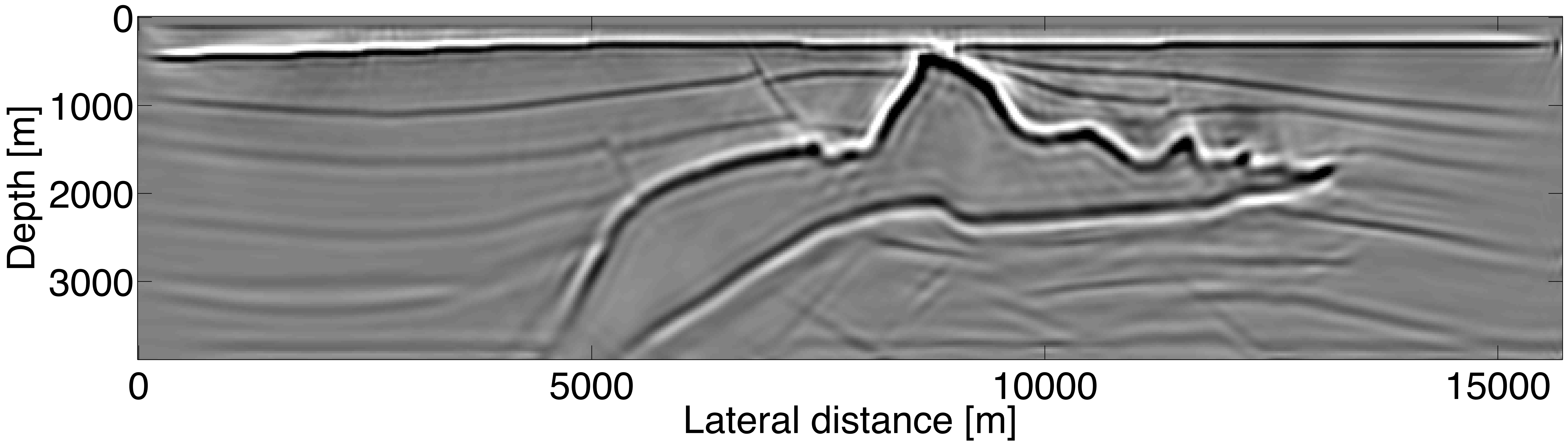


Fast SPLSM w/ CS

– 360 iterations, each w/ 8 frequencies/sim. shots

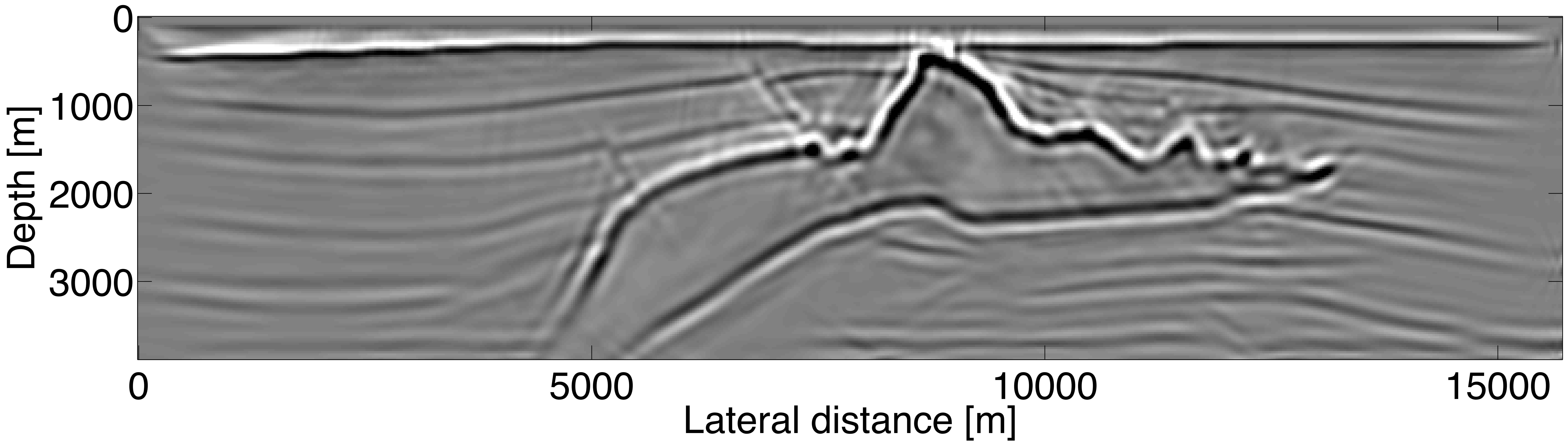


Fast SPLSM w/ CS
– 90 iterations, each w/ 16 frequencies/sim. shots



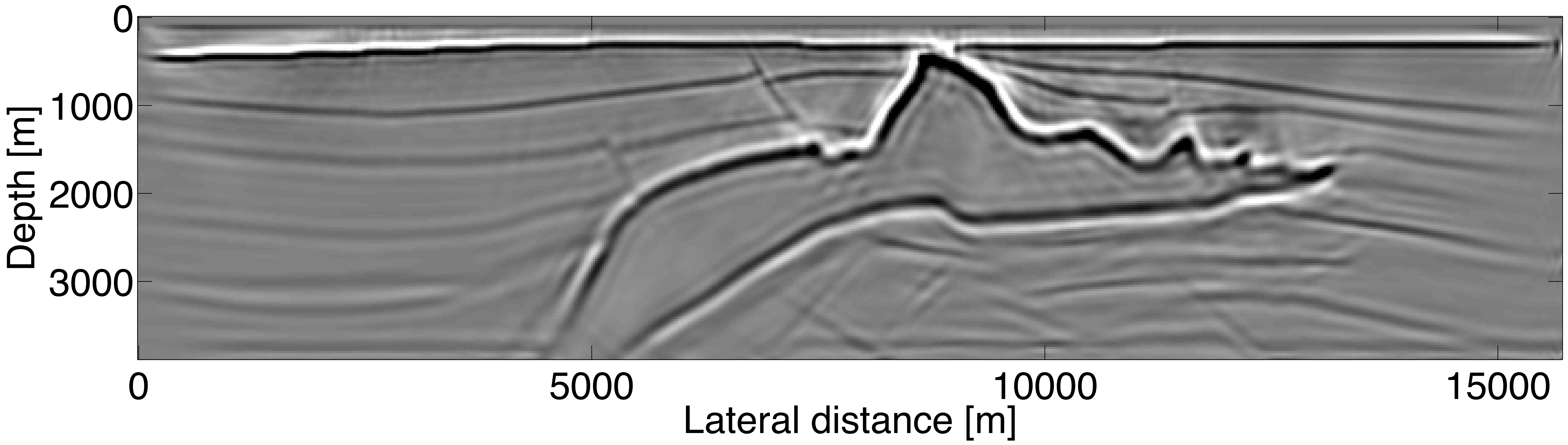
Fast SPLSM w/ CS

– 23 iterations, each w/ 32 frequencies/sim. shots



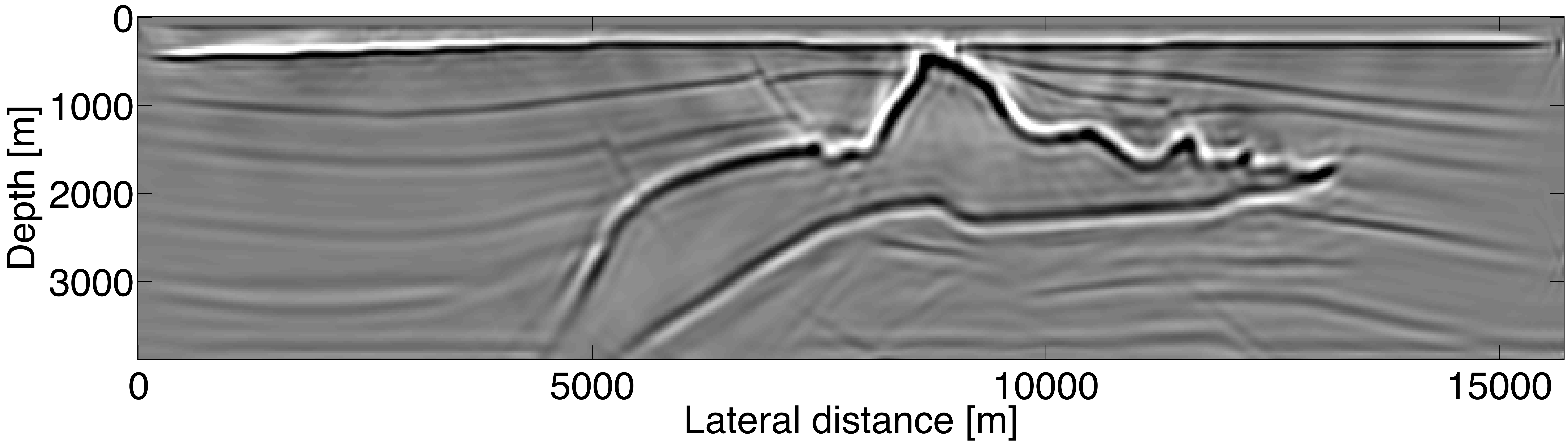
Fast SPLSM w/ CS

– 90 iterations, each w/ 16 frequencies/sim. shots



Fast SPLSM w/ CS

– 90 iterations, each w/ 16 frequencies/sequential shots



Fast SPLSM w/ CS

– on-the-fly source estimation

$$\underset{\mathbf{x}}{\text{minimize}} \quad \lambda\|\mathbf{x}\|_1 + \frac{1}{2}\|\mathbf{x}\|^2$$
$$\text{subject to} \quad \sum_{ij} \|\nabla\mathbf{F}_{ij}(\mathbf{m}_0, \mathbf{q}_{ij})\mathbf{C}^*\mathbf{x} - \delta\mathbf{d}_{ij}\| \leq \sigma$$

By iterating

1. **for** $k = 0, 1, \cdots$
2. $\Omega \in [1 \cdots n_f]$, $\Sigma \in [1 \cdots n_s]$ for $\#\Omega \ll n_f$, $\#\Sigma \ll n_s$
3. $\mathbf{A}_k = \{\nabla\mathbf{F}_{ij}(\mathbf{m}_0, \bar{\mathbf{q}}_{ij})\mathbf{C}^*\}_{i\in\Omega, j\in\Sigma}$ with $\bar{\mathbf{q}}_{ij} = \sum_{l=1}^{n_s} w_l \mathbf{q}_{i,l}$
4. $\mathbf{b}_k = \{\delta\bar{\mathbf{d}}_{ij}\}_{i\in\Omega, j\in\Sigma}$ with $\delta\bar{\mathbf{d}}_{ij} = \sum_{l=1}^{n_s} w_l \delta\mathbf{d}_{i,l}$
5. $\bar{\mathbf{q}}_{ij} = \frac{<A_k x_k, b_k>}{<A_k x_k, A_k x_k>}$, $\mathbf{A}_k = \{\nabla\mathbf{F}_{ij}(\mathbf{m}_0, \bar{\mathbf{q}}_{ij})\mathbf{C}^*\}_{i\in\Omega, j\in\Sigma}$
6. $\mathbf{z}_{k+1} = \mathbf{z}_k - t_k \mathbf{A}_k^* \mathcal{P}_\sigma(\mathbf{A}_k\mathbf{x}_k - \mathbf{b}_k)$
7. $\mathbf{x}_{k+1} = S_\lambda(\mathbf{z}_{k+1})$
8. **end for**

Fast SPLSM w/ source estimation
– experimental setup

Data:

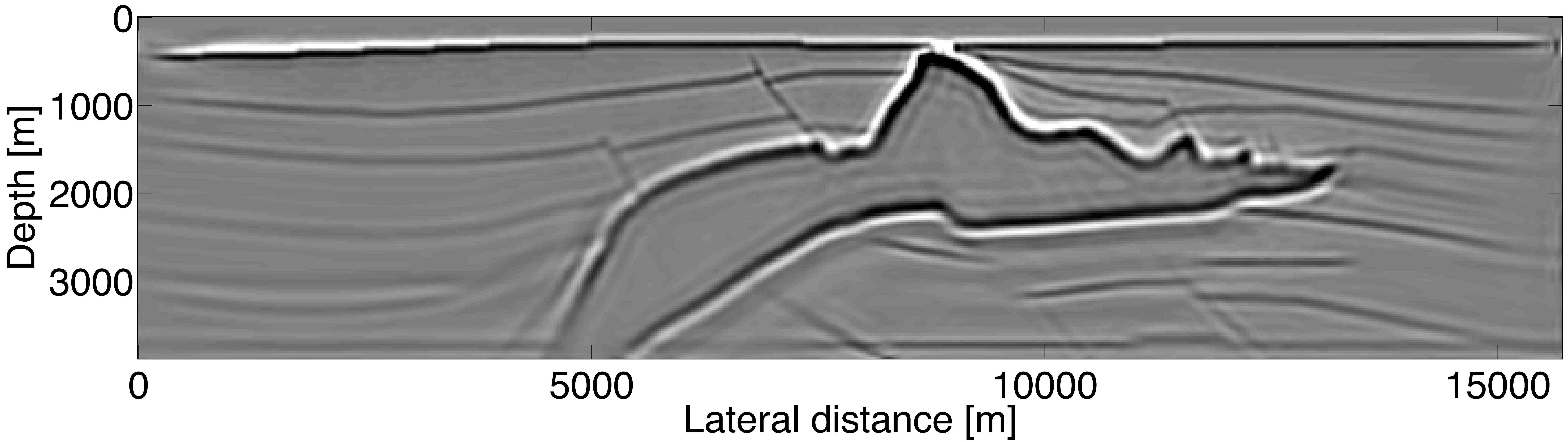
- 320 sources and receivers
- 72 frequency slices ranging from 3 - 12 Hz
- $\delta \mathbf{d} = \mathbf{F}(\mathbf{m} - \mathbf{m}_0)$ inverse crime data

Experiments:

- one pass through the data with the same block size
- simultaneous sources
- choose λ according to $\max\left(t_1 \cdot \mathbf{A}_1^* \mathbf{b}_1\right)$
- source estimation with delta Dirac as initial guess
- estimated source scaled w.r.t. true source

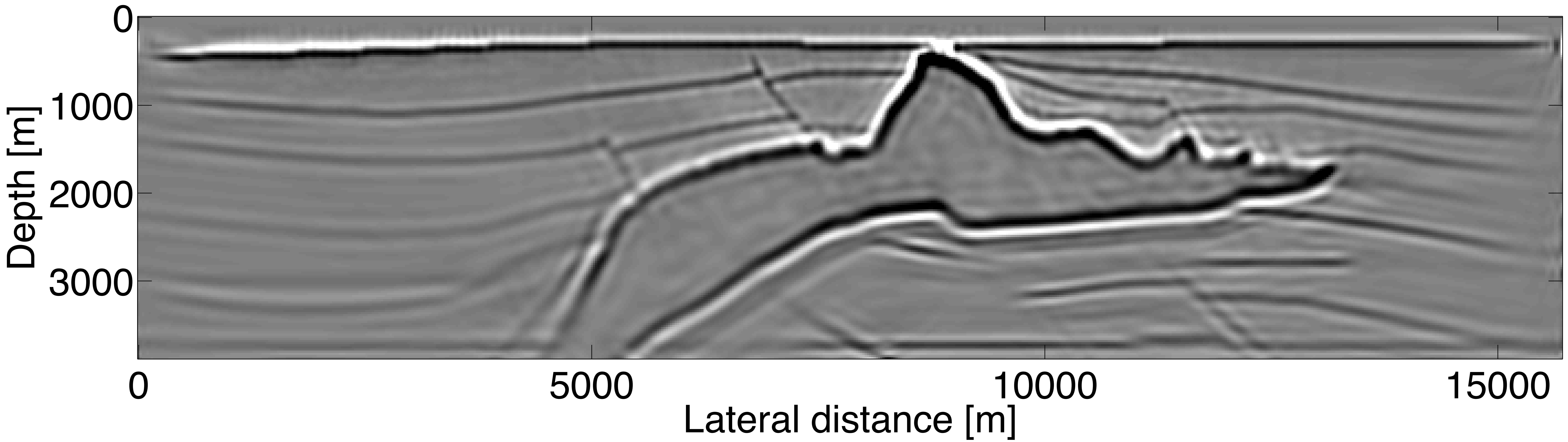
Fast SPLSM w/ source estimation

– 80 iterations, each w/ 72 frequencies/4sim. shots & true source



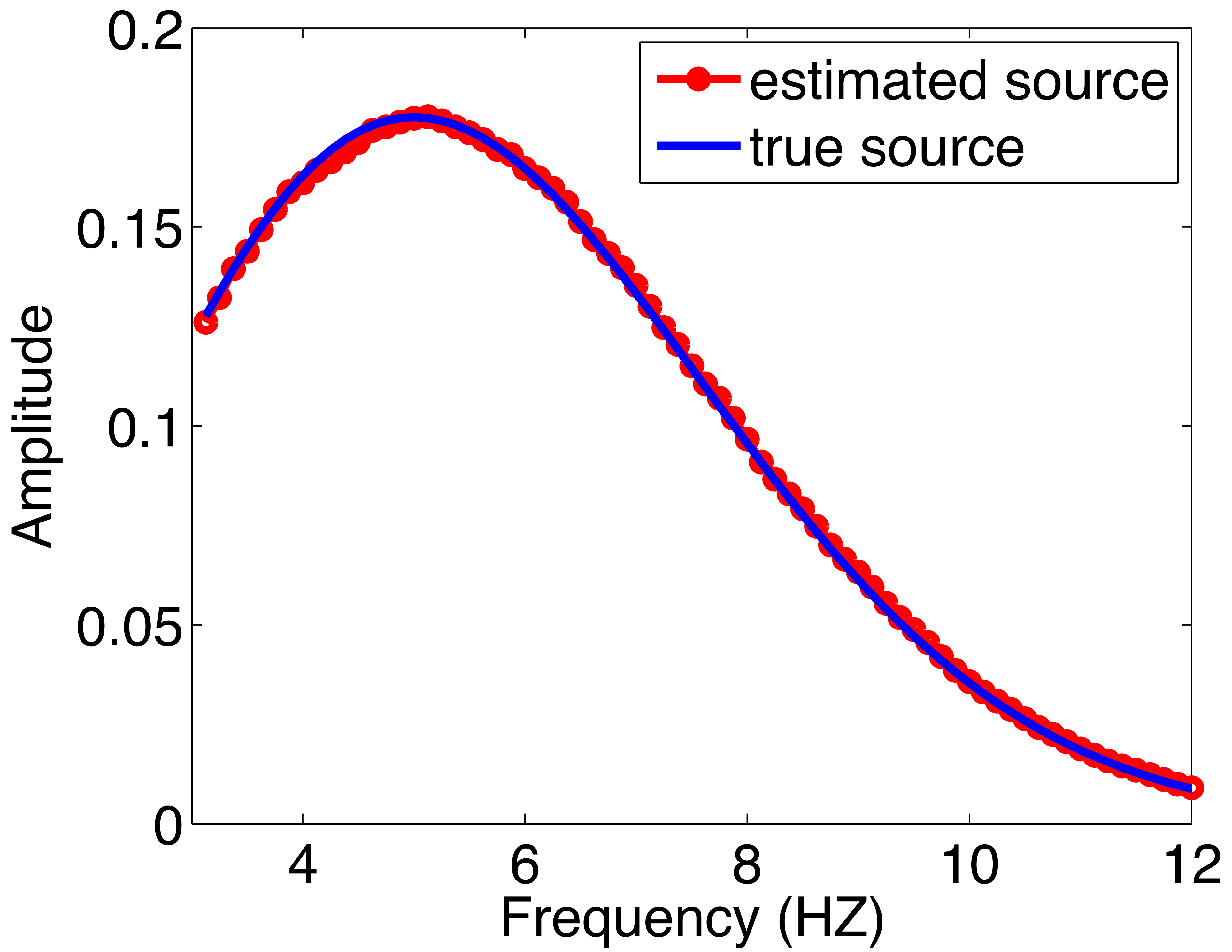
Fast SPLSM w/ source estimation

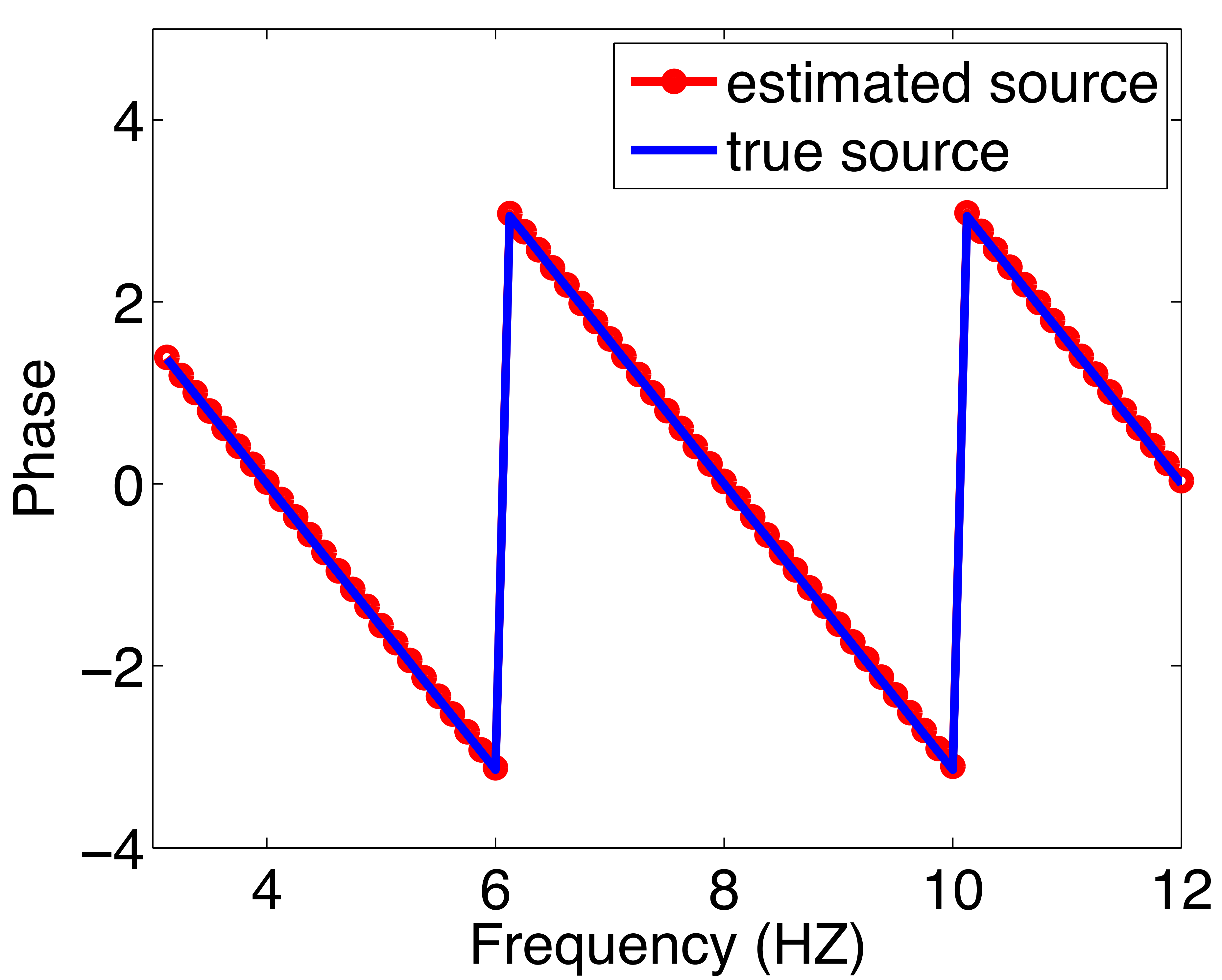
– estimated source



Fast SPLSM w/ source estimation

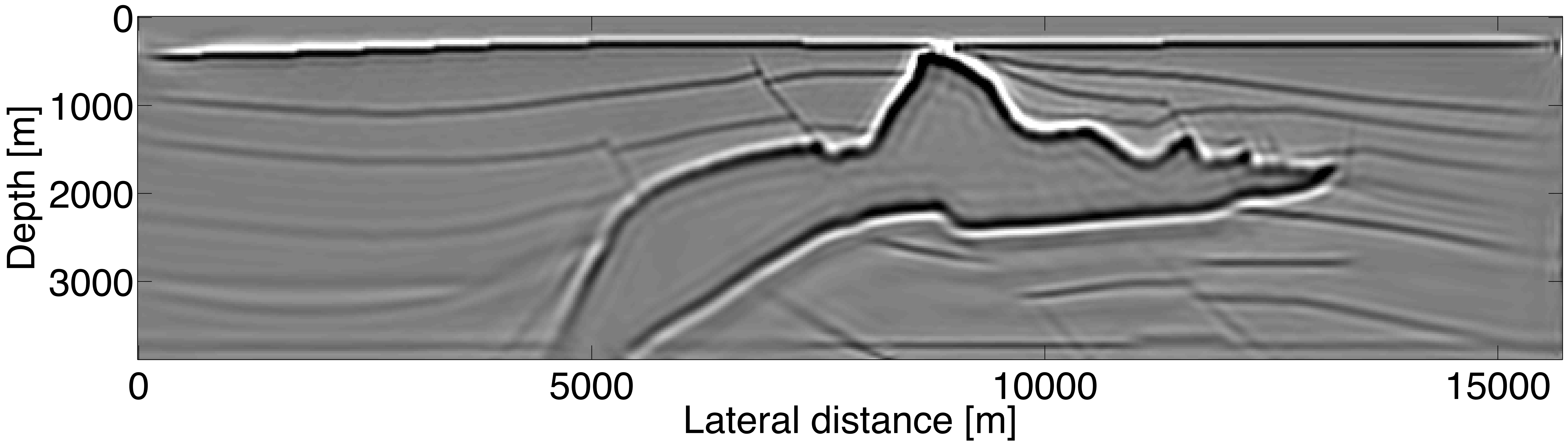
– estimated source





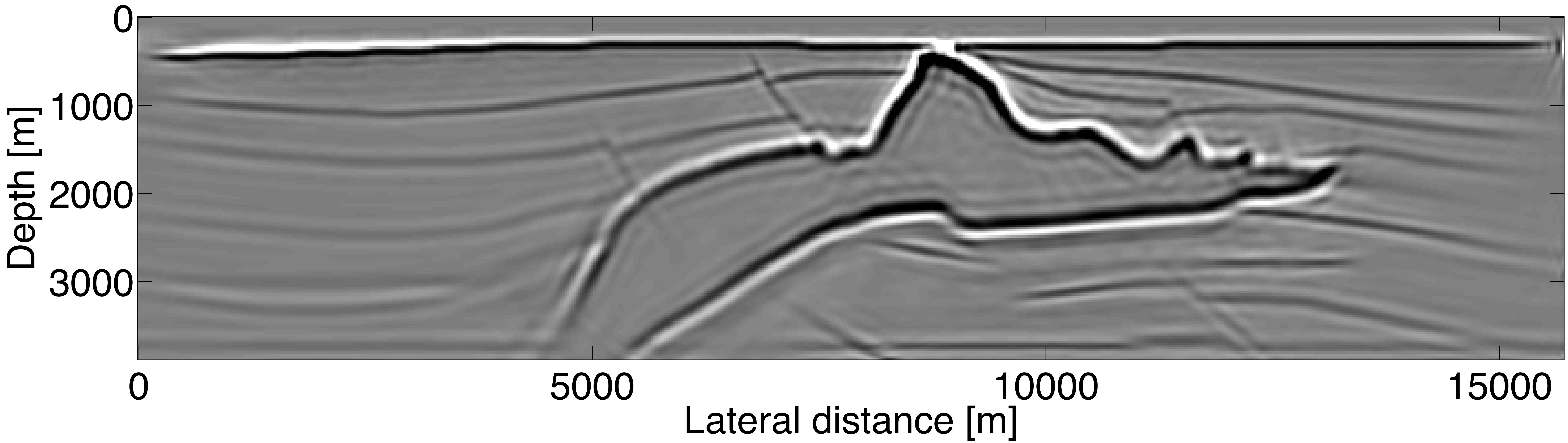
Fast SPLSM w/ source estimation

– 90 iterations, each w/ 16 frequencies/16sim. shots w/ true source



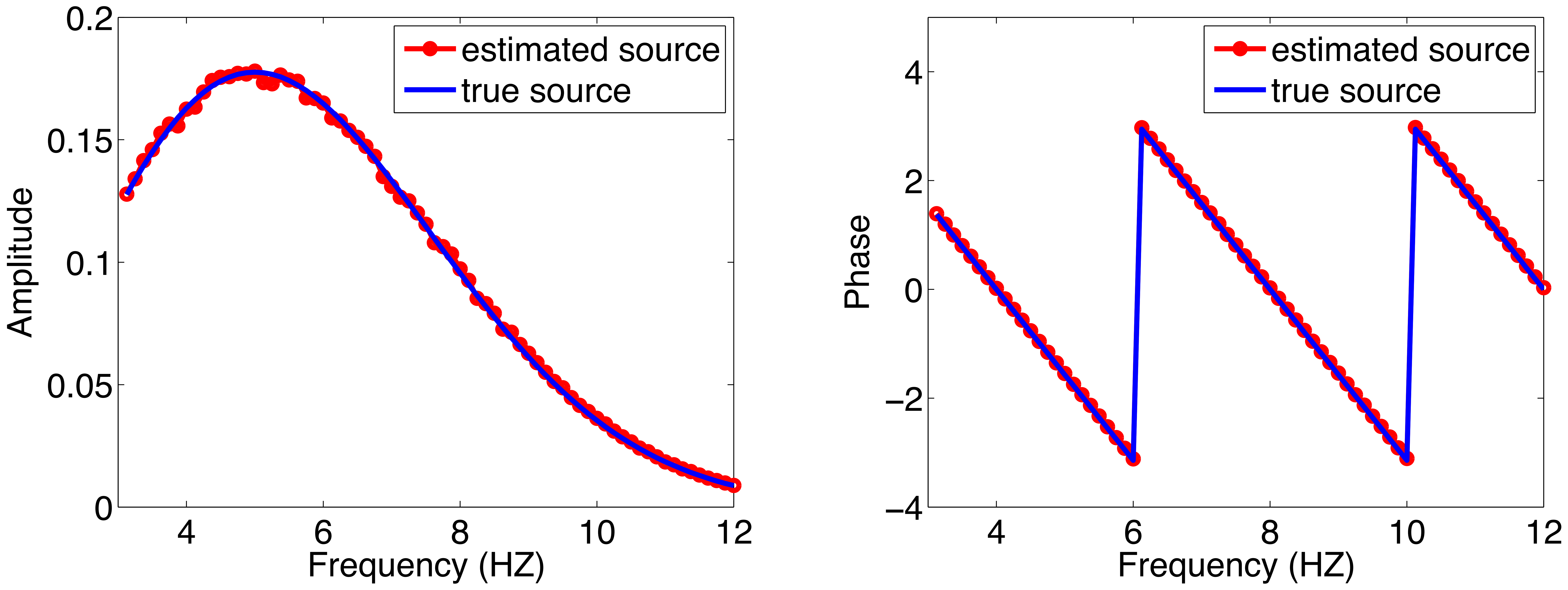
Fast SPLSM w/ source estimation

– estimated source



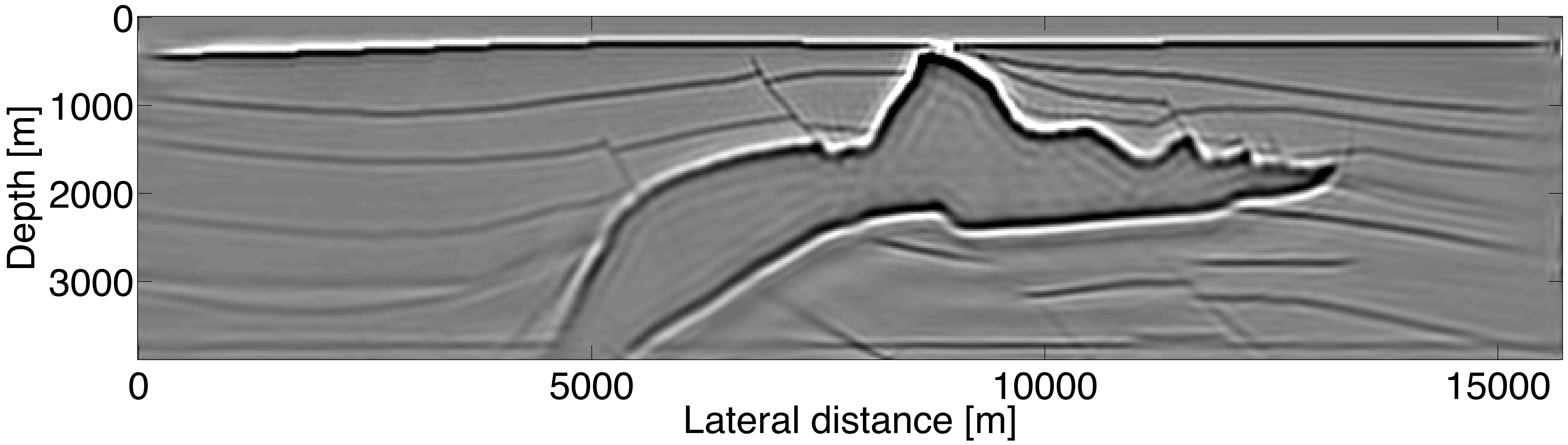
Fast SPLSM w/ source estimation

– estimated source



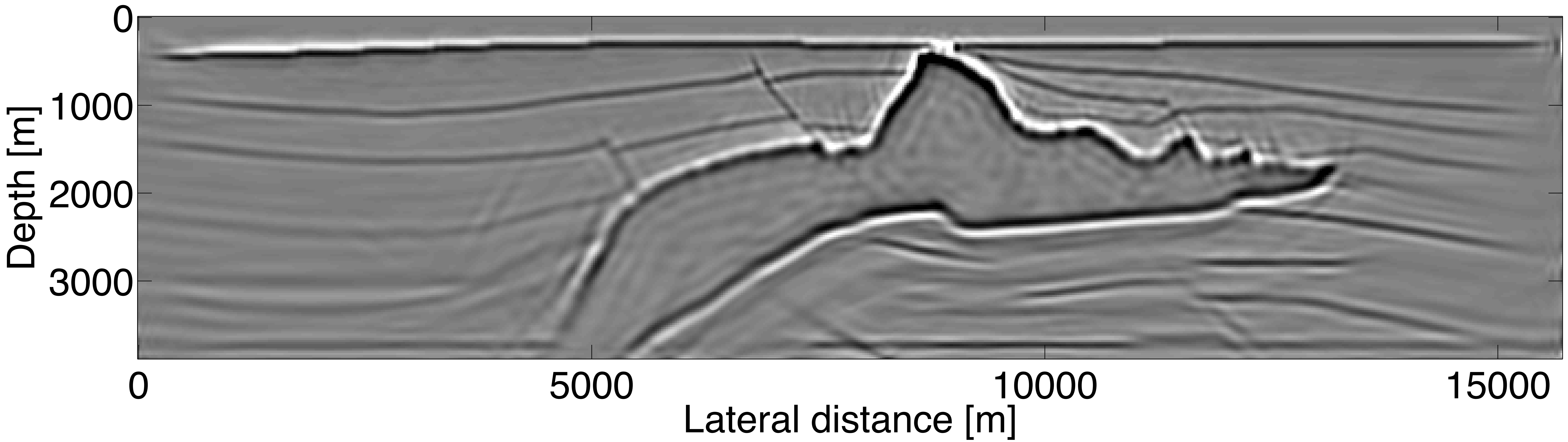
Fast SPLSM w/ source estimation

– 90 iterations, each w/ 4 frequencies/64sim. shots w/ true source



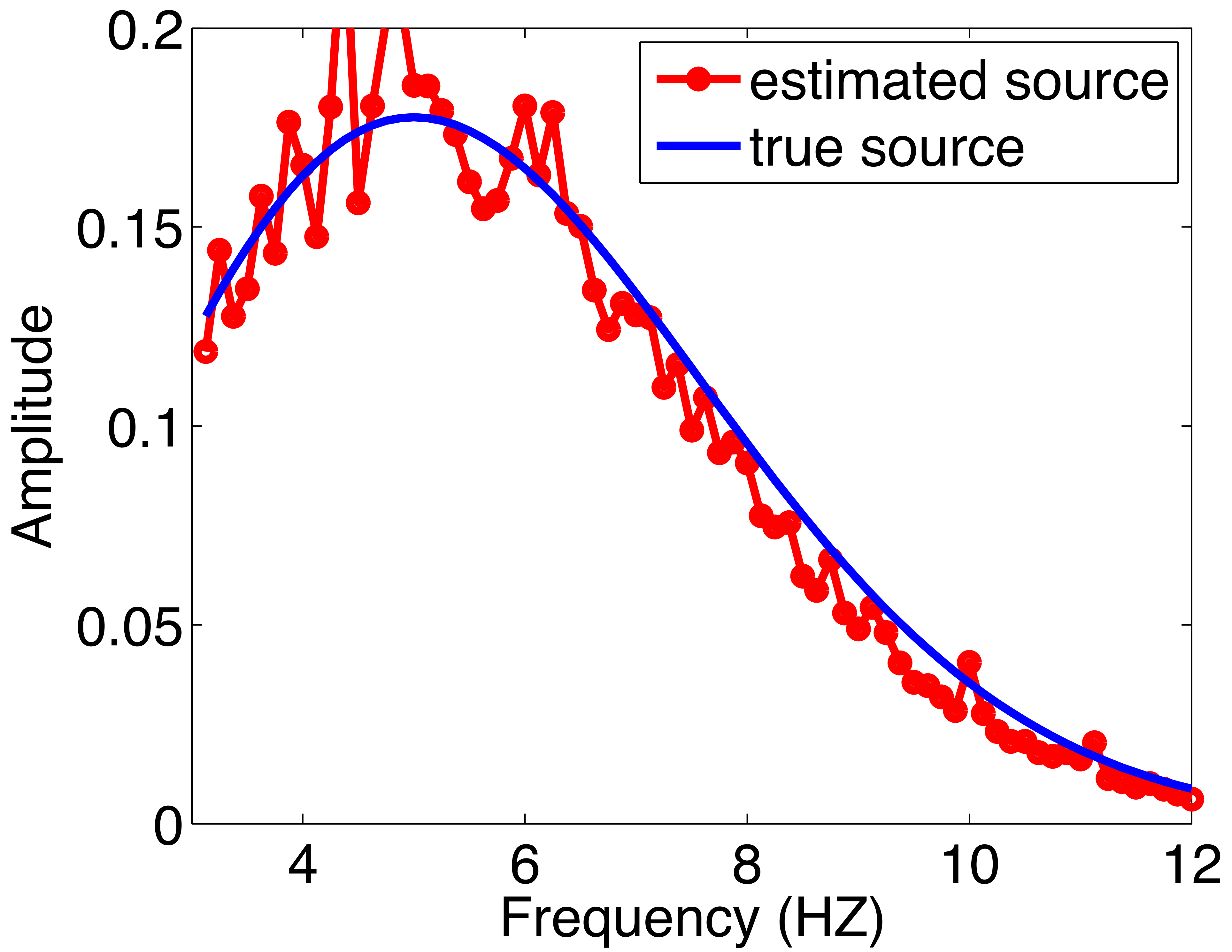
Fast SPLSM w/ source estimation

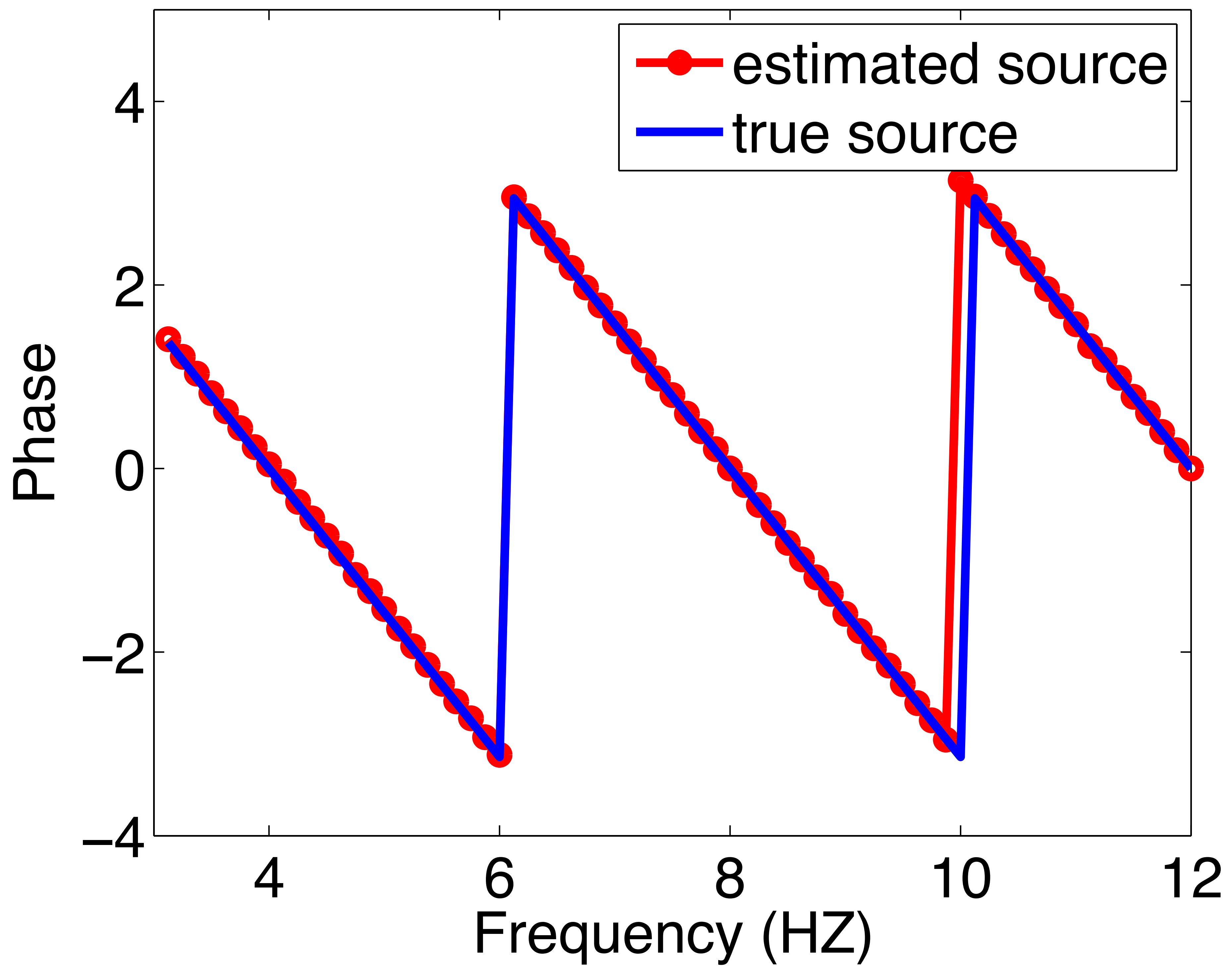
– estimated source



Fast SPLSM w/ source estimation

– estimated source





Observations

Inversions can be carried out at cost (no. of iterations x batch size) of ~ 1 RTM

For known source function:

- quality best for intermediate batch size & no. of iterations
- results for randomly selected and simultaneous sources are of similiar quality
- offers flexibility for parallelism

For unknown source function

- source function estimated best when no. of frequencies it not too low
- similar quality to when source function is known

SPLSM in time-domain

Algorithm is directly applicable in time domain as well:

- simply need Born modeling operator and its adjoint

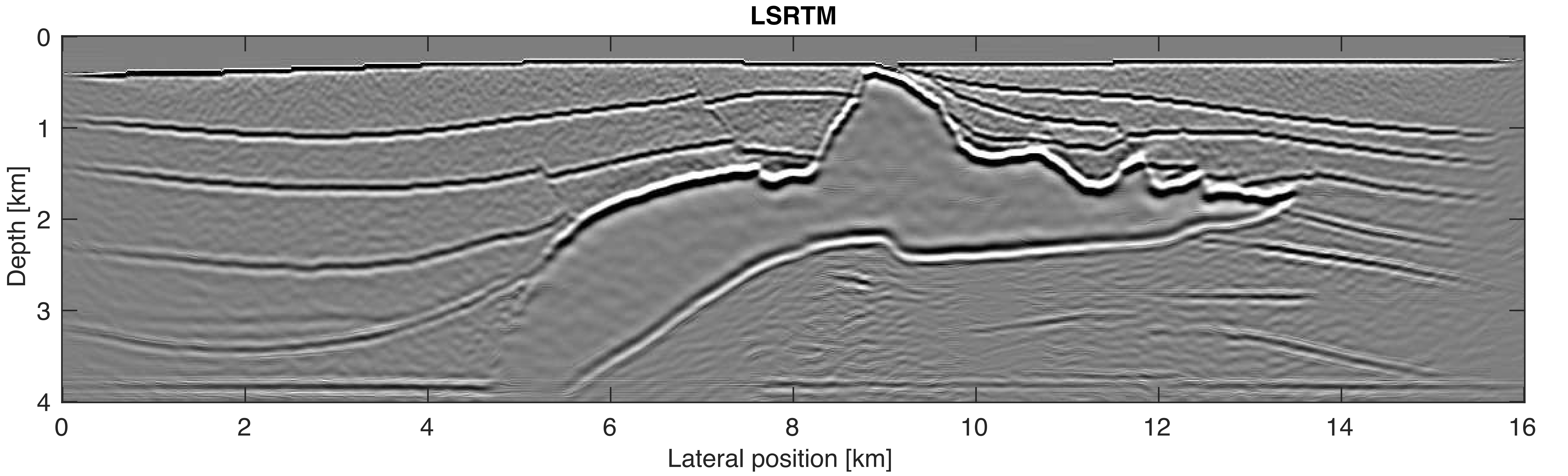
Data:

- 320 sources and receivers
- 5 seconds recording time, 10 Hz peak frequency
- $\delta\mathbf{d} = \mathbf{F}(\mathbf{m} - \mathbf{m}_0)$ inverse crime data

Experiments:

- one pass through the data
- one randomly selected source at a time
- no source estimation

SPLSM in time-domain



Bottom line
– what you need

Access to $\{\mathbf{A},\ \mathbf{A}^H\}$ or $\{\mathbf{A}^H,\ \mathbf{A}^H\mathbf{A}\}$

- migration, demigration or migration, Gauss-Newton Hessian
- norms of residual & gradient for step lengths

Ability to subsample data

- randomized simultaneous shots or randomly selected shots
- randomized traces (source/receiver pairs) in Kirchoff migration

Some idea of the maximum of $\mathbf{A}^H_{r(k)}\mathbf{b}_{r(k)}$ (for choosing λ)

Conclusions and extensions

Linearized Bregman algorithm:

- simple, converges and very few tuning parameters
- offers maximum flexibility for
    - finding right balance for parallelization over model and data space
    - extensions as source estimation, imaging with multiples
    - broad application for many other seismic problems as well: AVO, interpolation,...
- High resolution images at low cost comparable to regular RTM

Further extensions

- adaptive sampling strategies for source selection
- “online” inversion: can work with incoming stream of data
- LSRTM with Kirchhoff modeling/inversion

Acknowledgements

Thanks to all PhD students and Postdocs at SLIM!

This work was in part financially supported by the Natural Sciences and Engineering Research Council of Canada via the Collaborative Research and Development Grant DNOISEII (375142--08). This research was carried out as part of the SINBAD II project with support from the following organizations: BG Group, BGP, CGG, Chevron, ConocoPhillips, DownUnder GeoSolutions, Hess, ION, Petrobras, PGS, Schlumberger, Statoil, Sub Salt Solutions, Total SA, and Woodside.