Compressive sensing and sparse recovery in exploration seismology

Felix J. Herrmann

SLIM

Seismic Laboratory for Imaging and Modeling

the University of British Columbia

Compressive sensing and sparse recovery in exploration seismology

Felix J. Herrmann
SLIM team

SLIM
Seismic Laboratory for Imaging and Modeling
the University of British Columbia

Drivers

Recent technology push calls for collection

- high-quality *broad-band* data *volumes* (>100k channels)
- *larger* offsets & *full* azimuth

Exposes vulnerabilities in our *ability* to control

- *acquisition* costs / time
- *processing* costs / time

Drivers cont'd

Complexity of *inversion* algorithms *exposes* the "*curse of dimensionality*" in

- **sampling**: *exponential* growth of # *samples* for *high* dimensions
- **optimization**: *exponential* growth of # parameter *combinations* that need to be evaluated to minimize our *objective* functions

Today's agenda

Overview of

- basics of *exploration* seismology
- type of problems we encounter

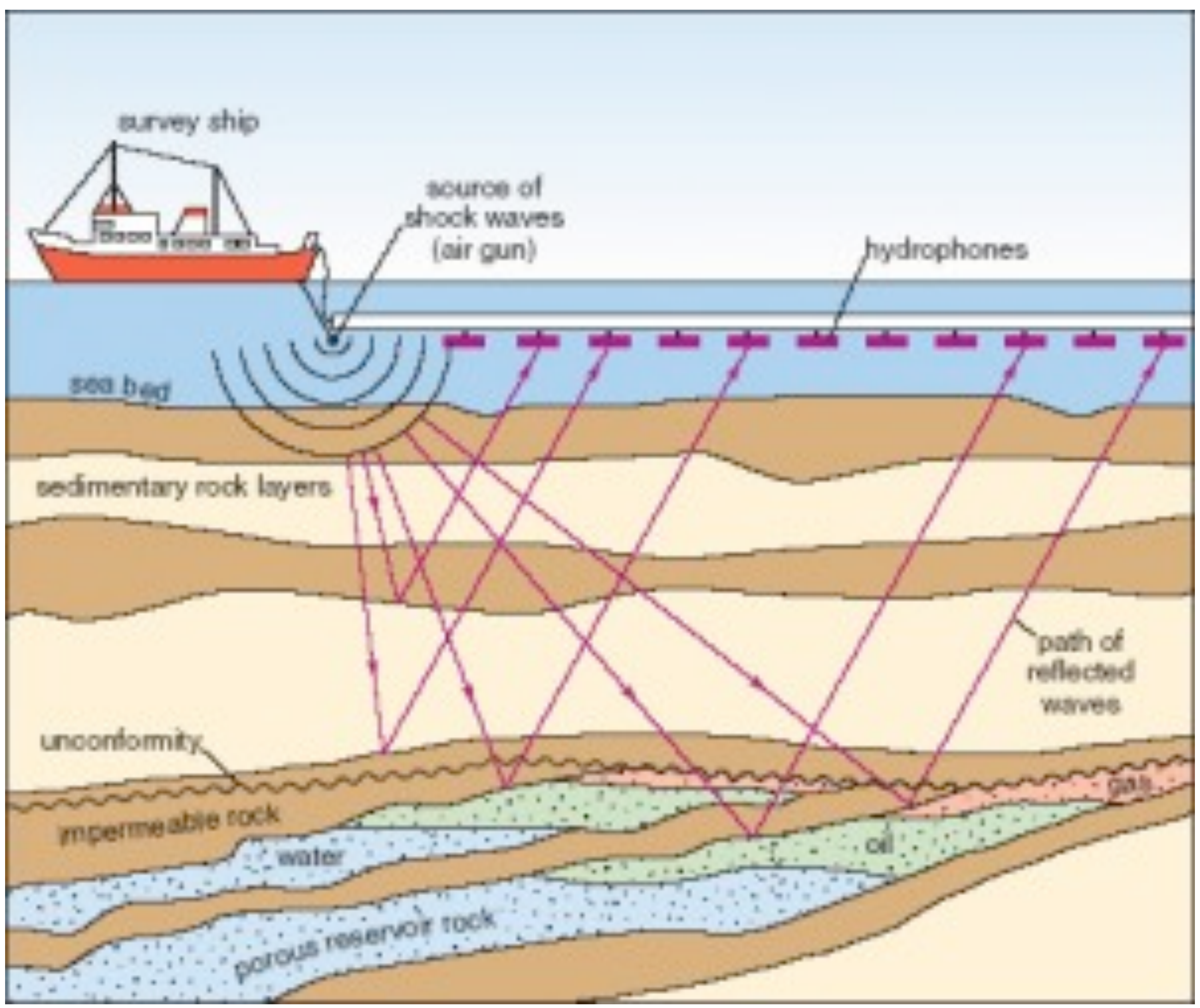
Examples of CS & convex optimization in seismic acquisition

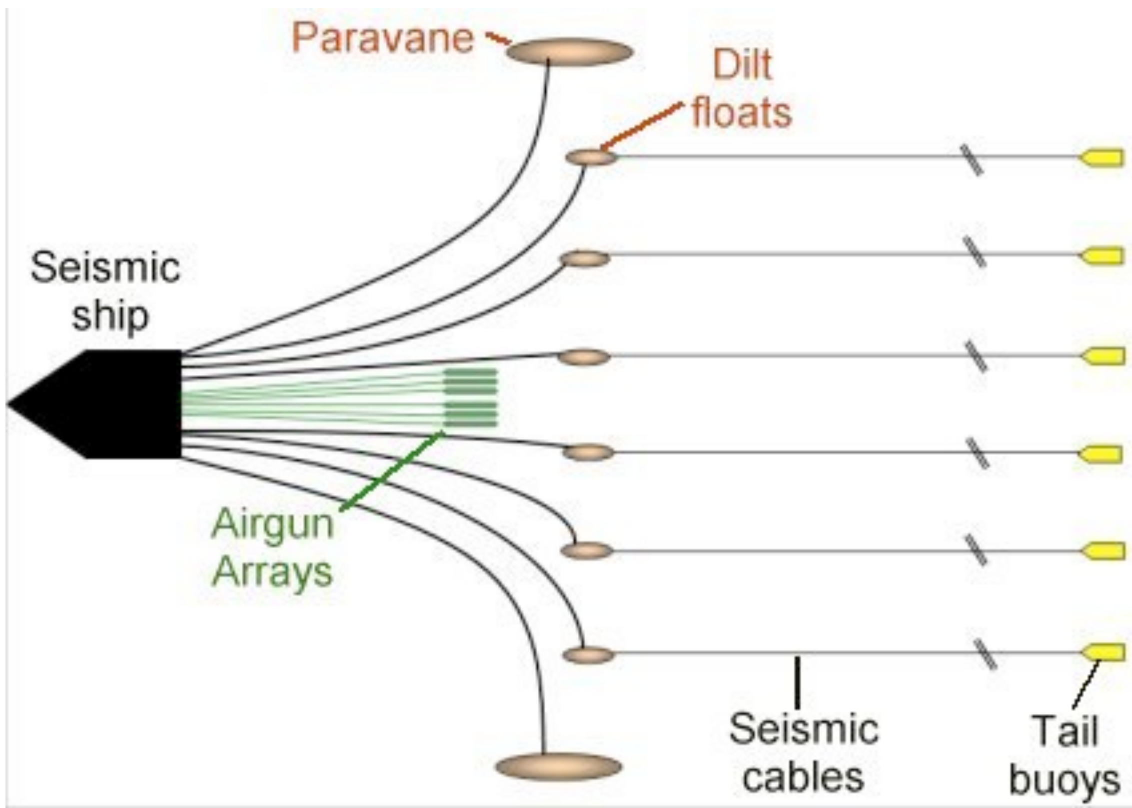
- successes & challenges

Dimensionality reduction in wave-equation based inversion

- "poor man's" approximate message passing

Basics seismic acquisition [Marine]





http://geomaticsolutions.com/seismic-surveys/

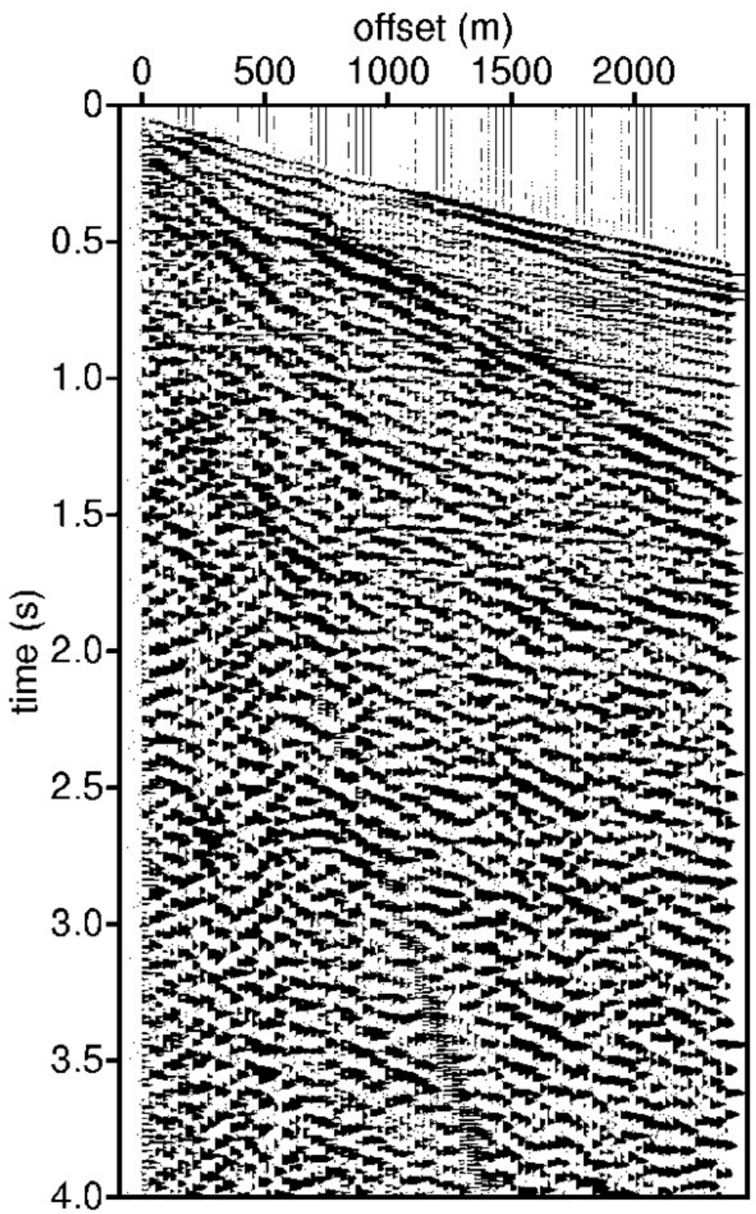
Geco Eagle over Oslo

Dynamite

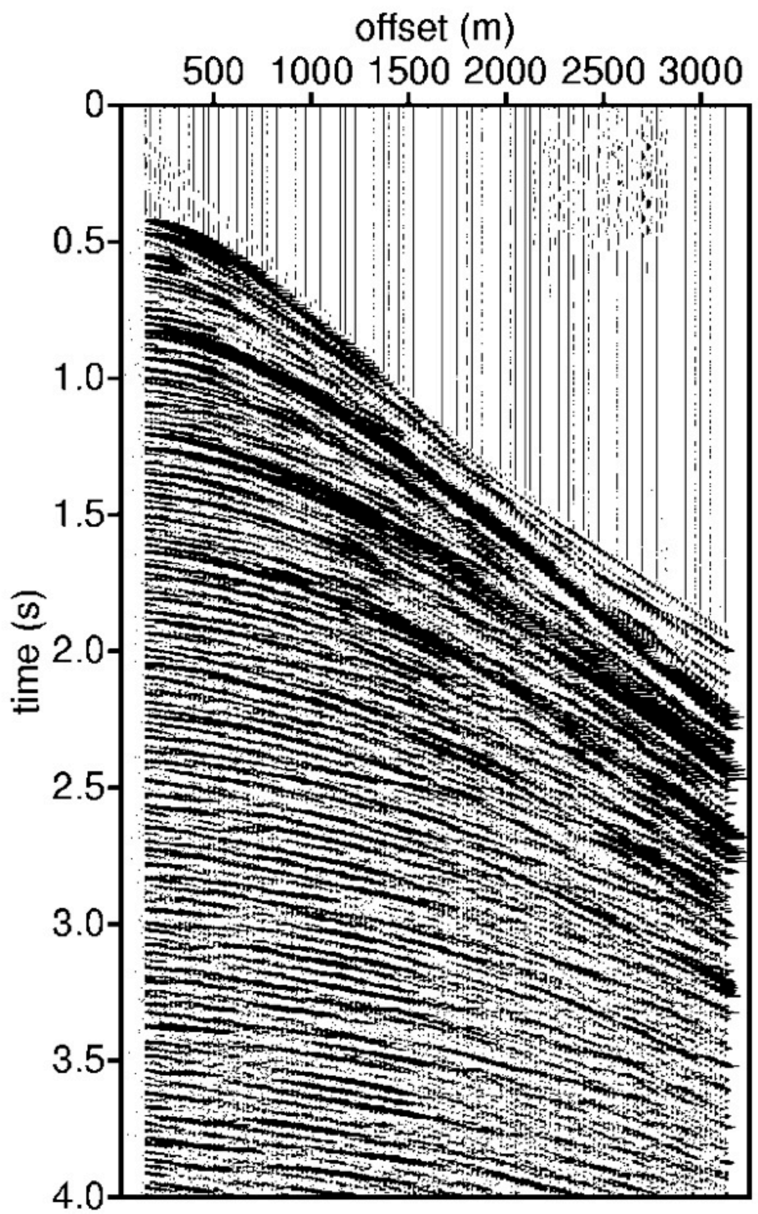
Dynamite

VibroSeis

Examples of records

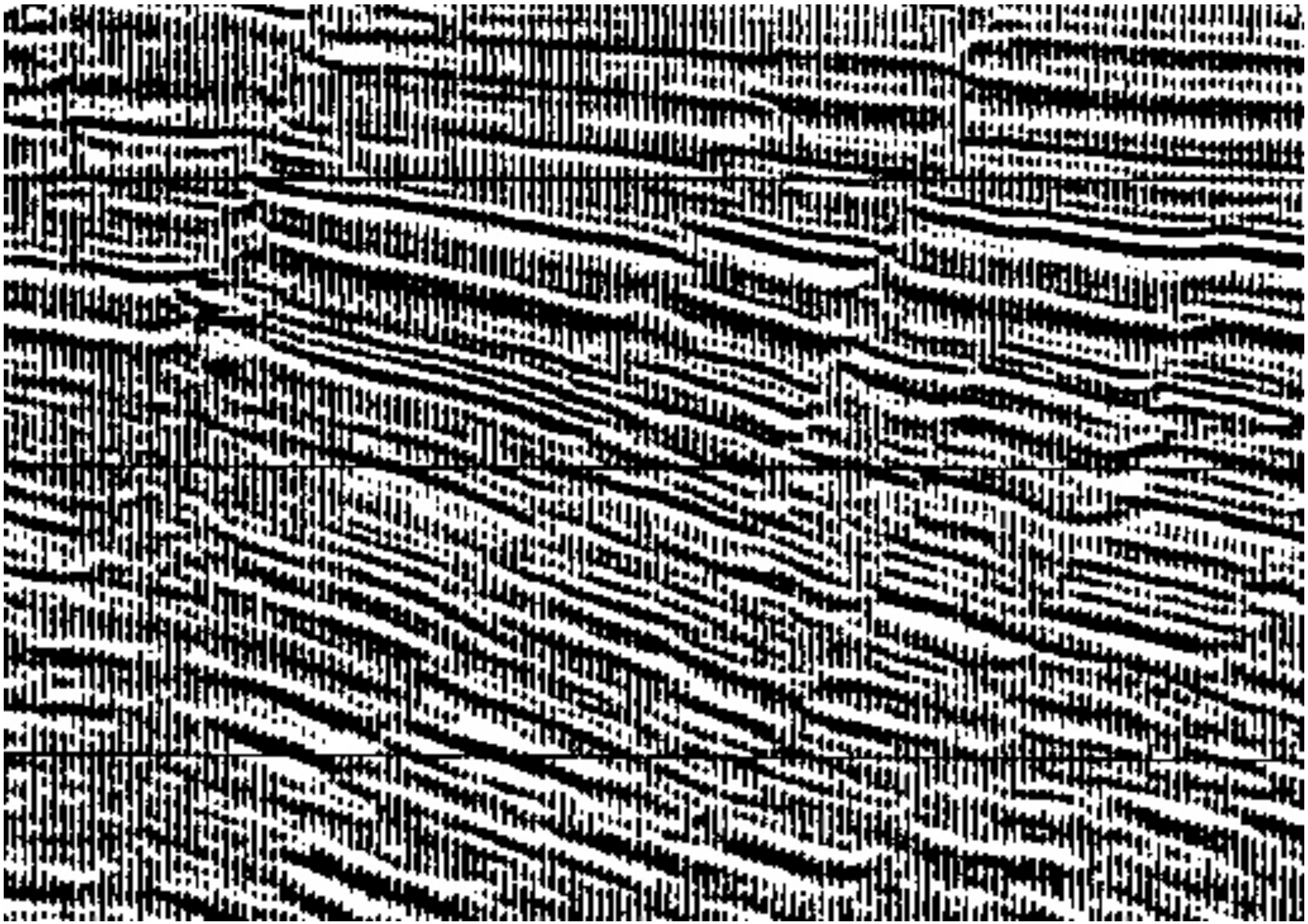


On land: Vibroseis



At sea: airguns

What is in the subsurface?

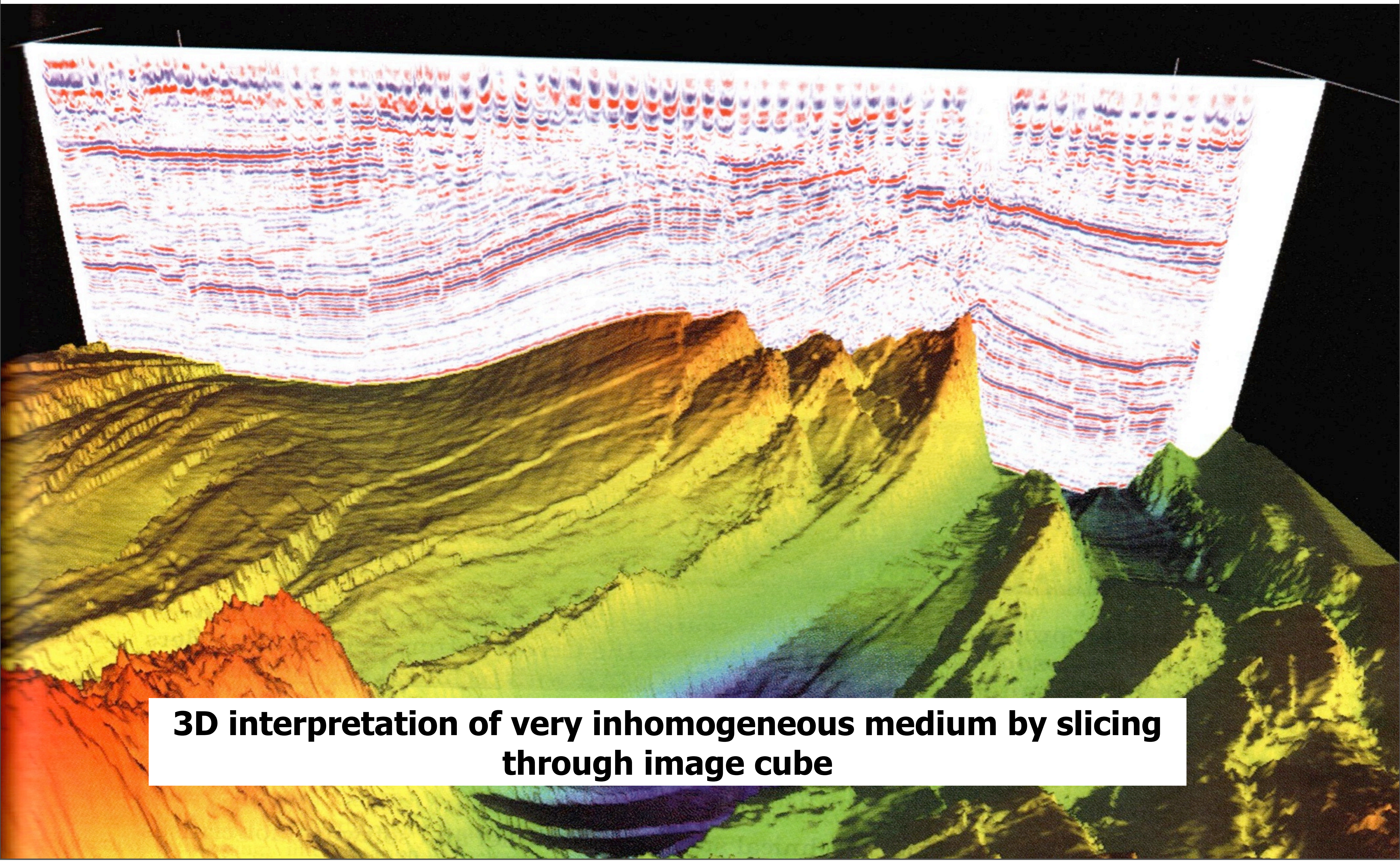


Detail of seismic image containing faults

What is in the subsurface?

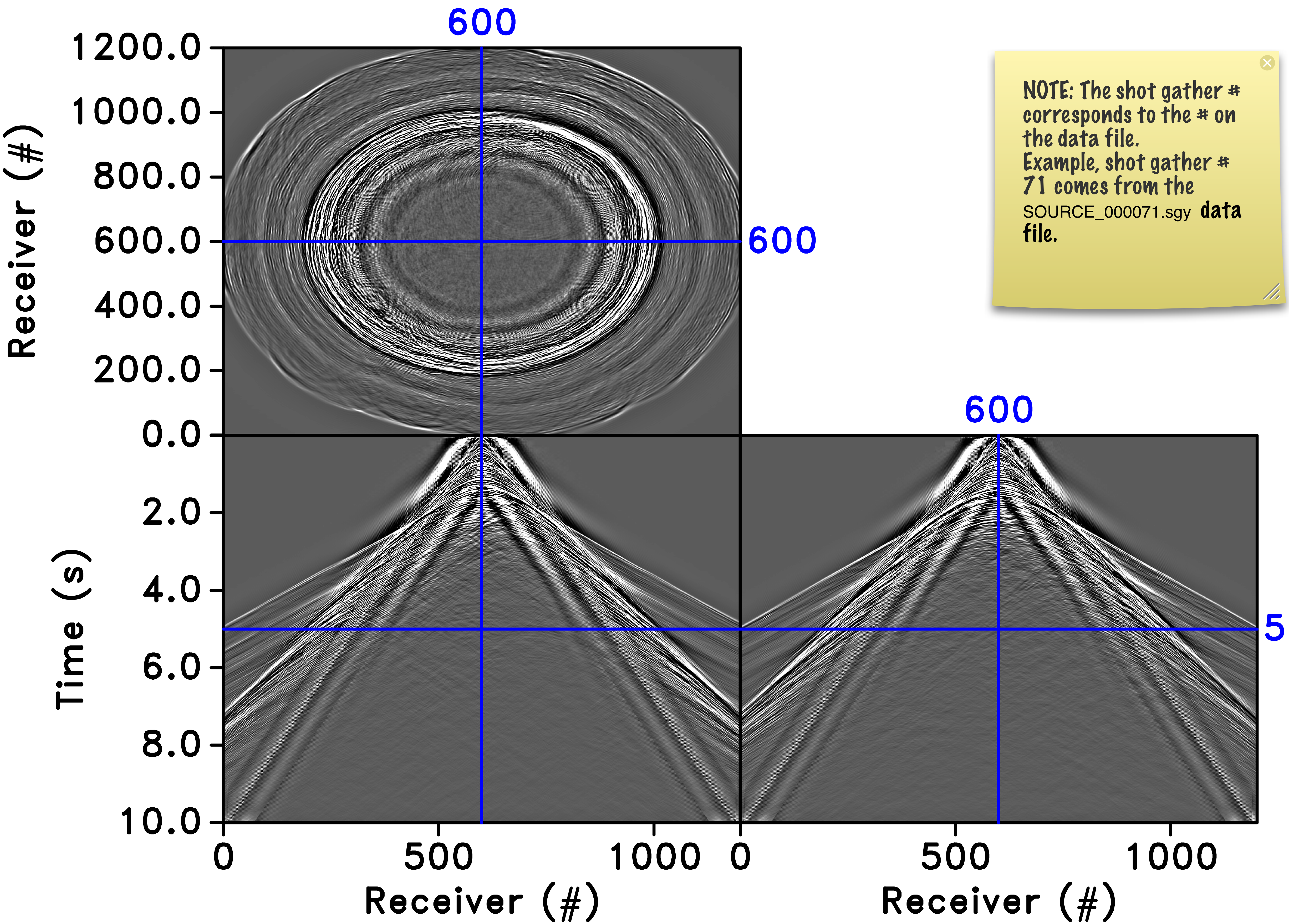
“Outcrop” with fault-blocks

3D seismic image interpretation



3D interpretation of very inhomogeneous medium by slicing through image cube

Common Shot Gather # 71: Rx = 600, Ry = 600



NOTE: The shot gather # corresponds to the # on the data file. Example, shot gather # 71 comes from the SOURCE_000071.sgy data file.

Problems

Seismic acquisition is "costly"

Difficult to acquire *complete* data volumes in 4 *spatial* dimensions

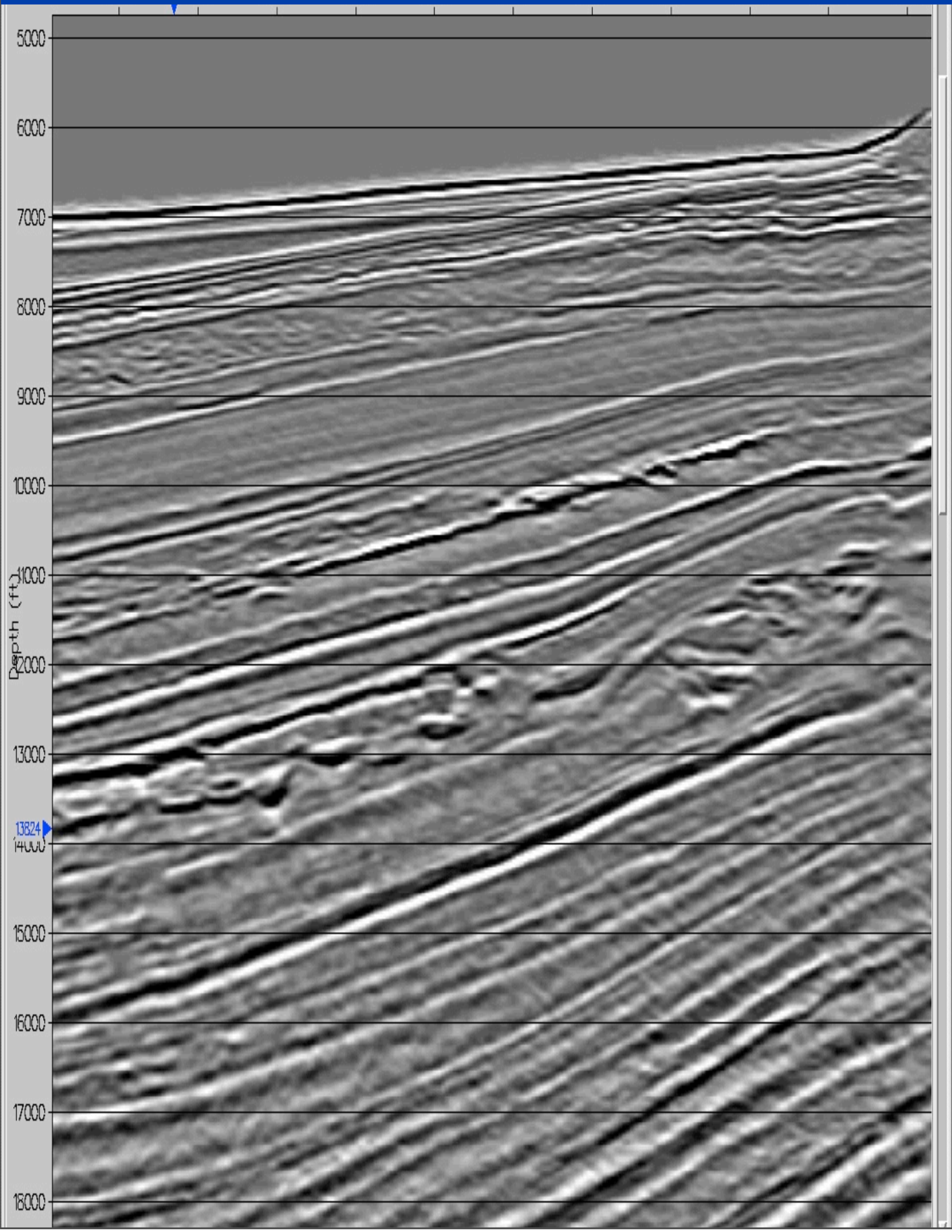
Physical constraints, noise, obstacles...

Inversion codes call for *more* and *higher* quality data

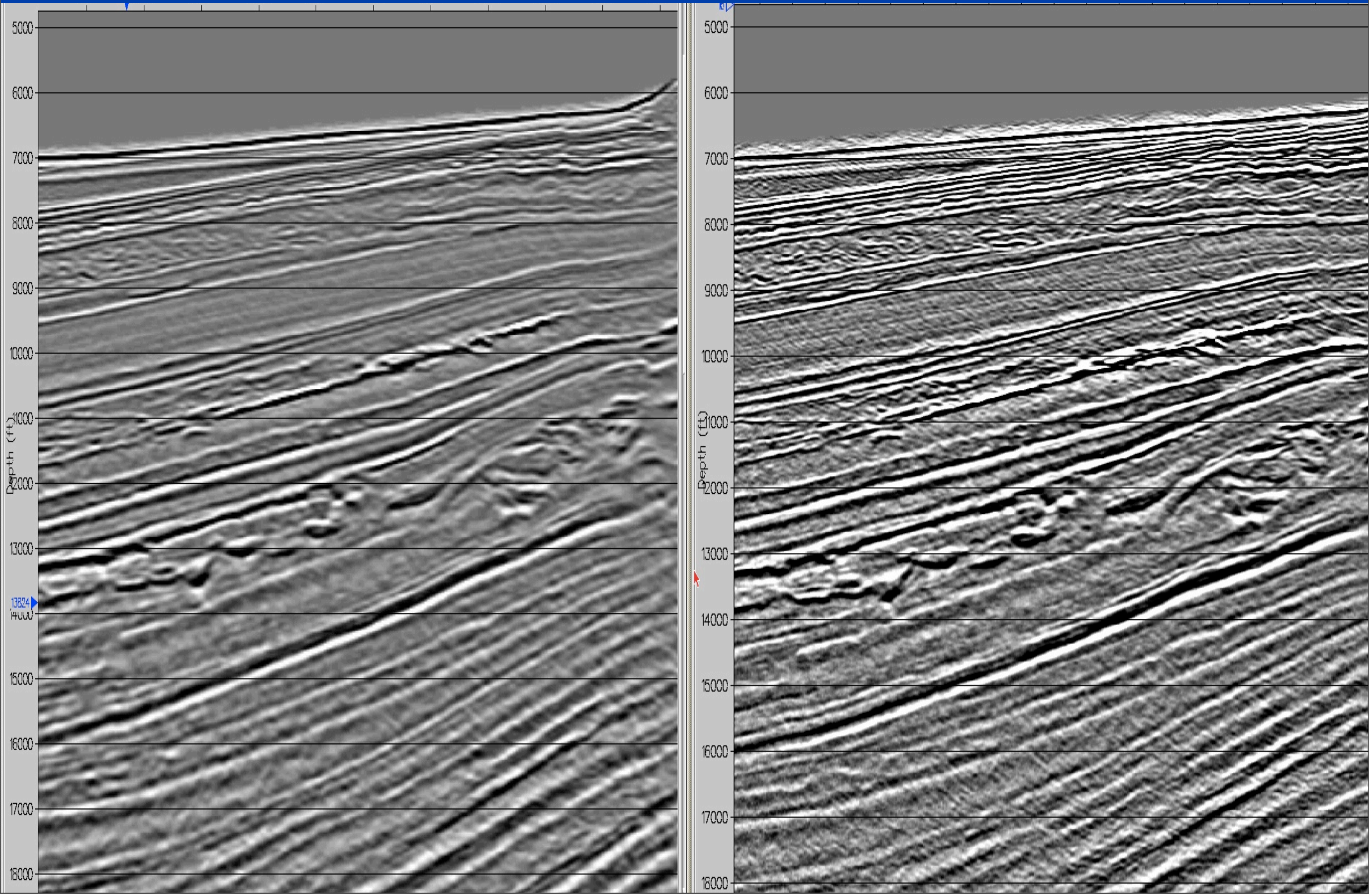
Seismic data volumes are becoming *excessively* large

Exposes vulnerabilities in our ability to compute our way out of this ...

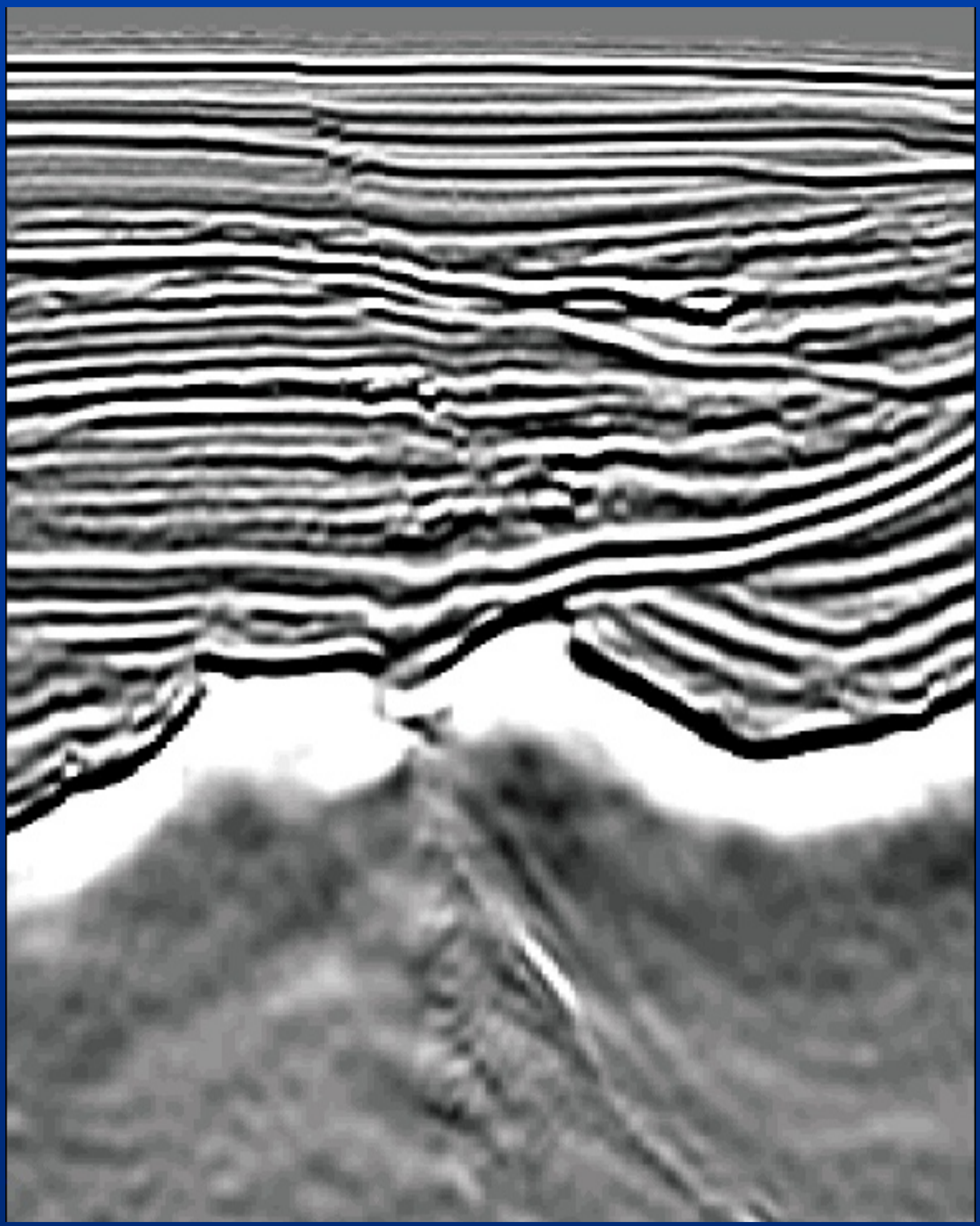
Migration output 12.5 m x 30 m and 12.5 m x 15 m



Migration output 12.5 m x 30 m and 12.5 m x 15 m

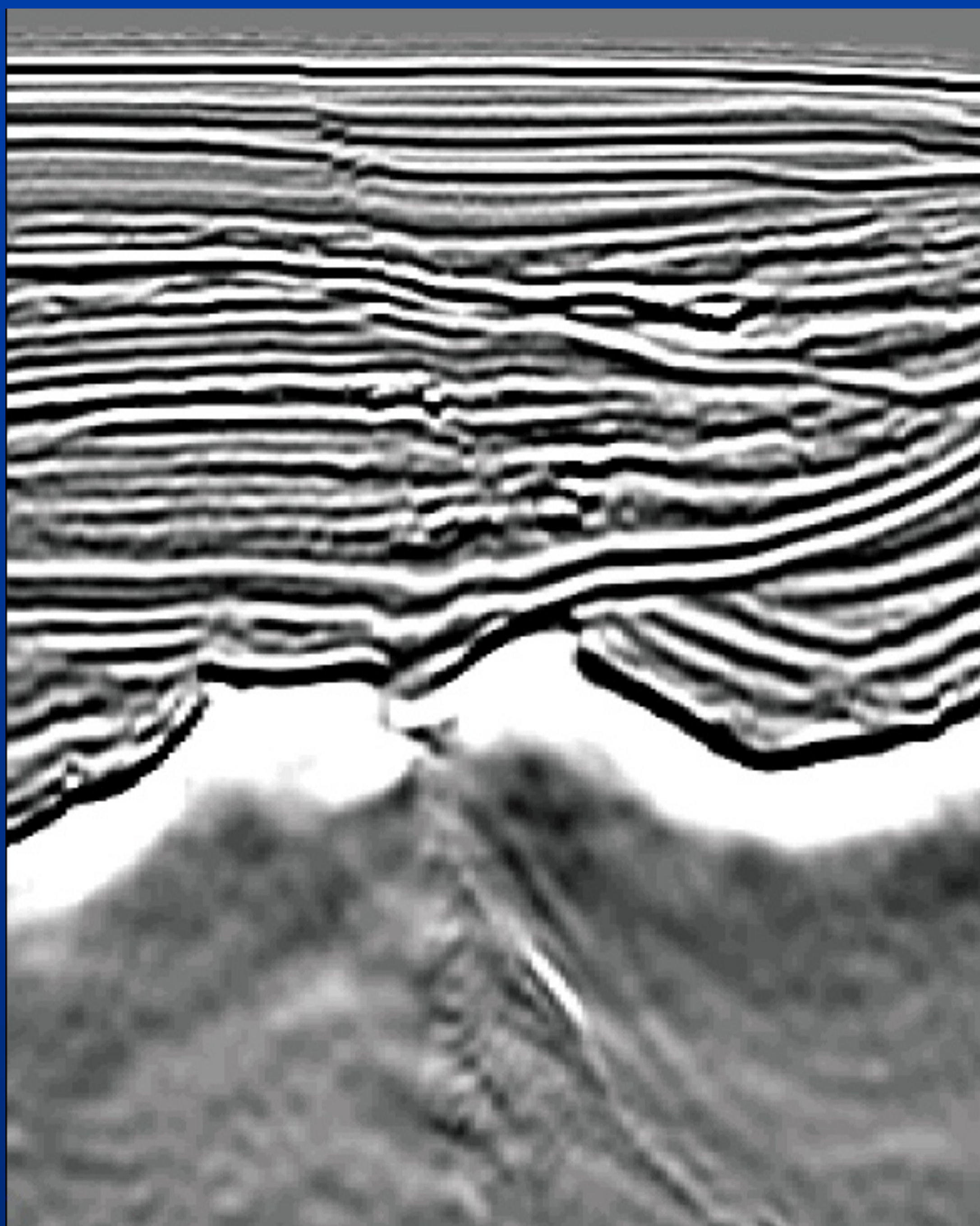


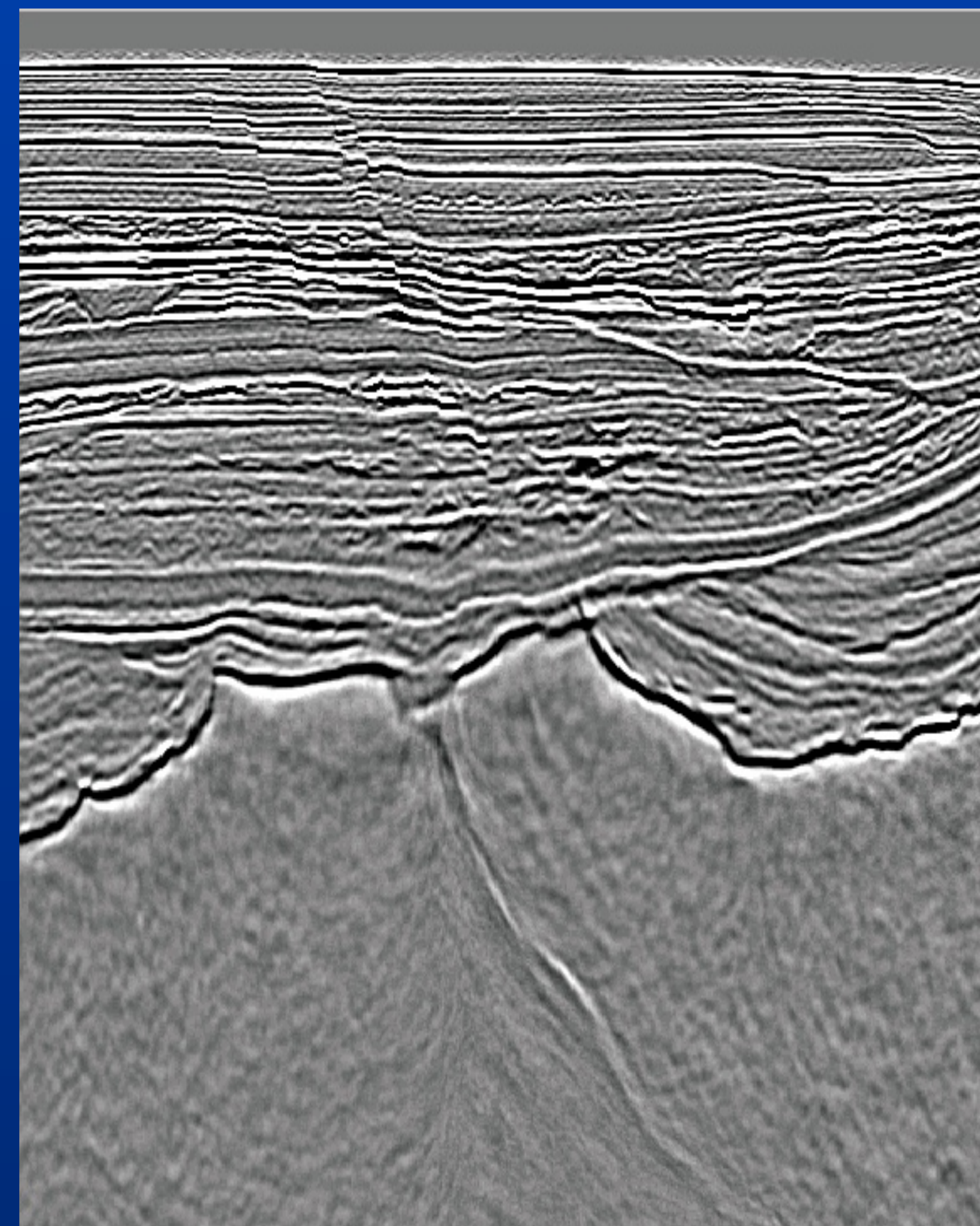
Migration output at 25 m x 30 m and 10 m x 10 m



Courtesy of BHP Billiton,Hess Corp,Repsol-YPF

Migration output at 25 m x 30 m and 10 m x 10 m



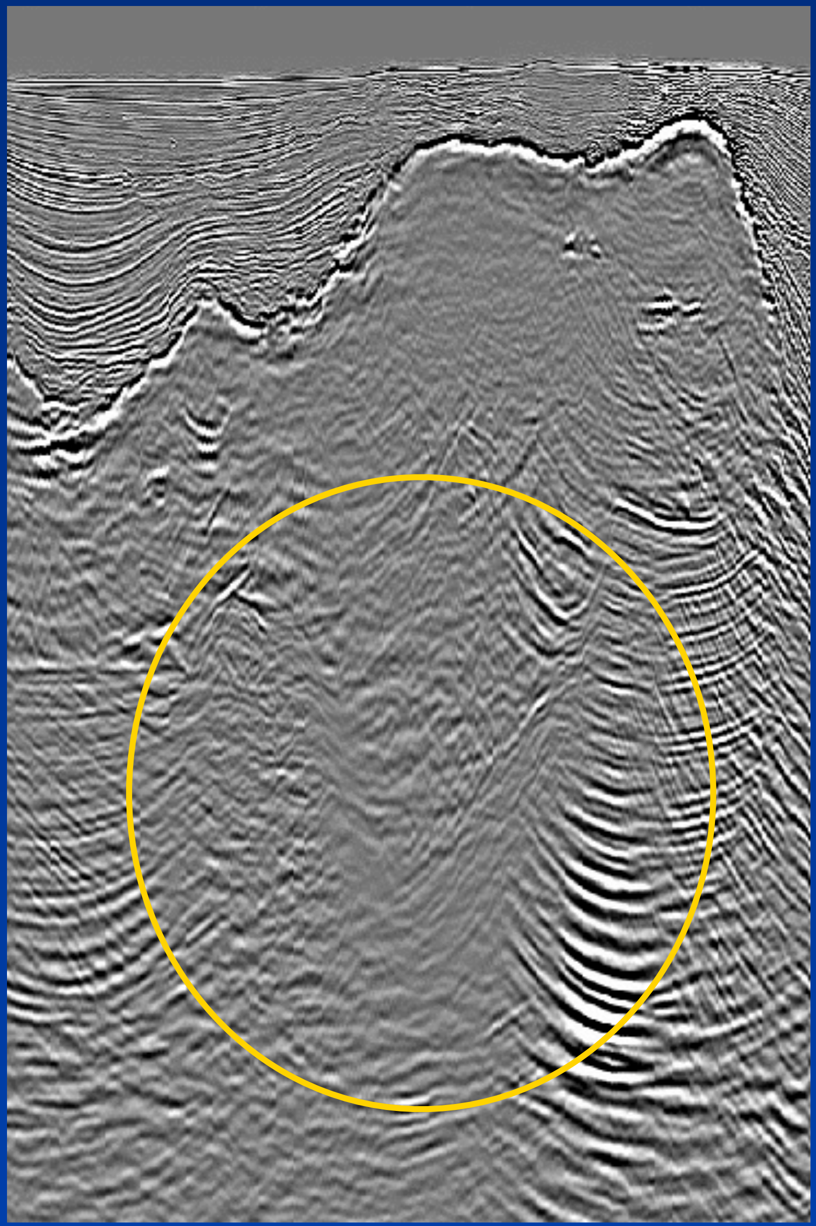


Courtesy of BHP Billiton,Hess Corp,Repsol-YPF

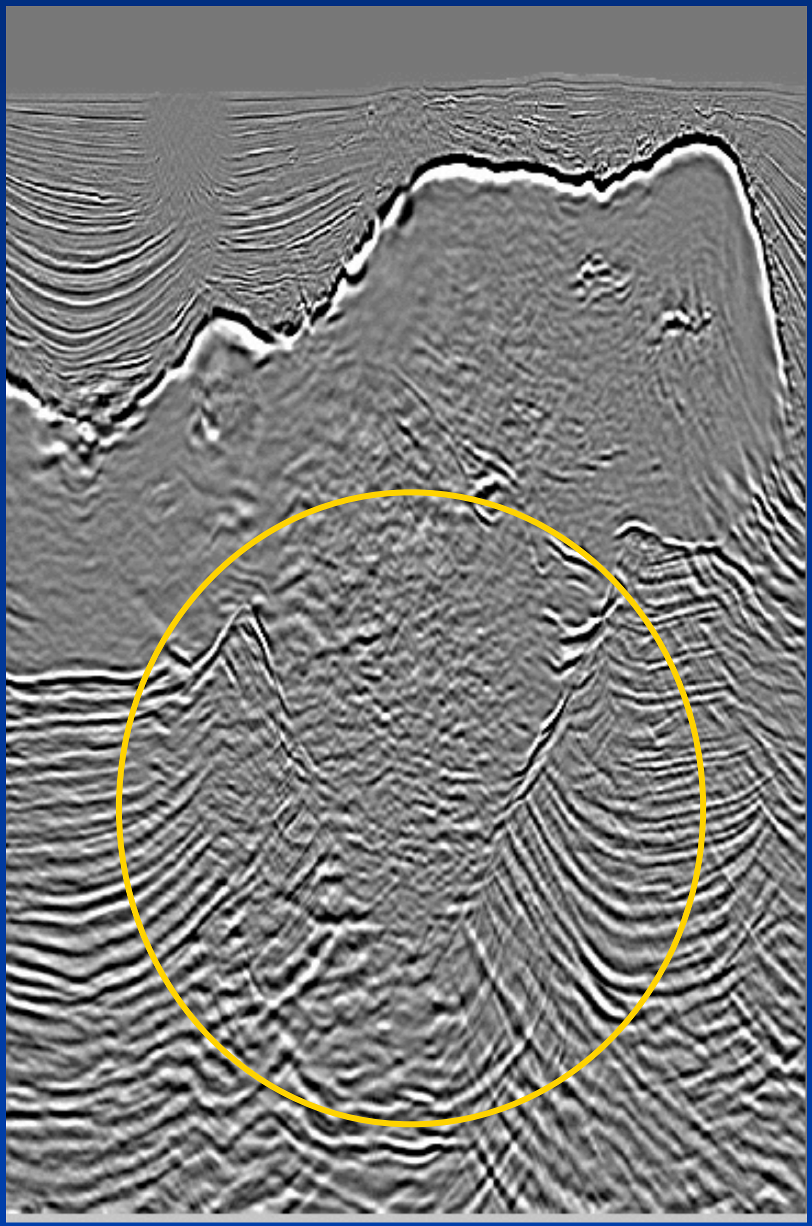
WesternGeco

Narrow Azimuth vs. Wide Azimuth

NAZ

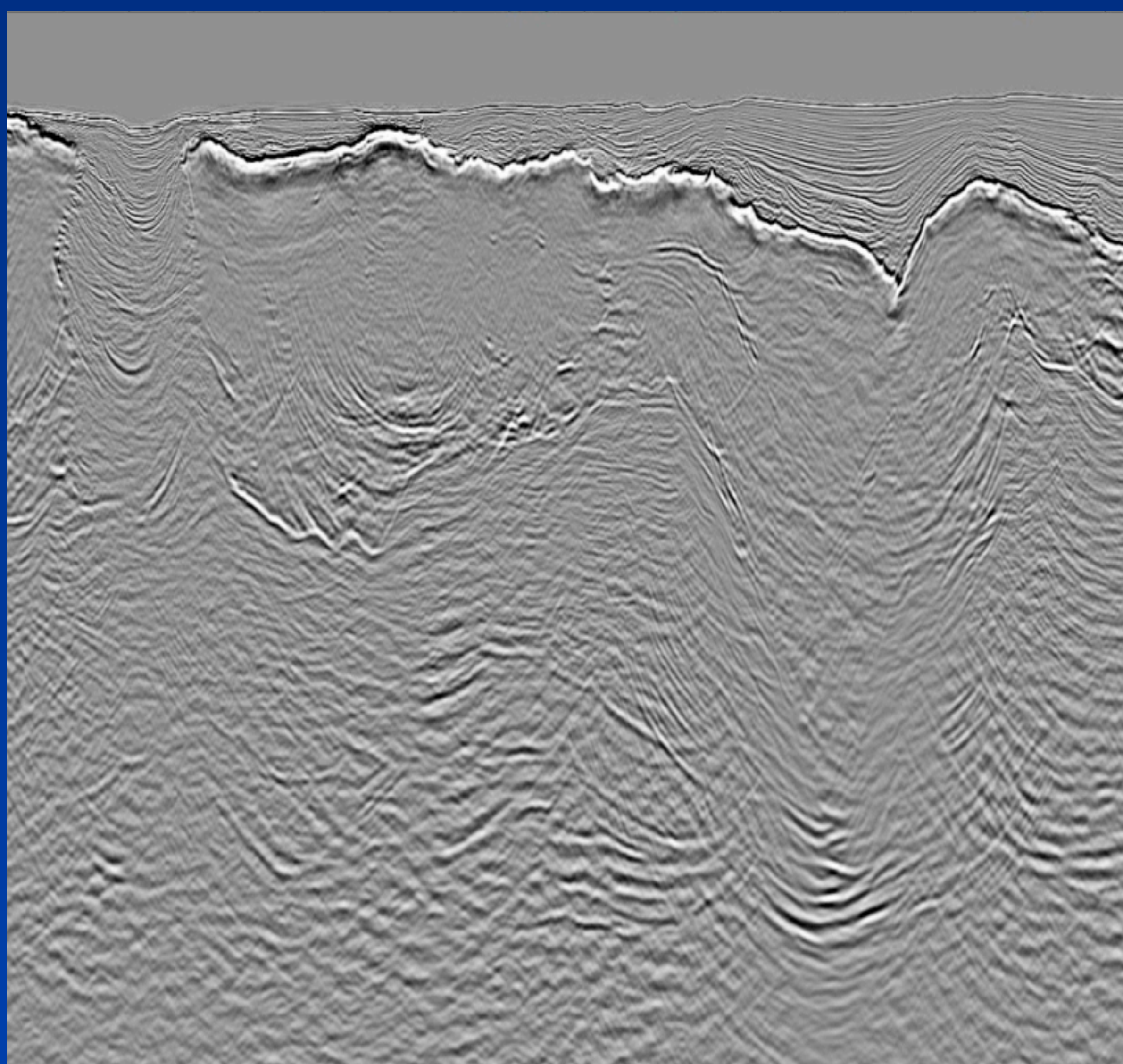


WAZ

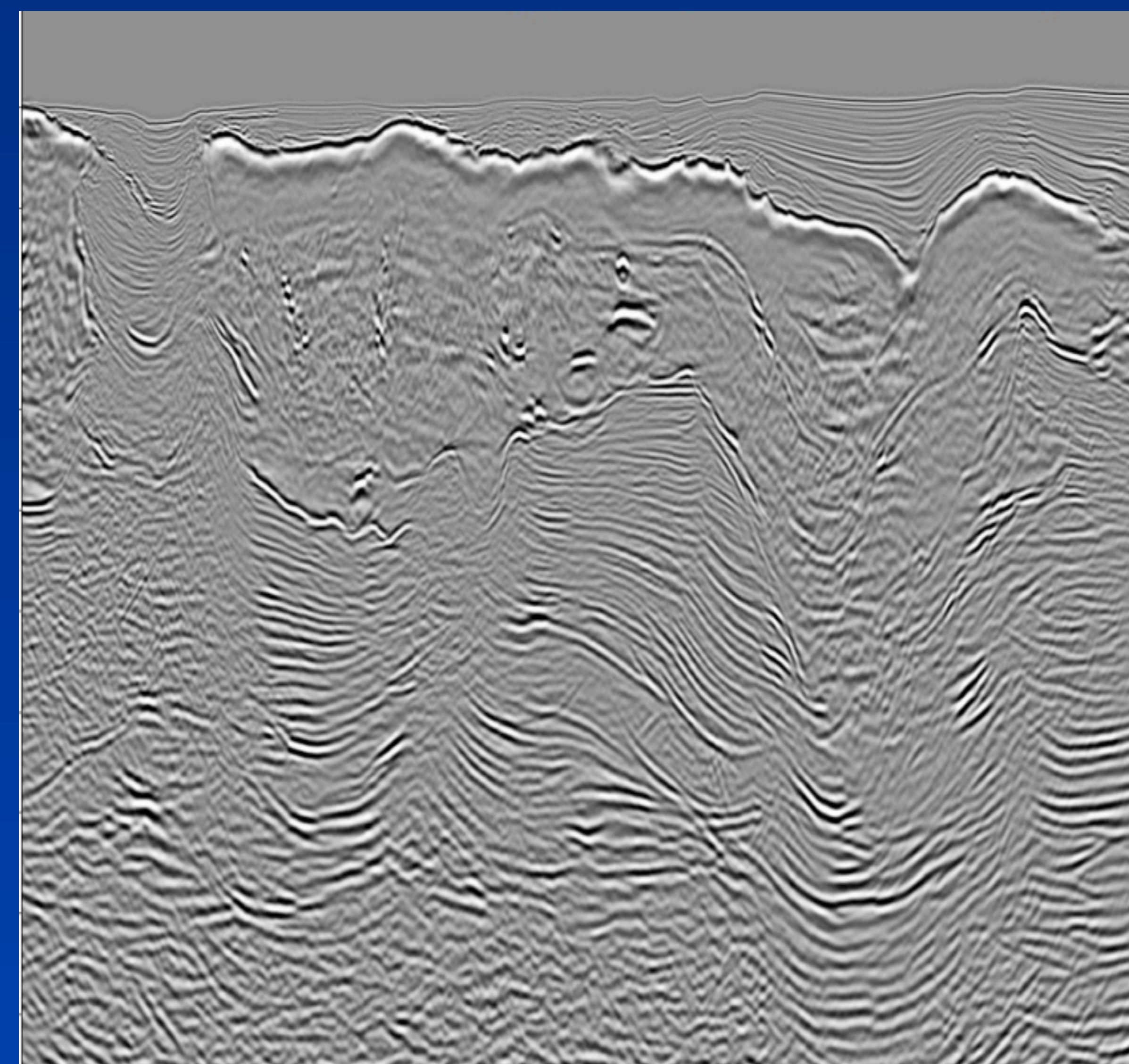


Subsalt imaging improvements from 2005 to 2010: GSMP, FWI, RTM

2005 technologies
NAZ/SRME/WEM

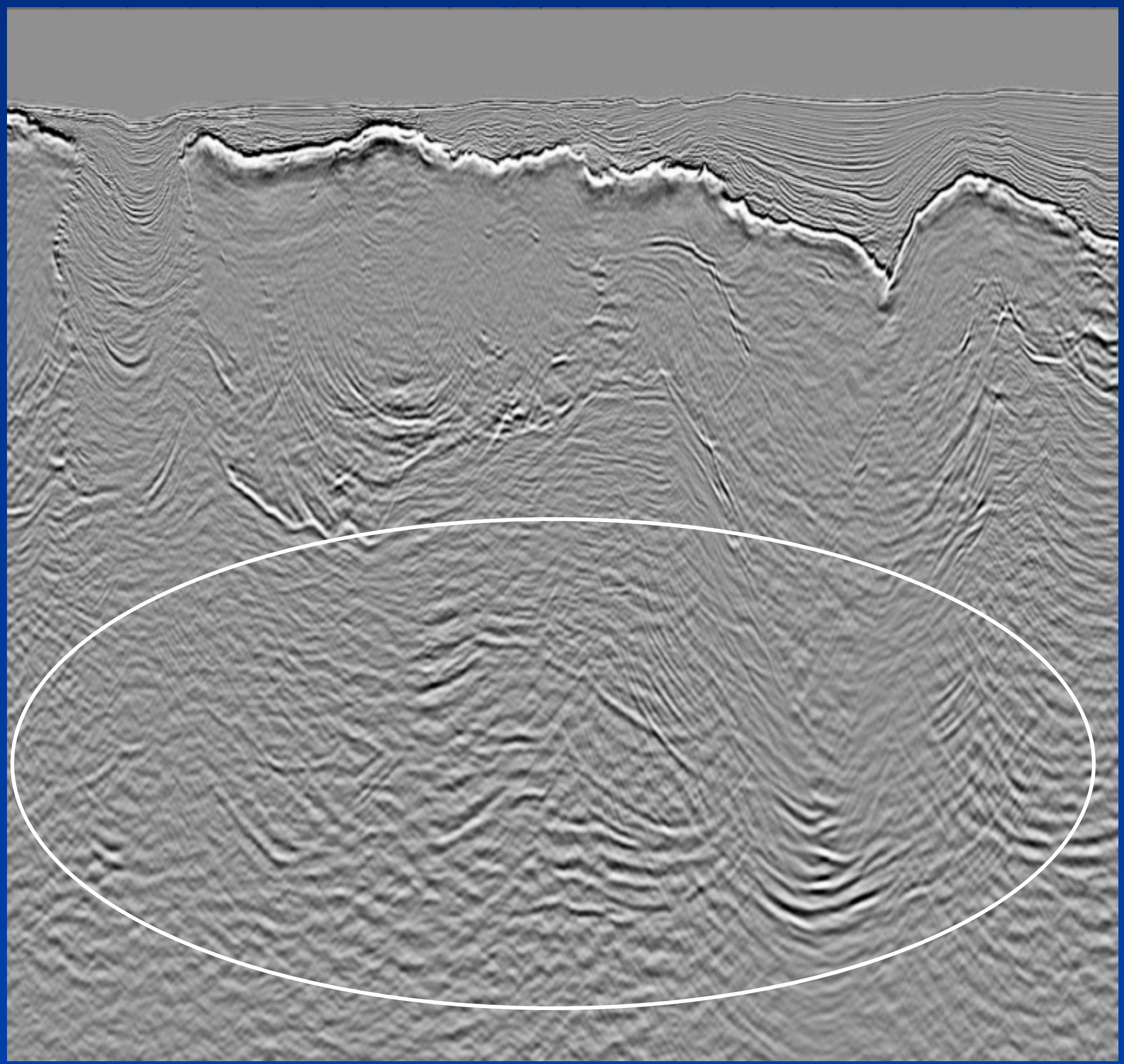


2010 technologies
WAZ/GSMP/FWI/RTM

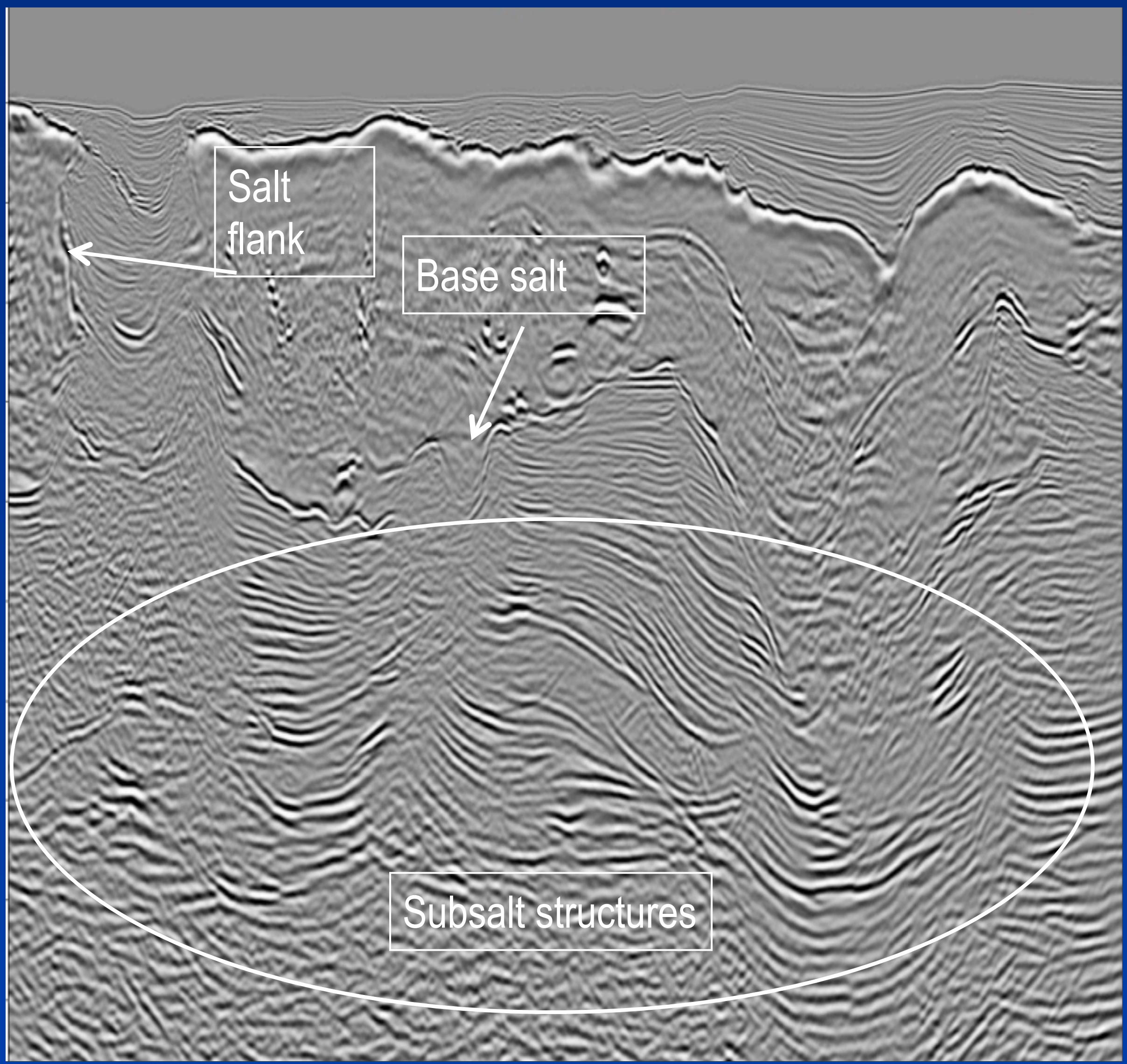


Subsalt imaging improvements from 2005 to 2010: GSMP, FWI, RTM

2005 technologies
NAZ/SRME/WEM



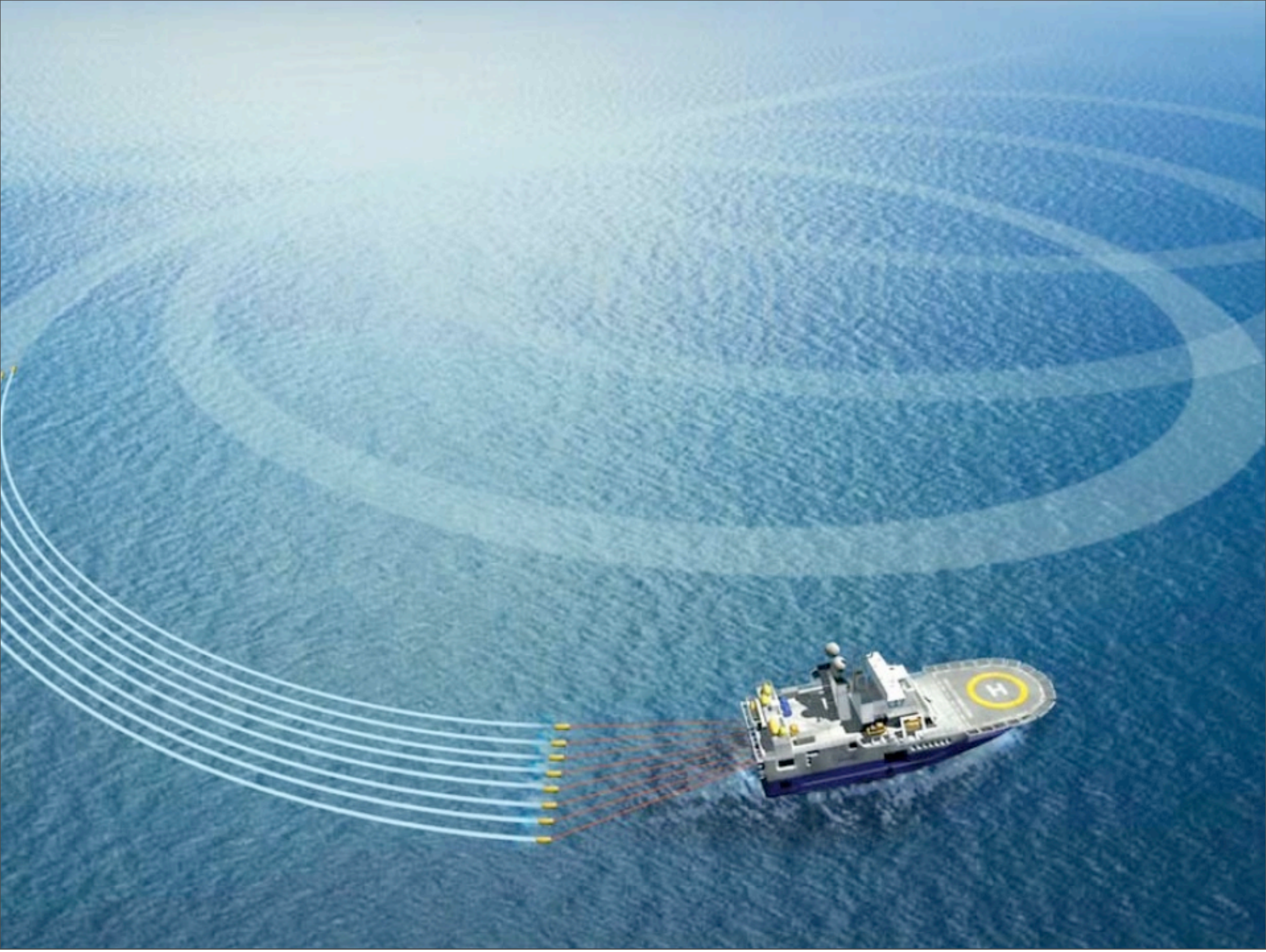
2010 technologies
WAZ/GSMP/FWI/RTM



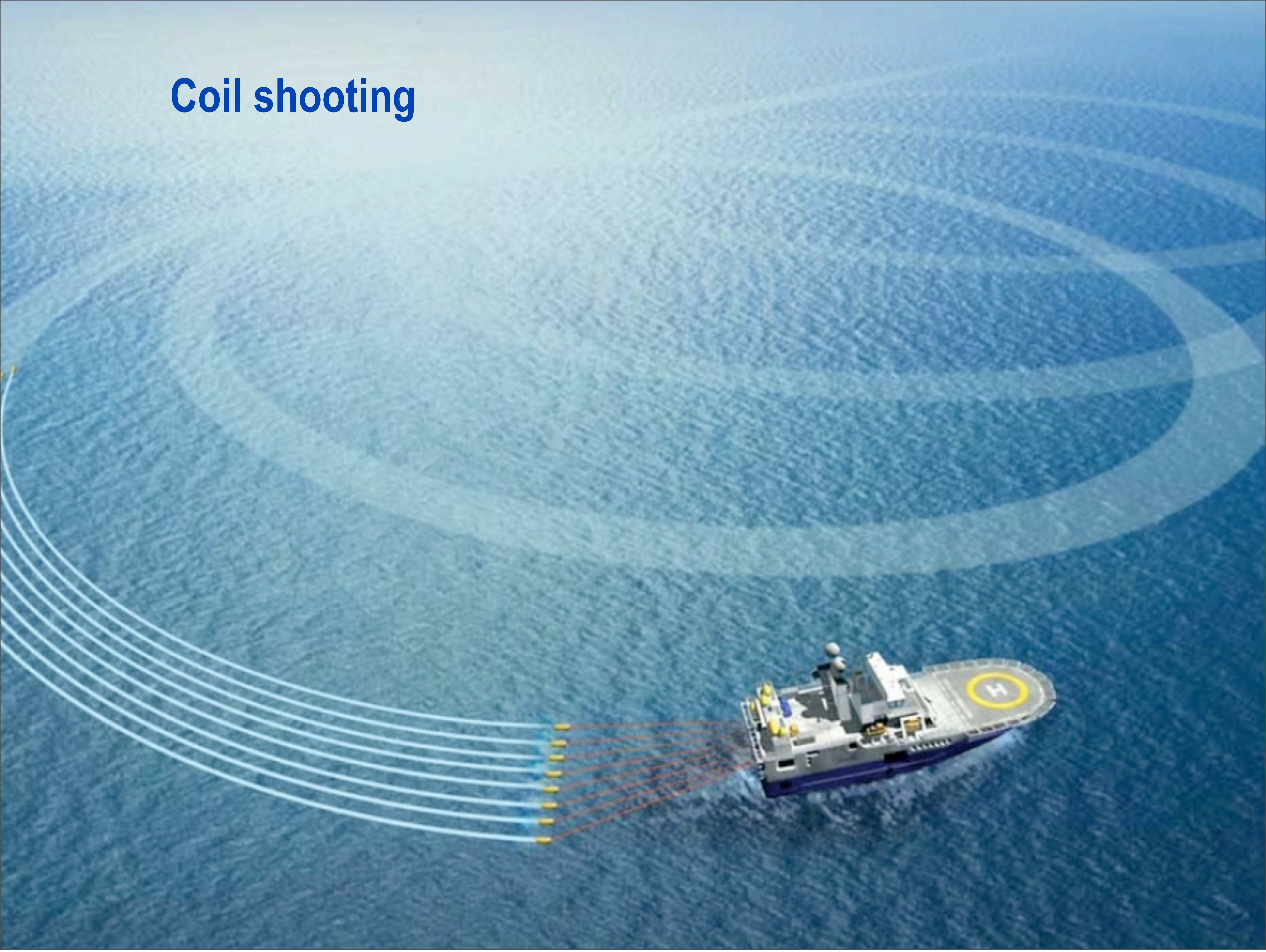
Our contributions

Proposal to *randomize* acquisition

- *random* source/receiver locations
- *random* time dithering in (simultaneous) source marine acquisition
- recovery via *curvelet*-domain *sparsity* promotion or *low-rank* promotion

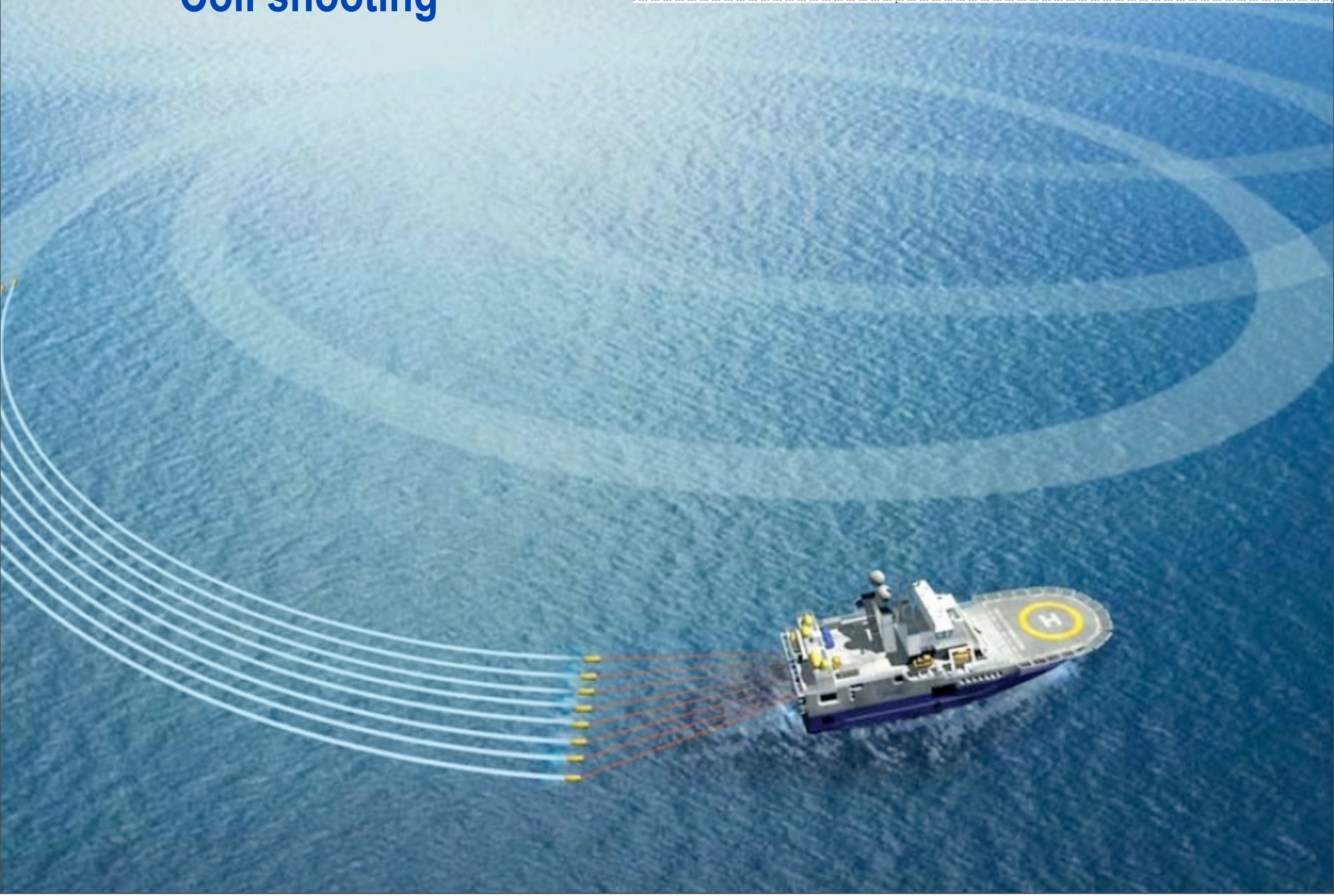


Coil shooting



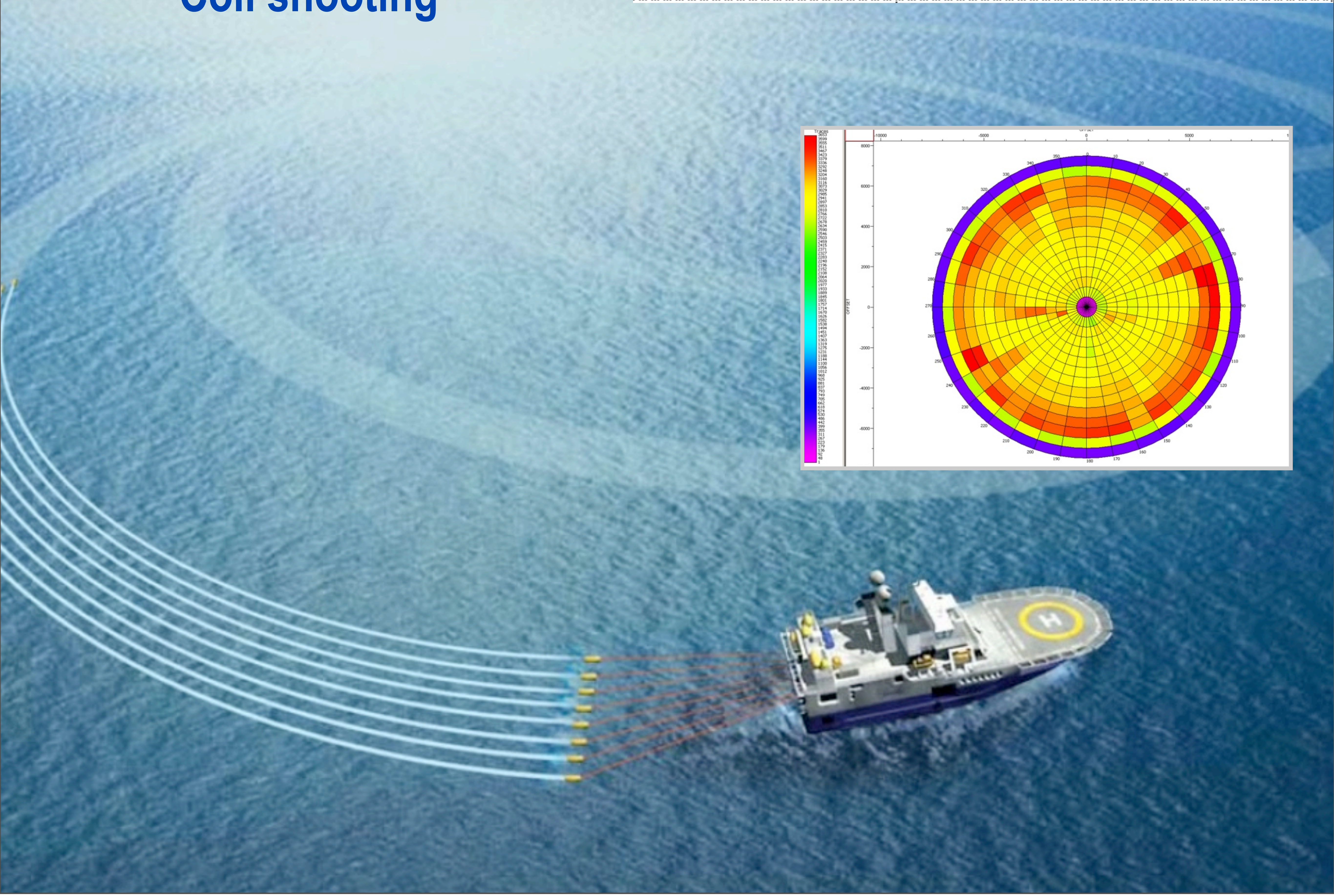
Coil shooting

$$W(t, x_s, y_s, x_r, y_r)$$



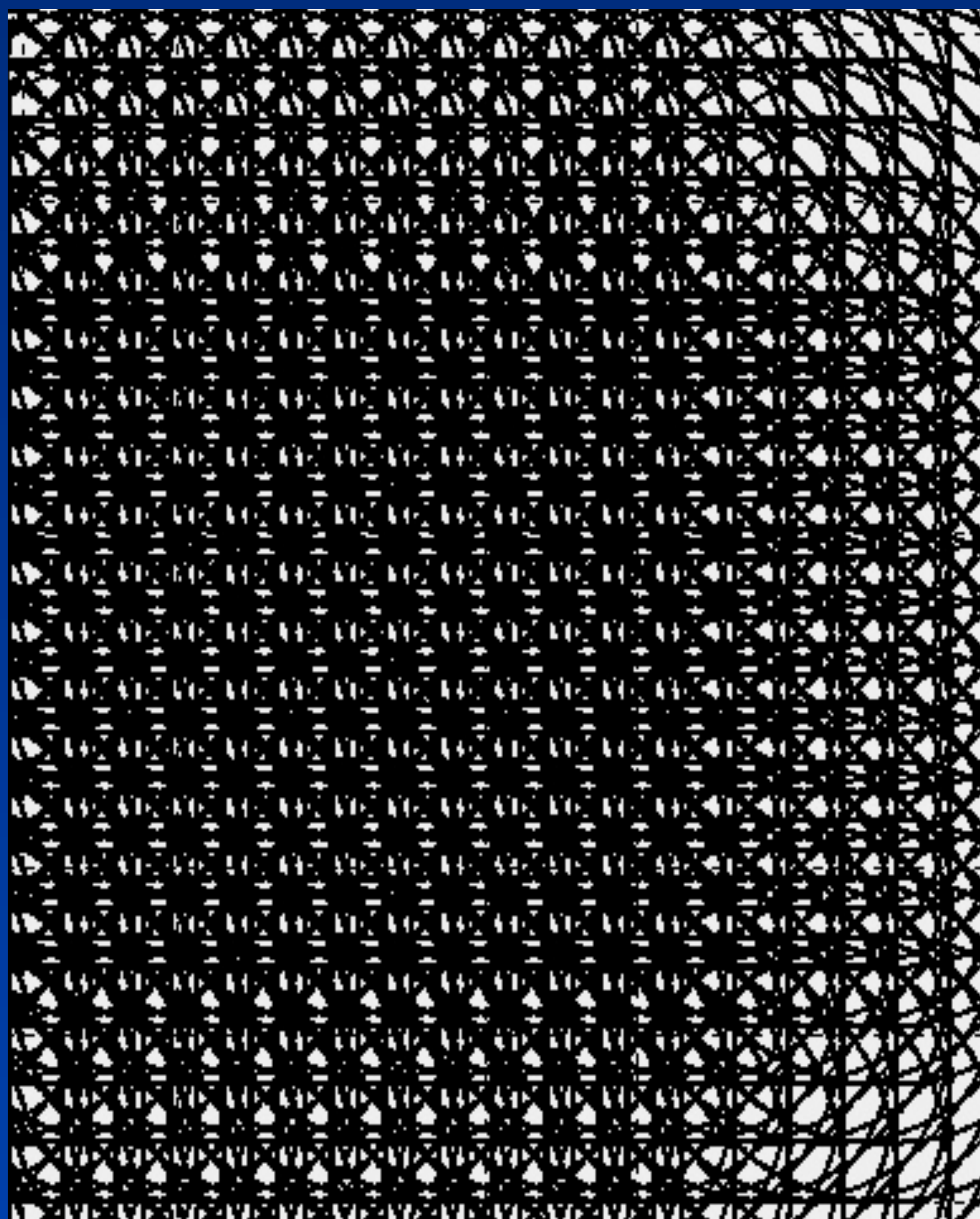
$$W(t, x_s, y_s, x_r, y_r)$$

Coil shooting

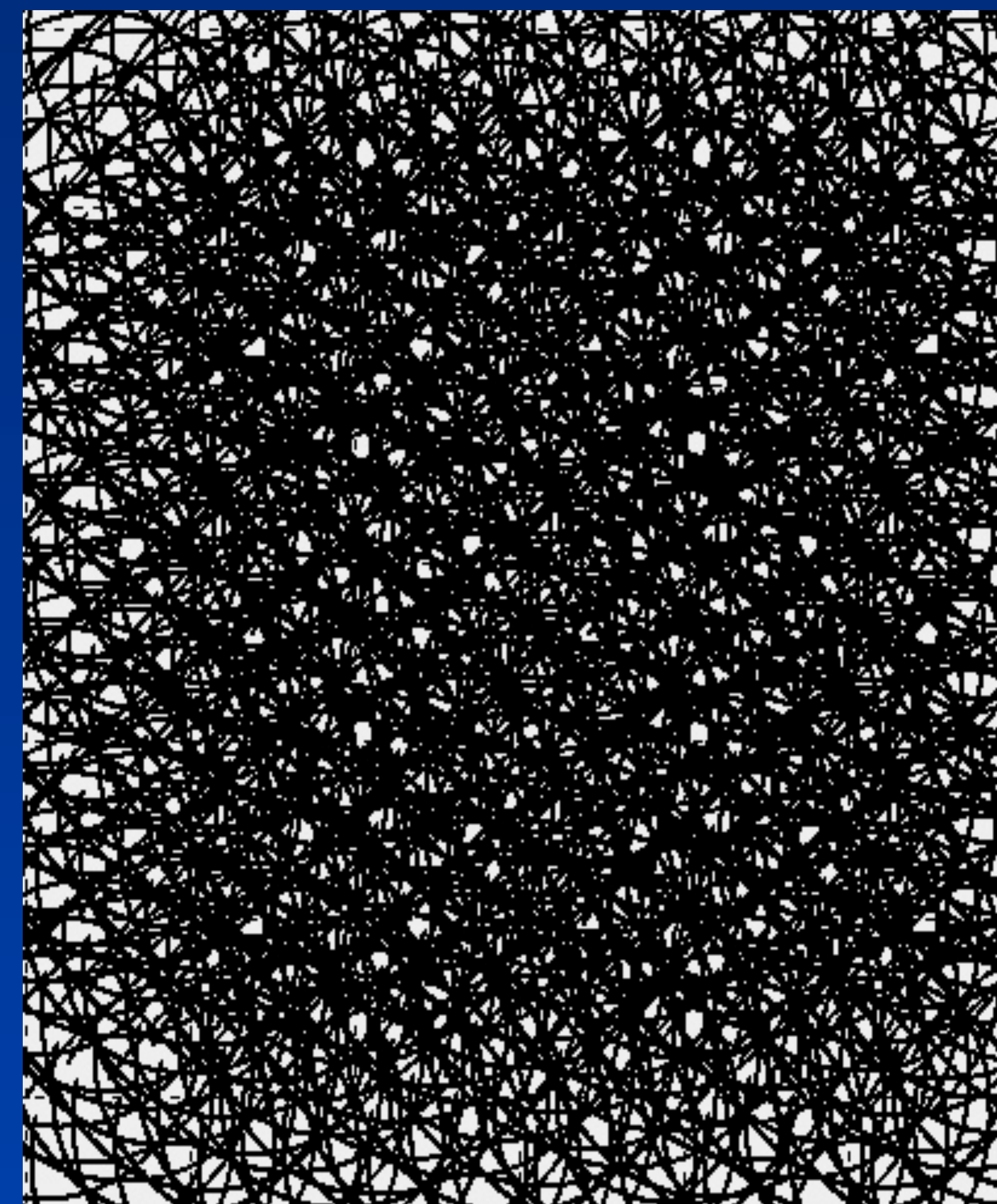


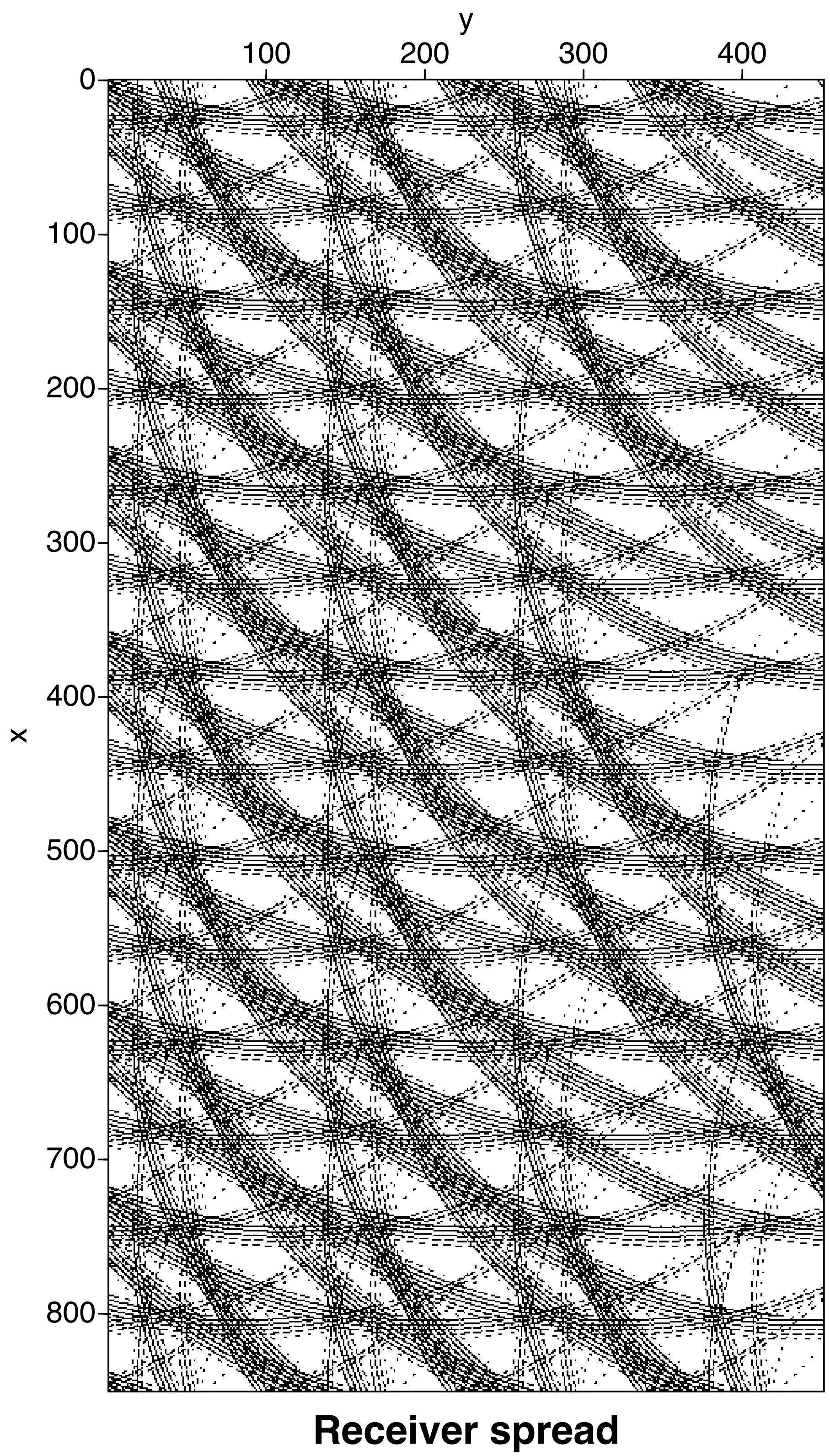
Shot distribution for single vessel coil shooting

Regular center distribution



Random center distribution



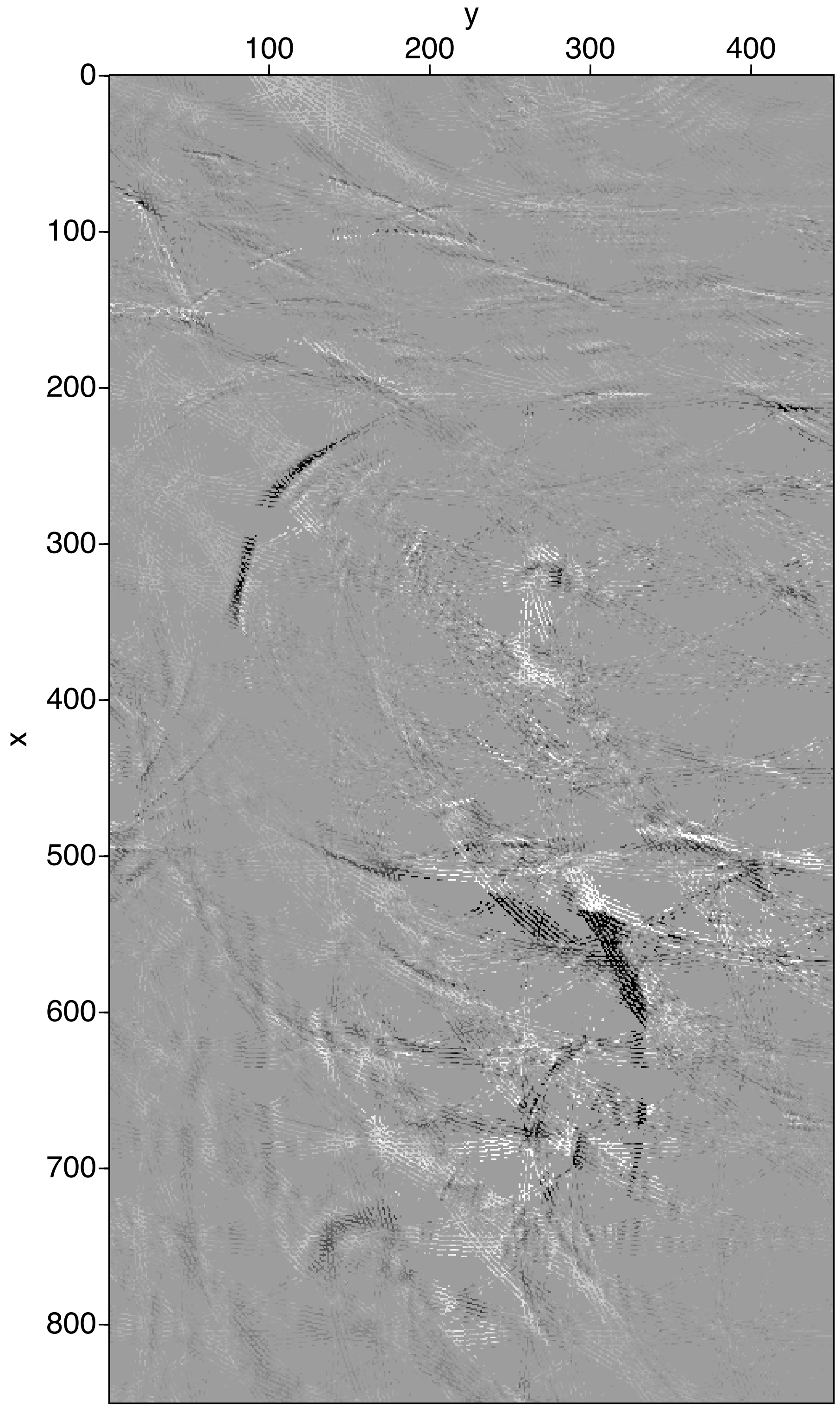


34 % of samples

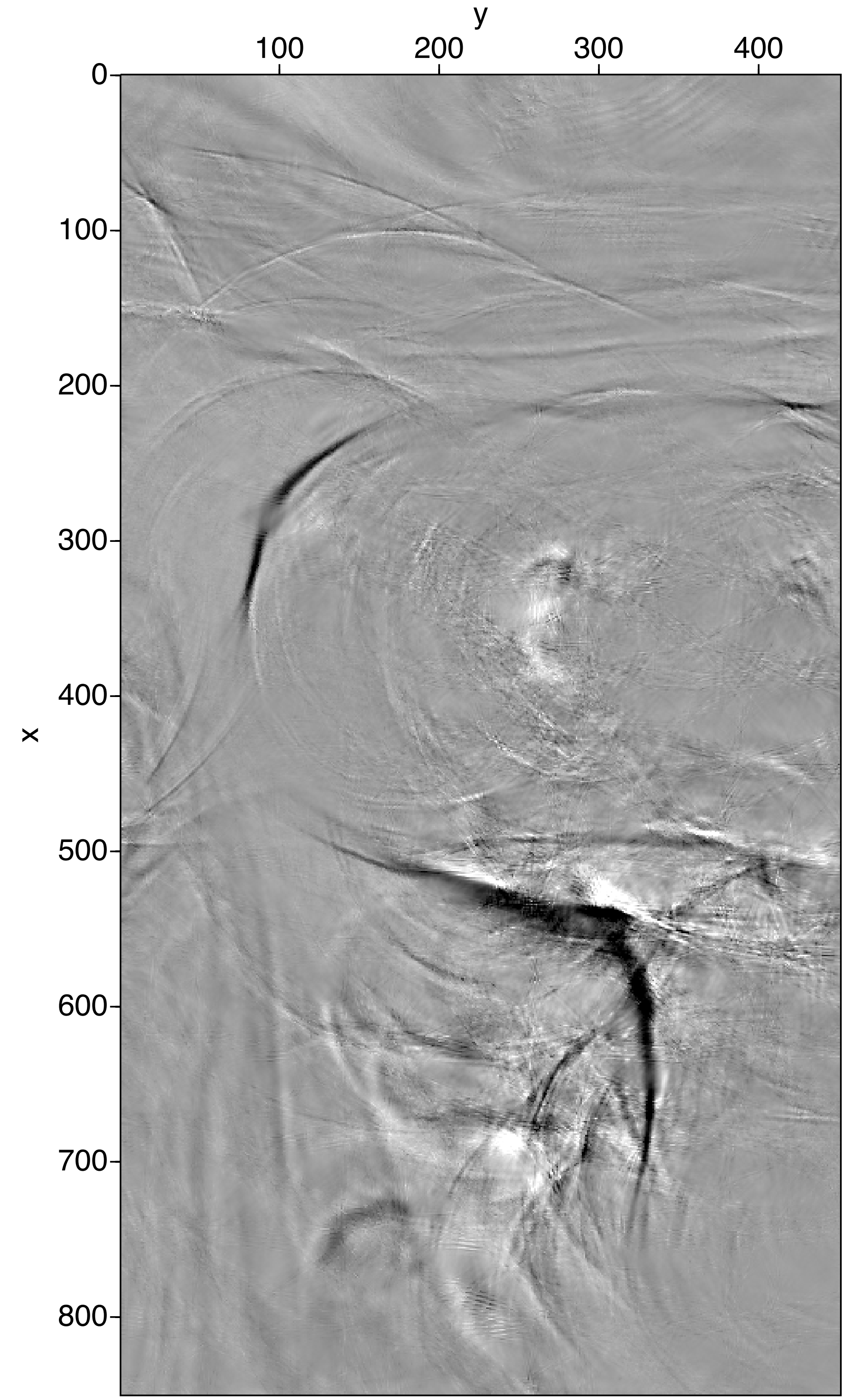
Courtesy Nick Moldoveanu

Challenge

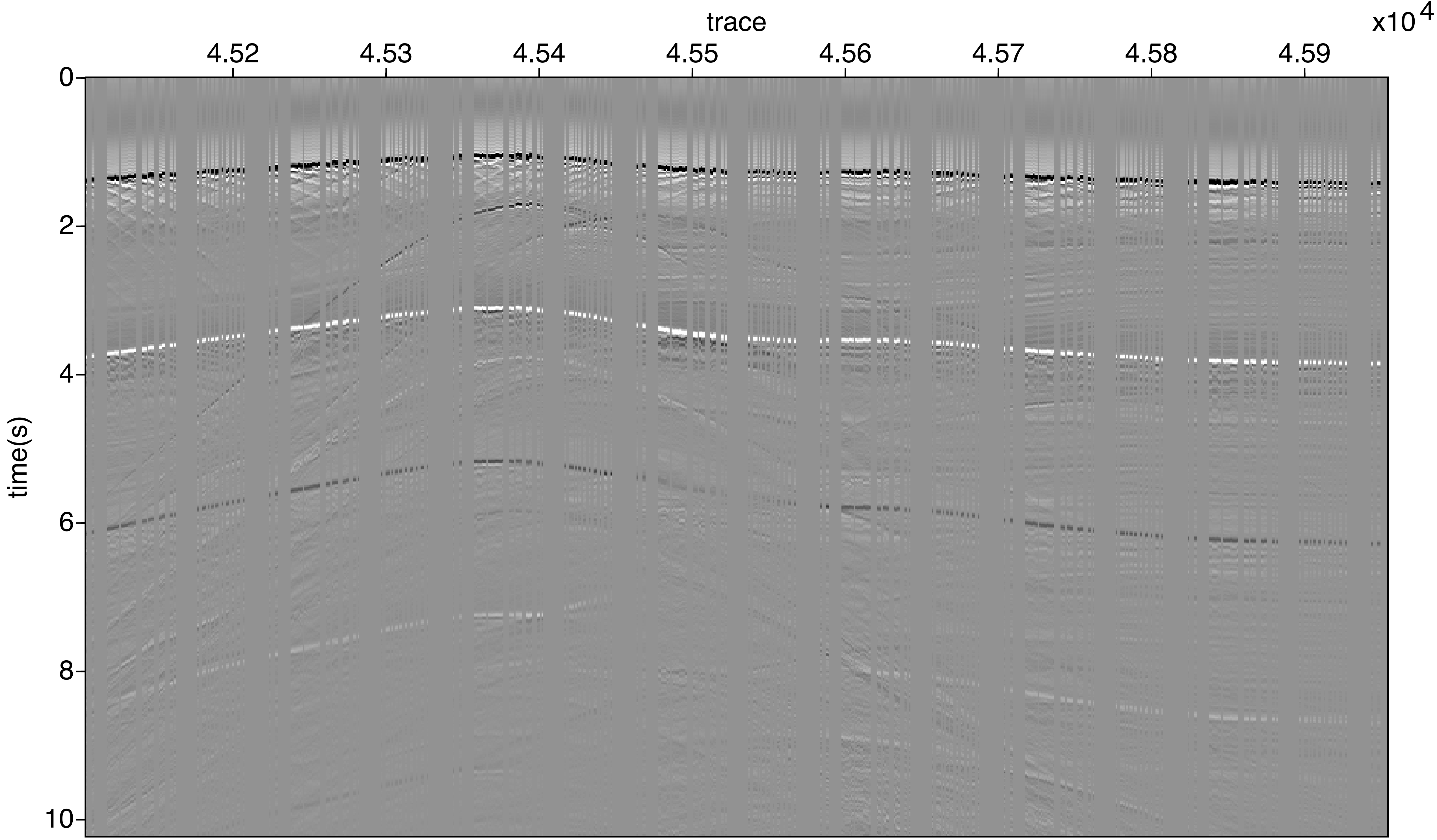
```
Starting SPGl1 recovery...

 ================================================================================
 SPGL1_SLIM v. 46  (Tue, 14 Jun 2011) based on v.1017
 ================================================================================
 No. rows              : 103672320       No. columns           : 1459253760
 Initial tau           : 0.00e+00       Two-norm of b         : 3.92e+05
 Optimality tol        : 1.00e-04       Target objective      : 0.00e+00
 Basis pursuit tol     : 1.00e-06       Maximum iterations    :      110

  Iter      Objective   Relative Gap  Rel Error      gNorm   stepG    nnzX    nnzG            tau
     0  3.9236638e+05  0.0000000e+00   1.00e+00  6.903e+03     0.0       0       0  2.2303101e+07
     1  3.9219958e+05  1.9364118e+00   1.00e+00  6.677e+03    -0.3       2       0
     2  3.4192692e+05  2.1884194e+00   1.00e+00  5.147e+03     0.0   14452       0
     3  3.2859582e+05  4.1722491e-01   1.00e+00  1.373e+03     0.0   48295       0

   108  1.5609476e+03  1.6347854e+04   1.00e+00  7.335e+00     0.0  356264726       0
   109  1.5850938e+03  9.3198454e+04   1.00e+00  4.283e+01     0.0  346355398       0
   110  1.5641524e+03  6.9308202e+04   1.00e+00  3.104e+01     0.0  345144021       0

 ERROR EXIT -- Too many iterations

 Products with A     :     125        Total time   (secs) :  34838.7
 Products with A'    :     112        Project time (secs) :  2875.2
 Newton iterations   :      26        Mat-vec time (secs) :  25882.1
 Line search its     :      23        Subspace iterations :       0
```



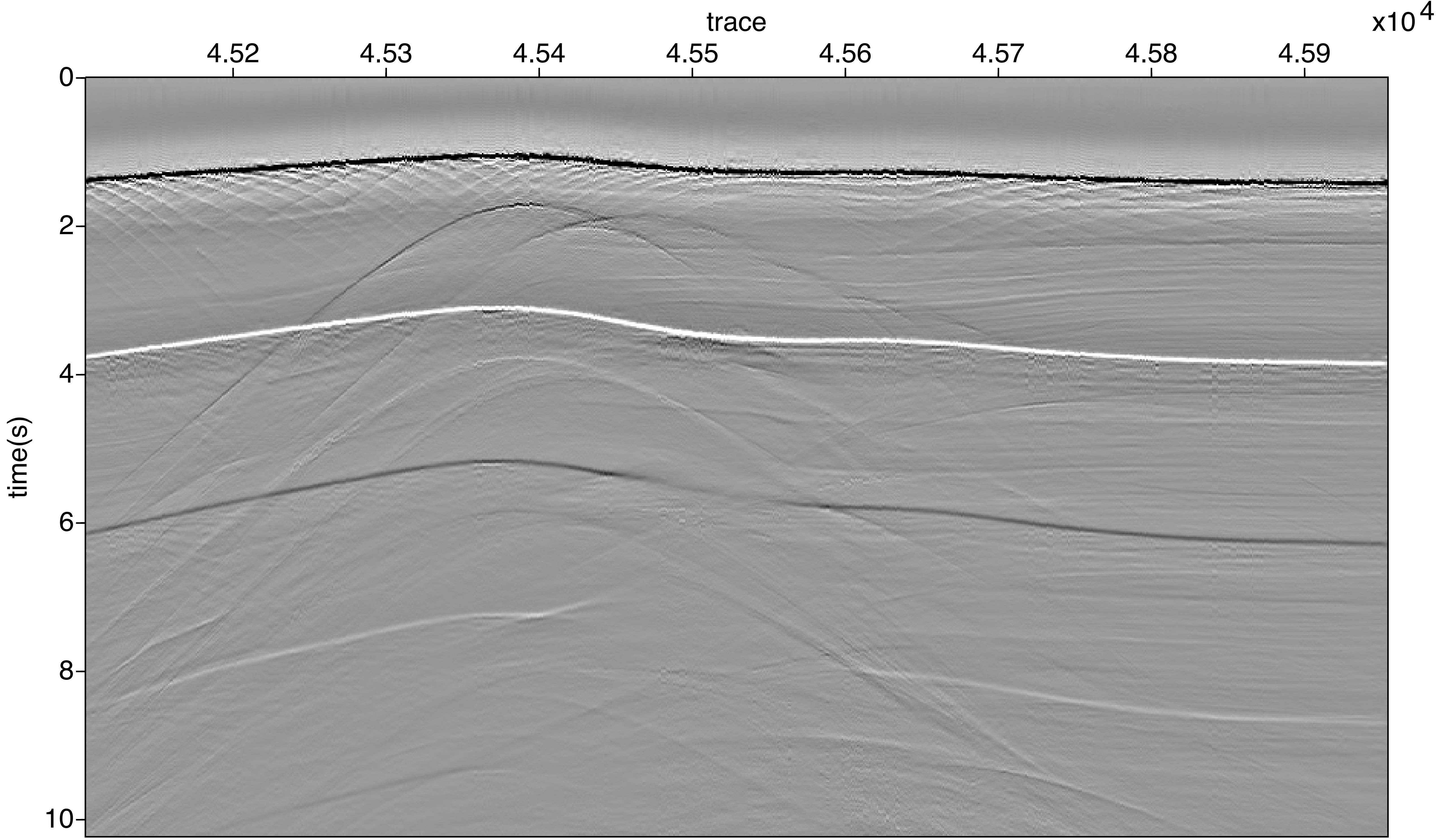
Input data



Interpolation with 2D Curvelet



Input data



Interpolation with 2D Curvelet

Open questions

Sparse recovery gives *encouraging* results

Able to *scale* sparse recovery to “large” problem sizes

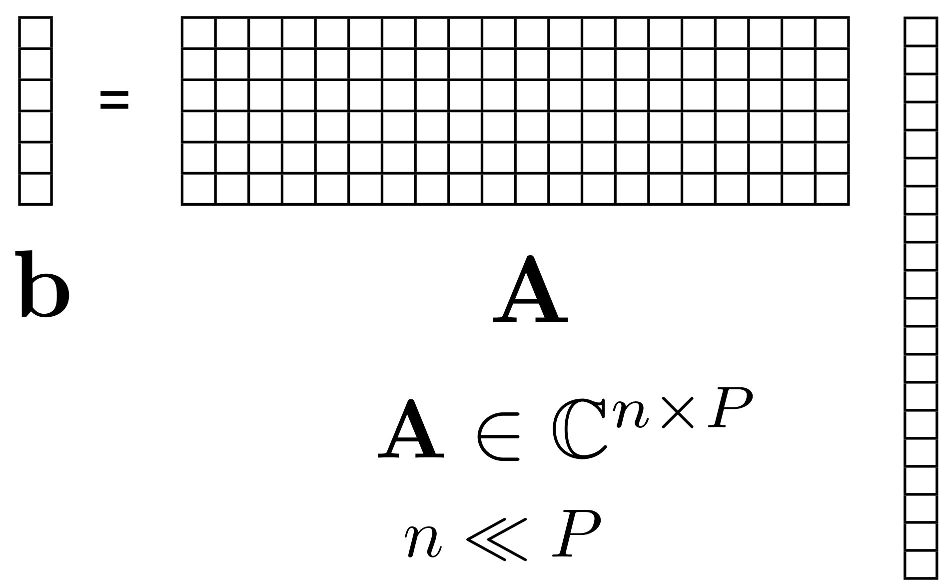
- true 3D remains a big challenge

Sparsity-promoting program *far* from reaching *convergence*

- what are *good* criteria to *measure* performance
- how can we *improve* convergence & *scale*

Problem statement

Solve an *underdetermined* system of *linear* equations:



data (measurements /observations)

$\mathbf{b} \in \mathbb{C}^n$

$\mathbf{A} \in \mathbb{C}^{n \times P}$

$n \ll P$

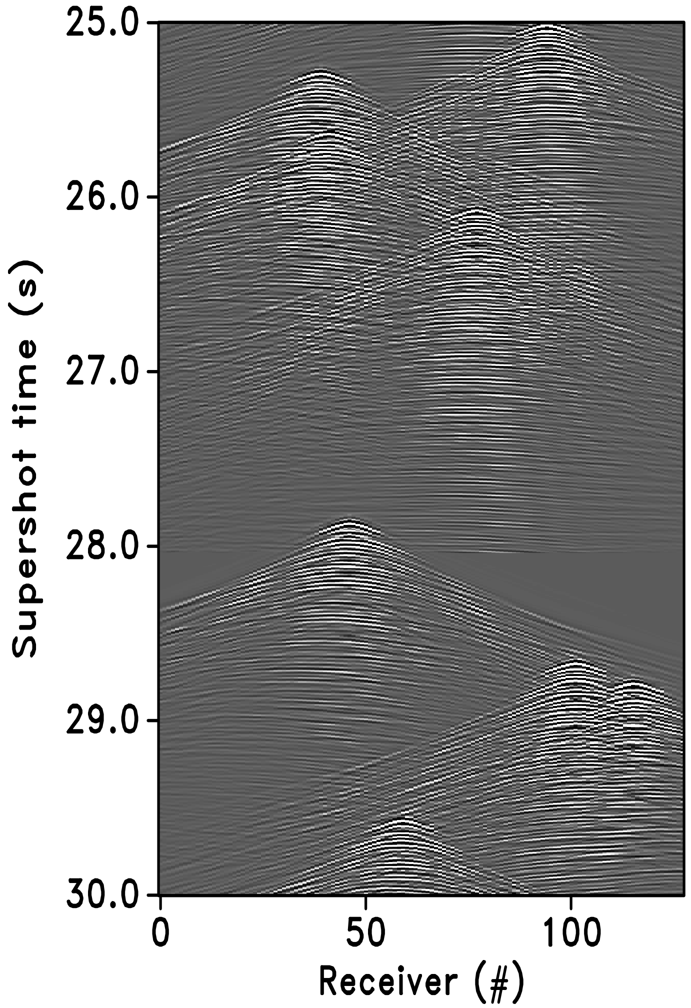
$\mathbf{x_0}$ unknown

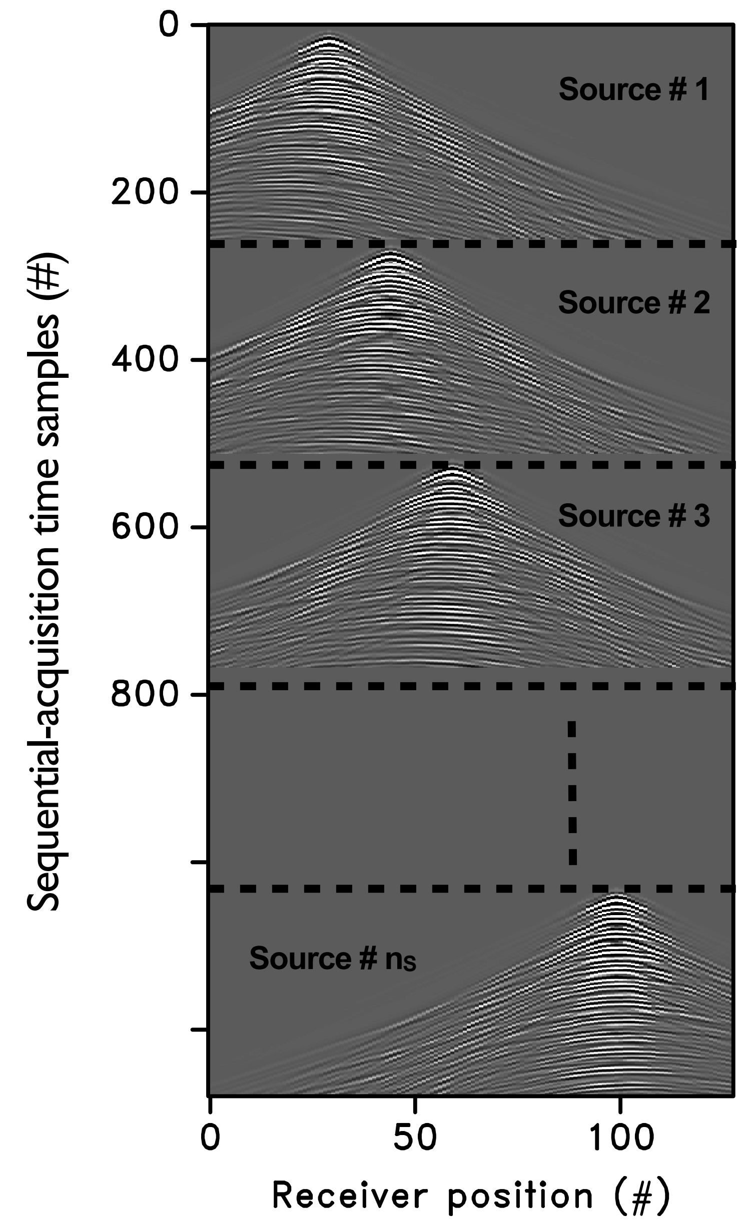
$\mathbf{x_0} \in \mathbb{C}^P$

$$\mathbf{A} = \underbrace{\mathbf{RM}}_{\text{sampling matrix}}\mathbf{S^H}$$

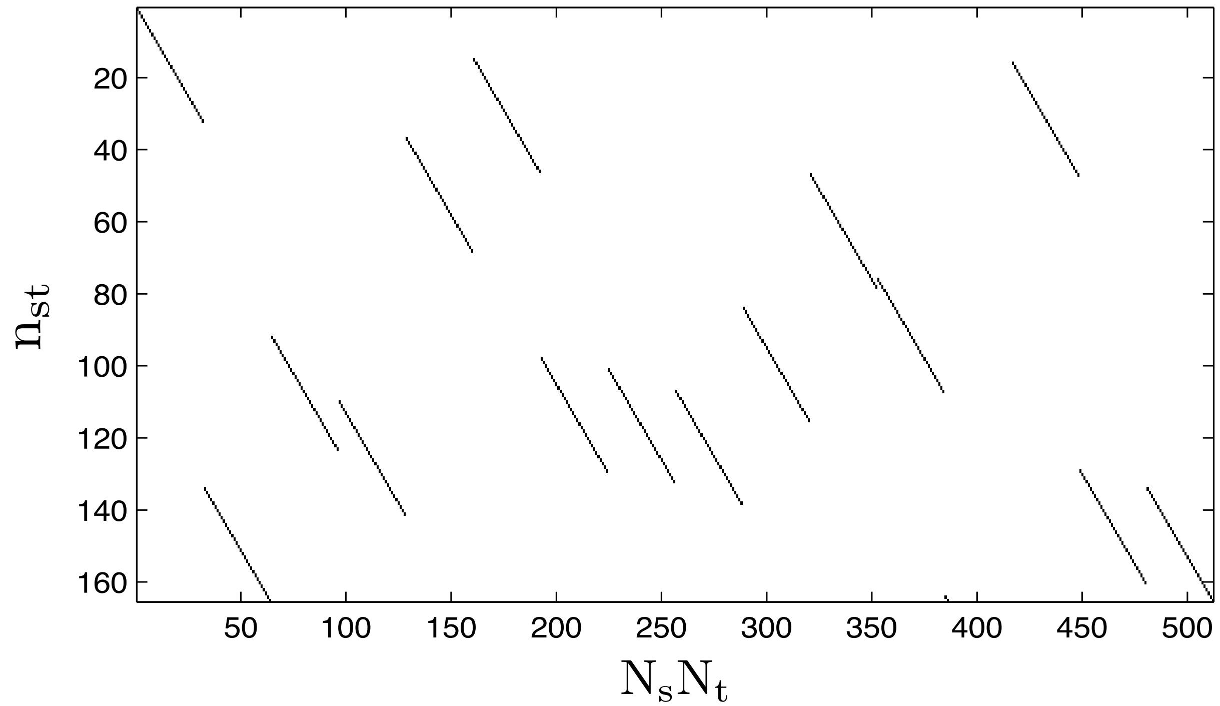
transform matrix

$$\mathbf{b} = \boxed{\mathbf{RM}} \; \mathbf{d}$$

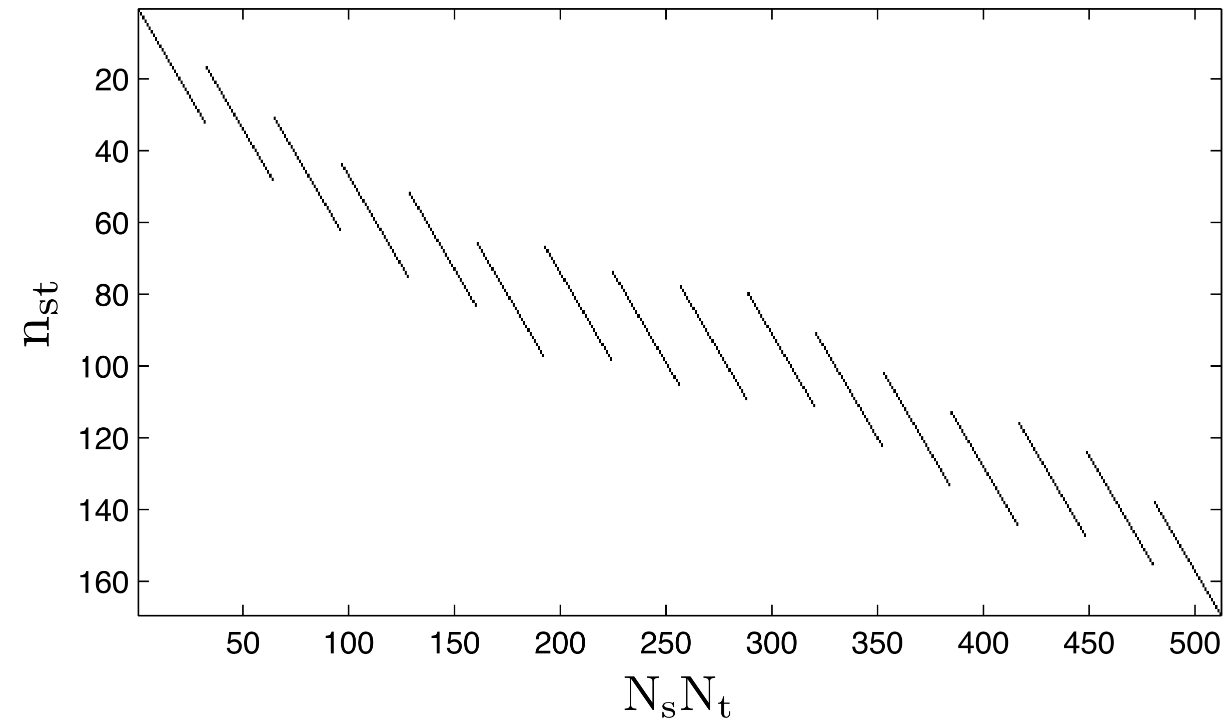




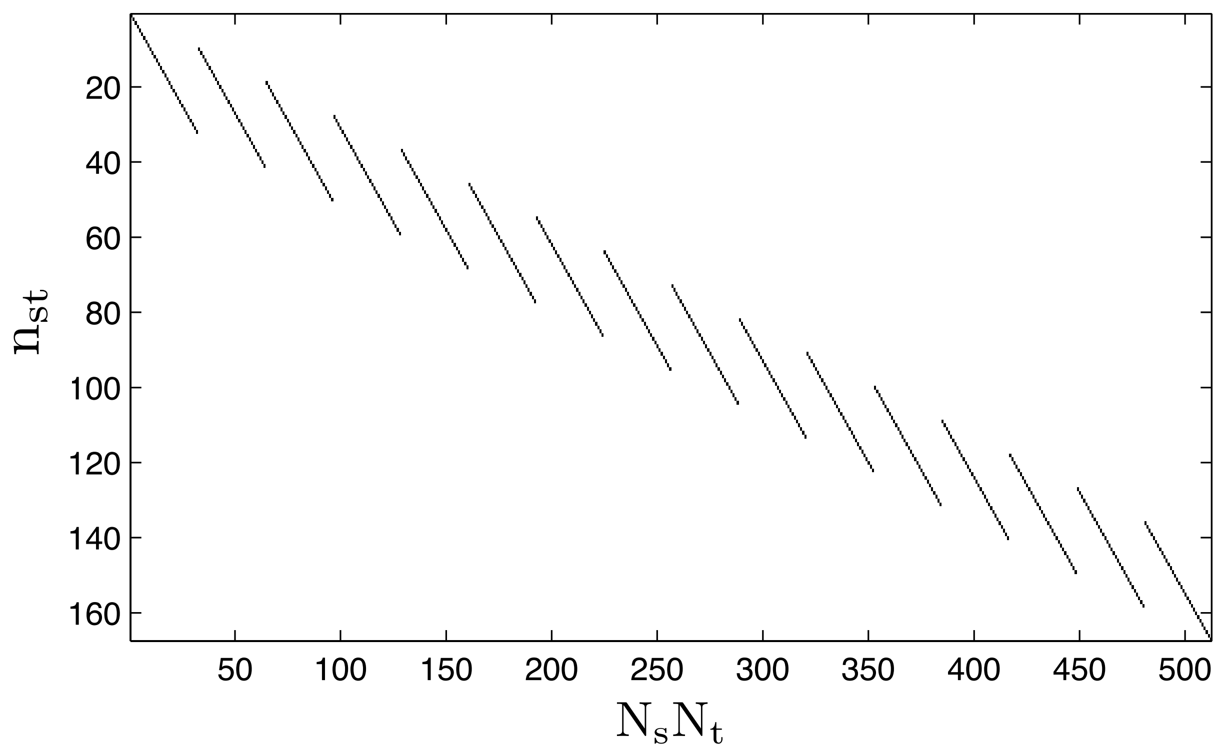
Sampling matrix (RM)



"Ideal" simultaneous acquisition



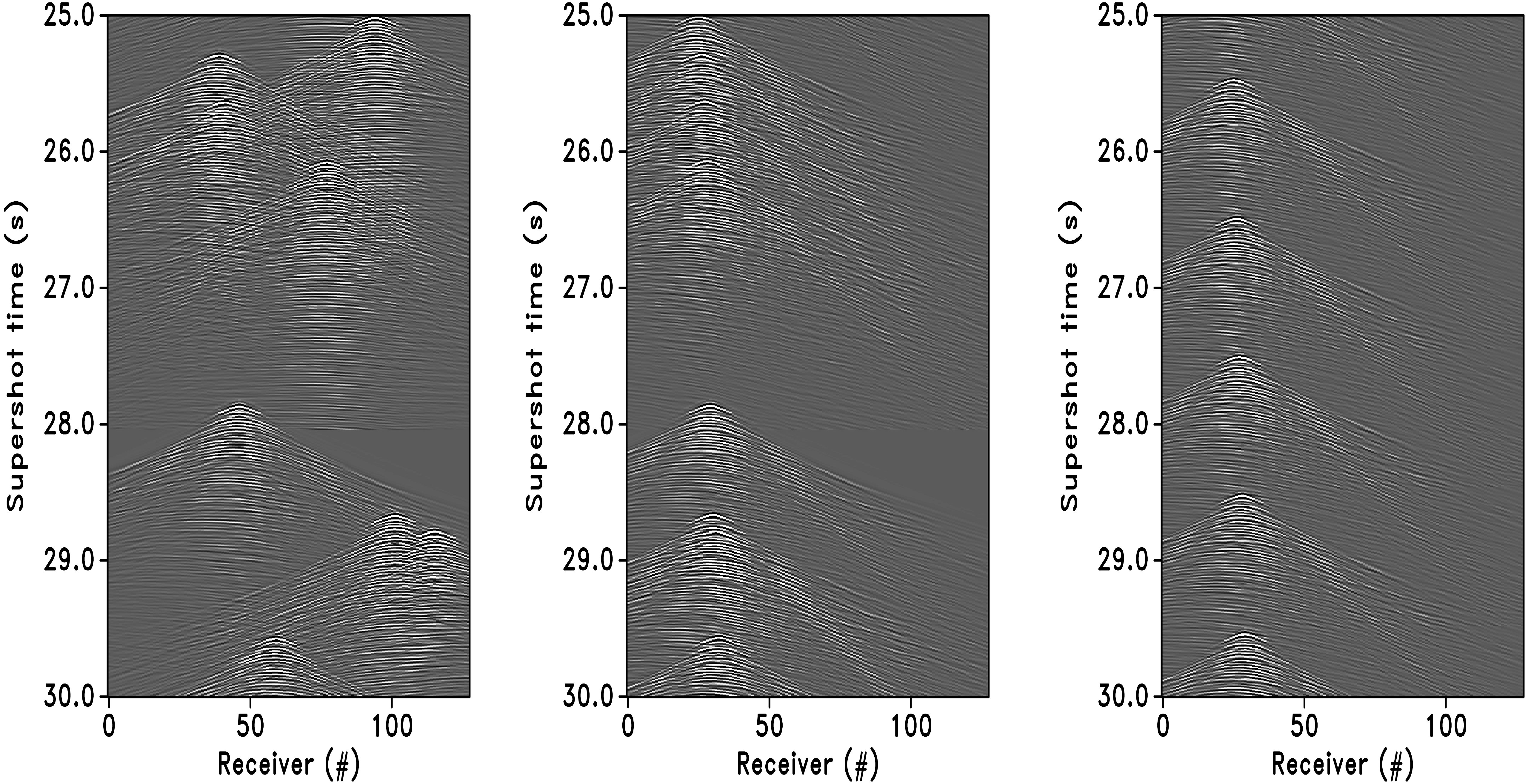
Random time-dithering



Periodic time-dithering

Measurements (b)

"Ideal" simultaneous acquisition | Random time-dithering | Periodic time-dithering



Sparse recovery

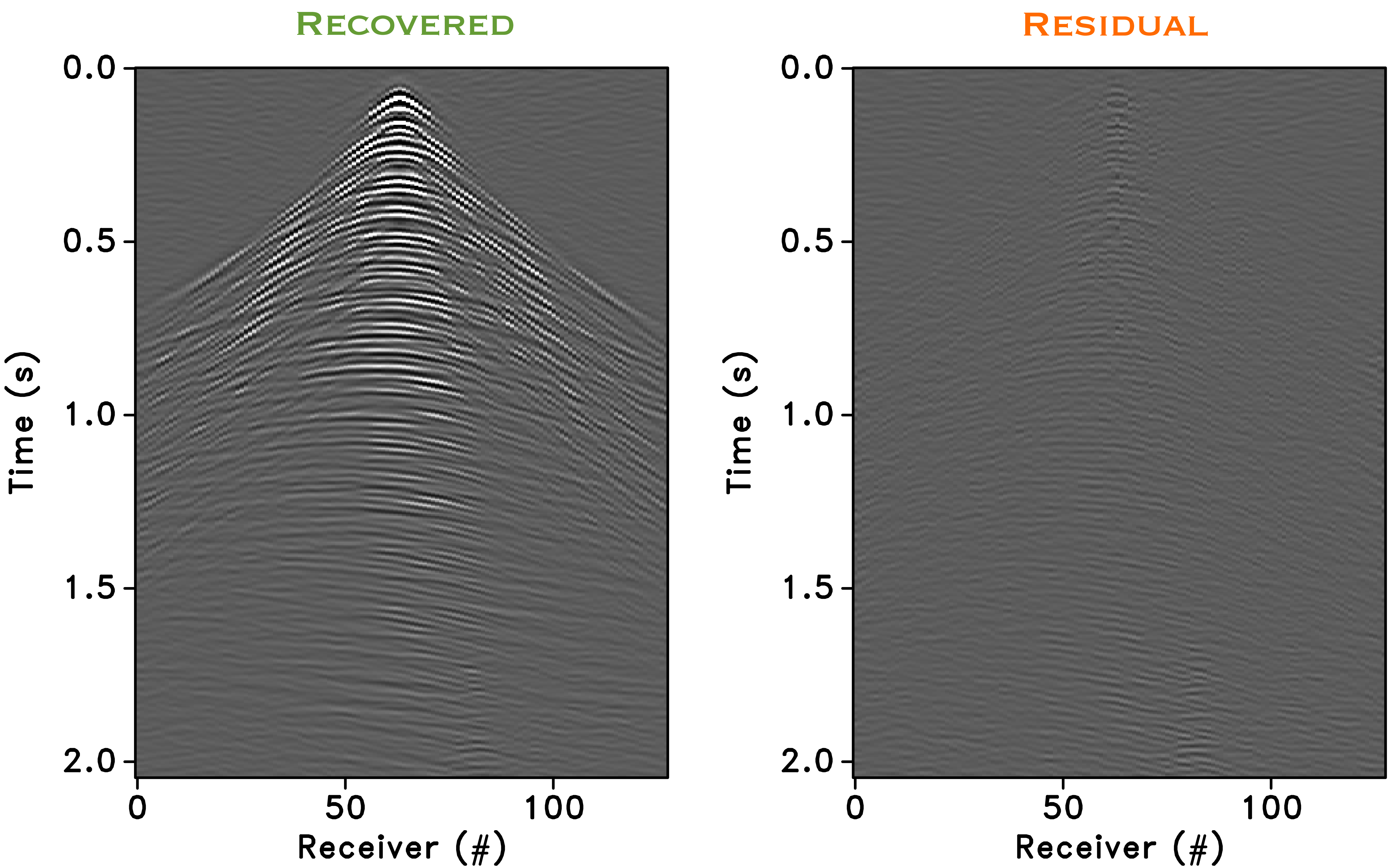
Solve the convex optimization problem (one-norm minimization):

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \underbrace{\mathbf{A}\mathbf{x} = \mathbf{b}}_{\substack{\text{data-consistent} \\ \text{amplitude recovery}}}$$

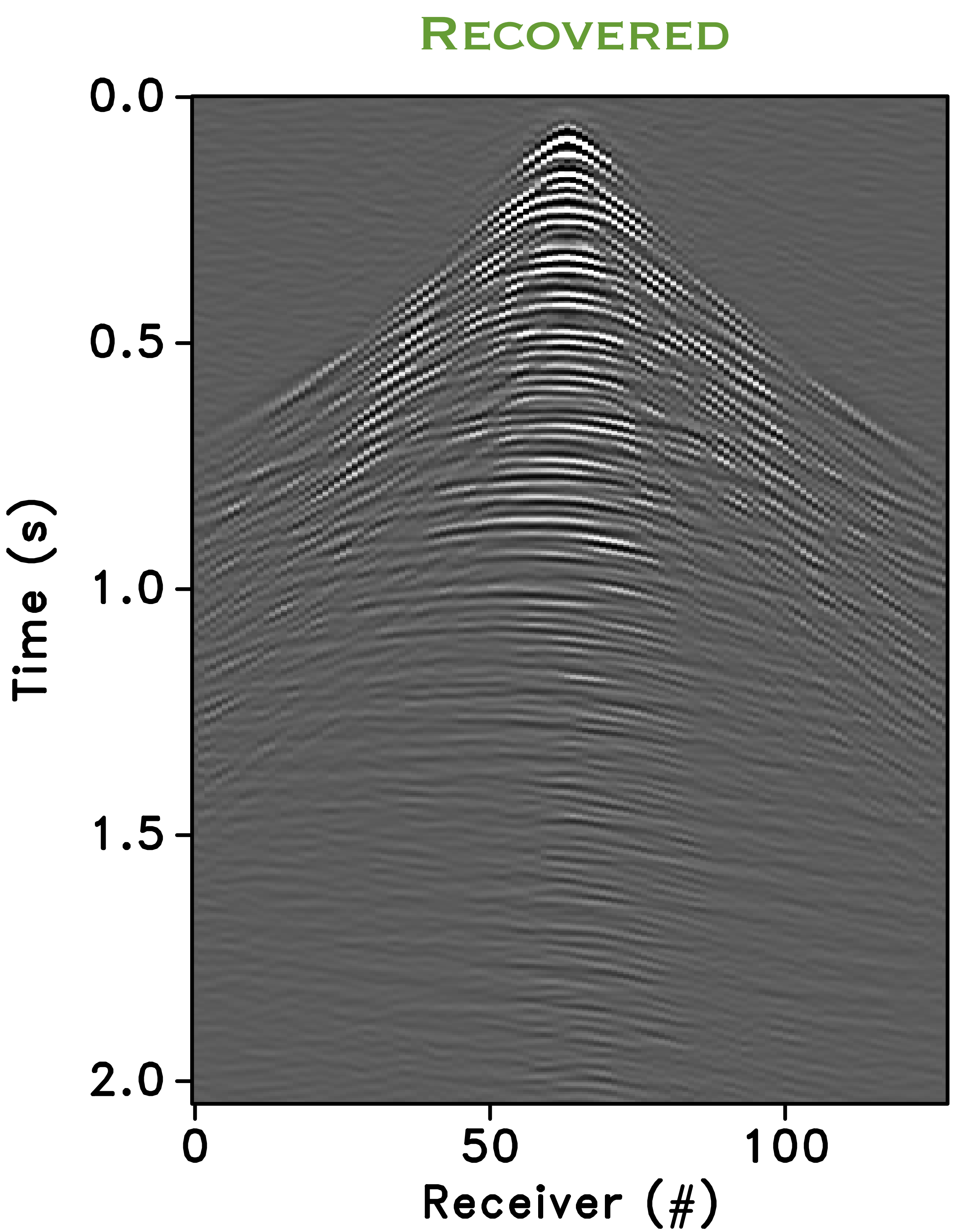
Sparsity-promoting solver: $\mathbf{SPG}\ell_1$ [van den Berg and Friedlander, 2008]

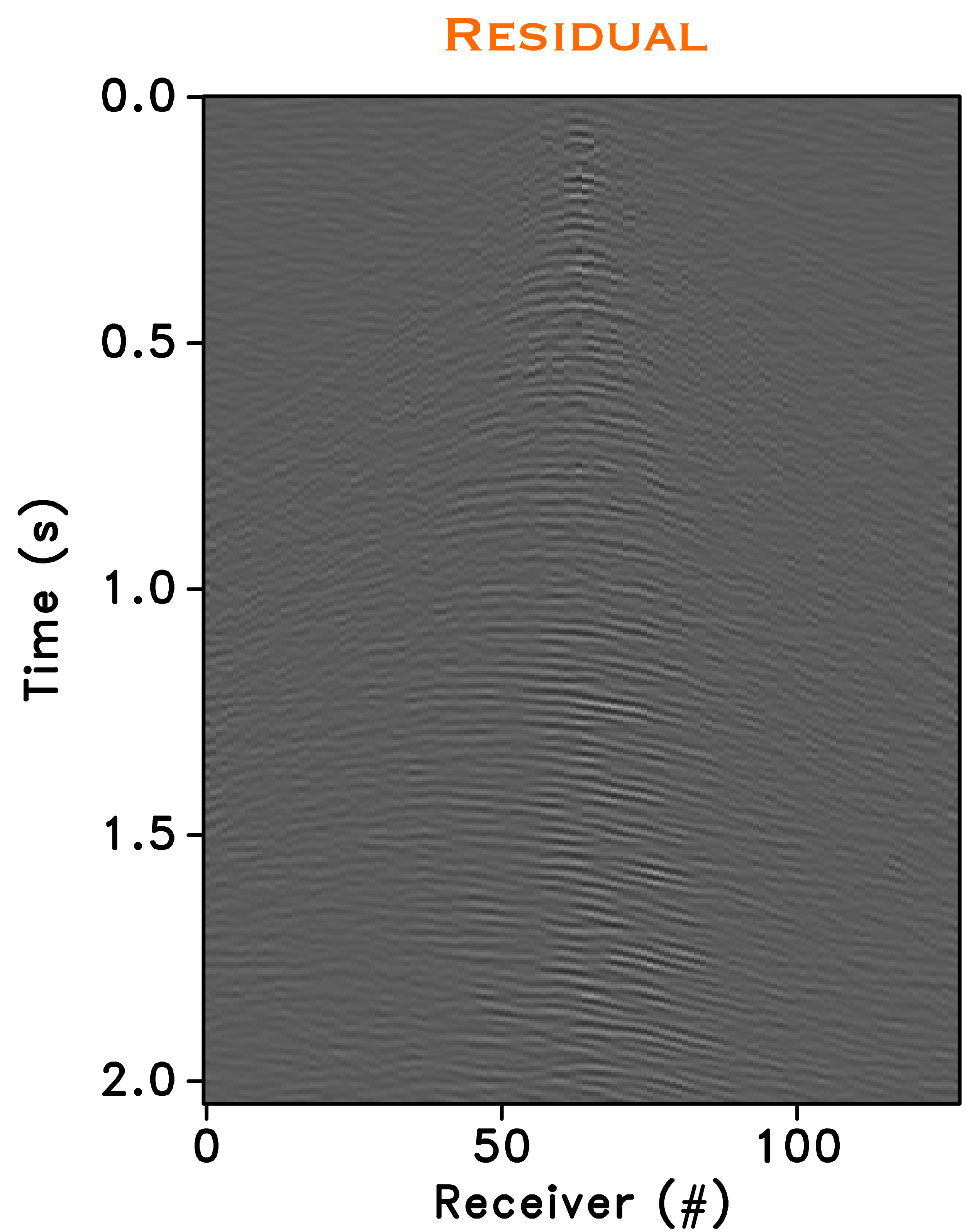
Recover single-source prestack data volume: $\tilde{\mathbf{d}} = \mathbf{S}^{\mathbf{H}}\tilde{\mathbf{x}}$

“Ideal” simultaneous acquisition
Sparsity-promoting recovery : 10.5 dB

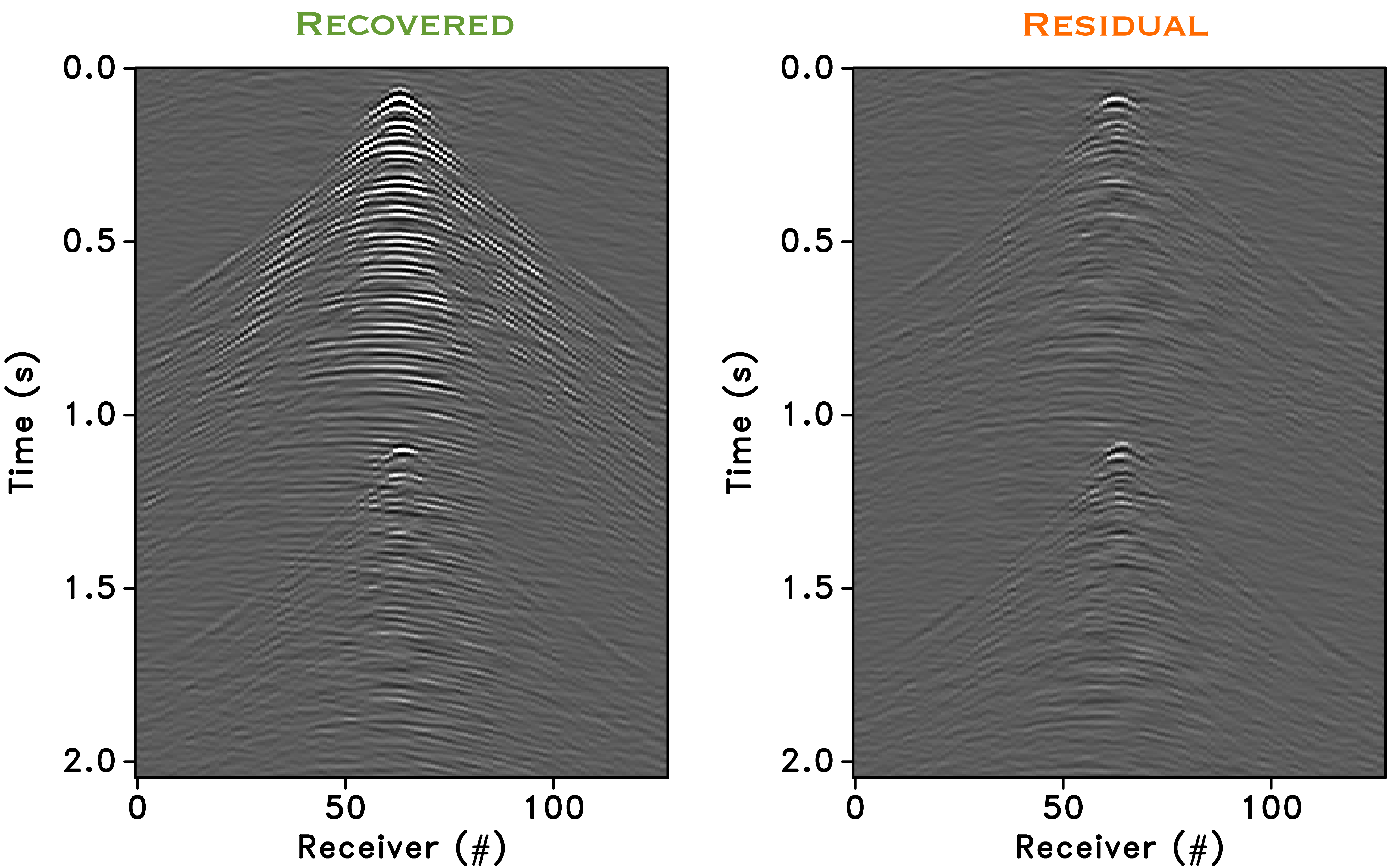


Random time-dithering
Sparsity-promoting recovery : 8.06 dB





Periodic time-dithering
Sparsity-promoting recovery : 4.80 dB



Gram matrices

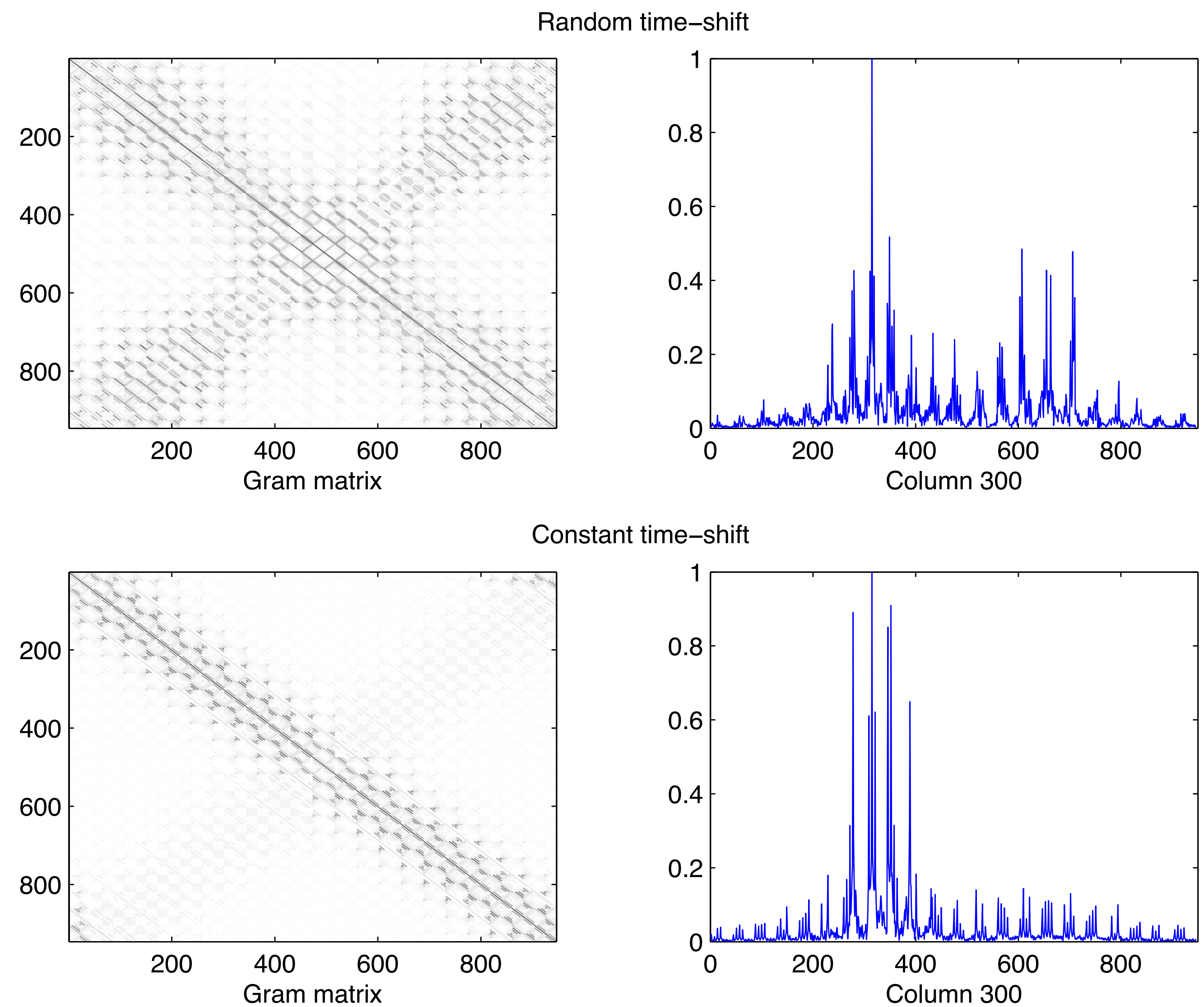


Figure 3: Gram matrices of randomized and constant time shifting operators, top and bottom left, respectively, coupled with a curvelet transform. The top and bottom right plots show column 300 of the Gram matrices.

Different transforms

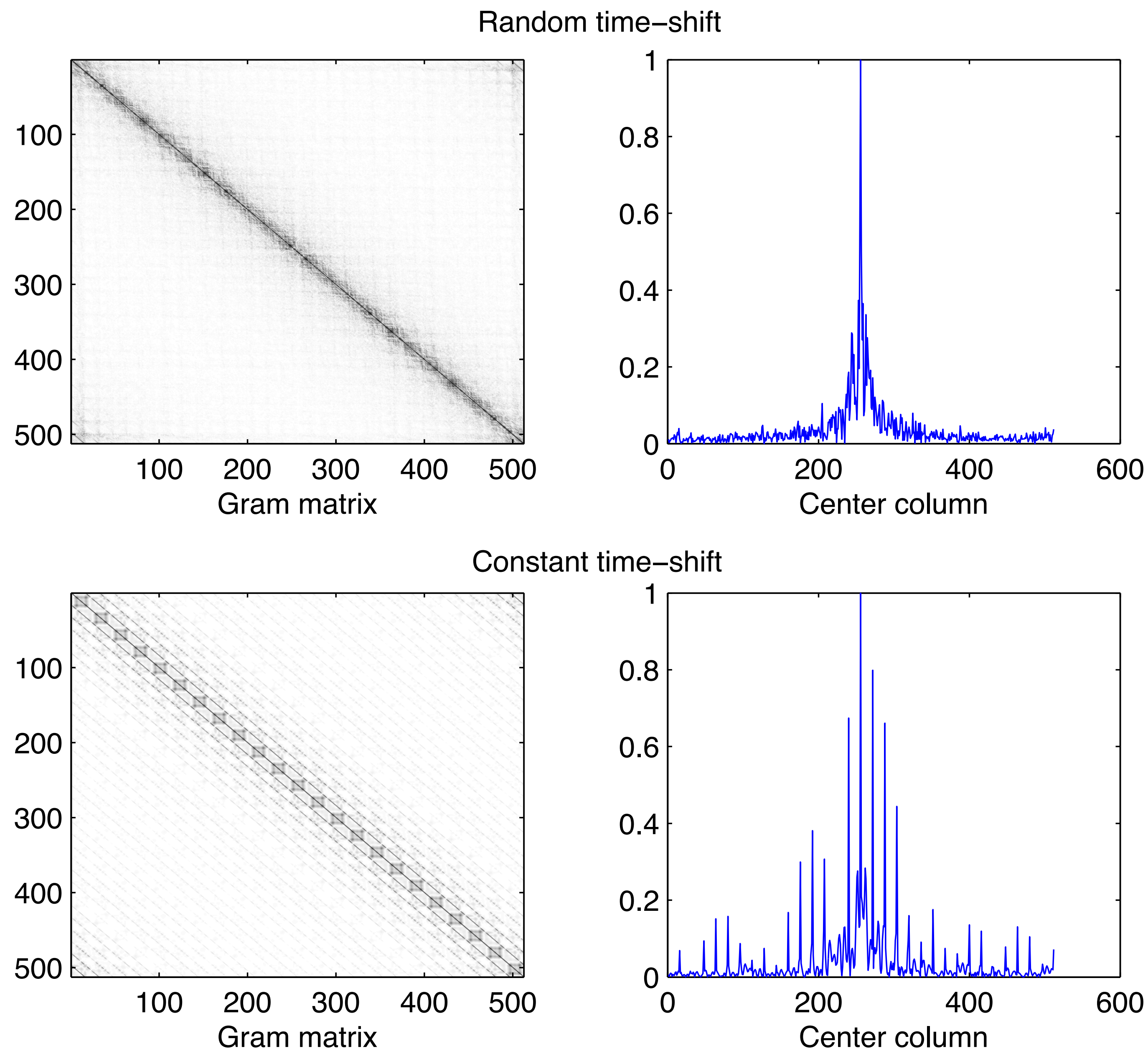
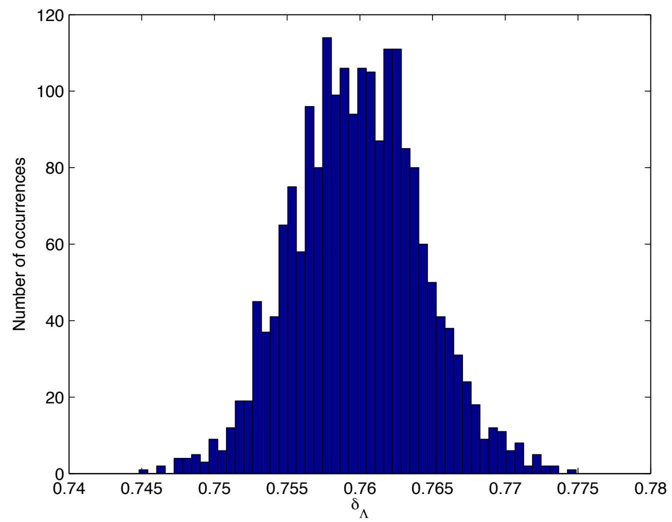
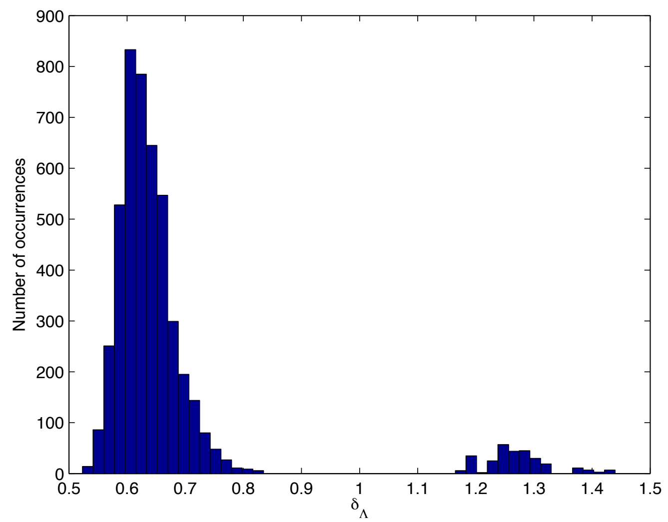


Figure 4: Gram matrices of randomized and constant time shifting operators, top and bottom left, respectively, coupled with a Fourier transform. The top and bottom right plots show the center columns of the Gram matrices.

RIP constants



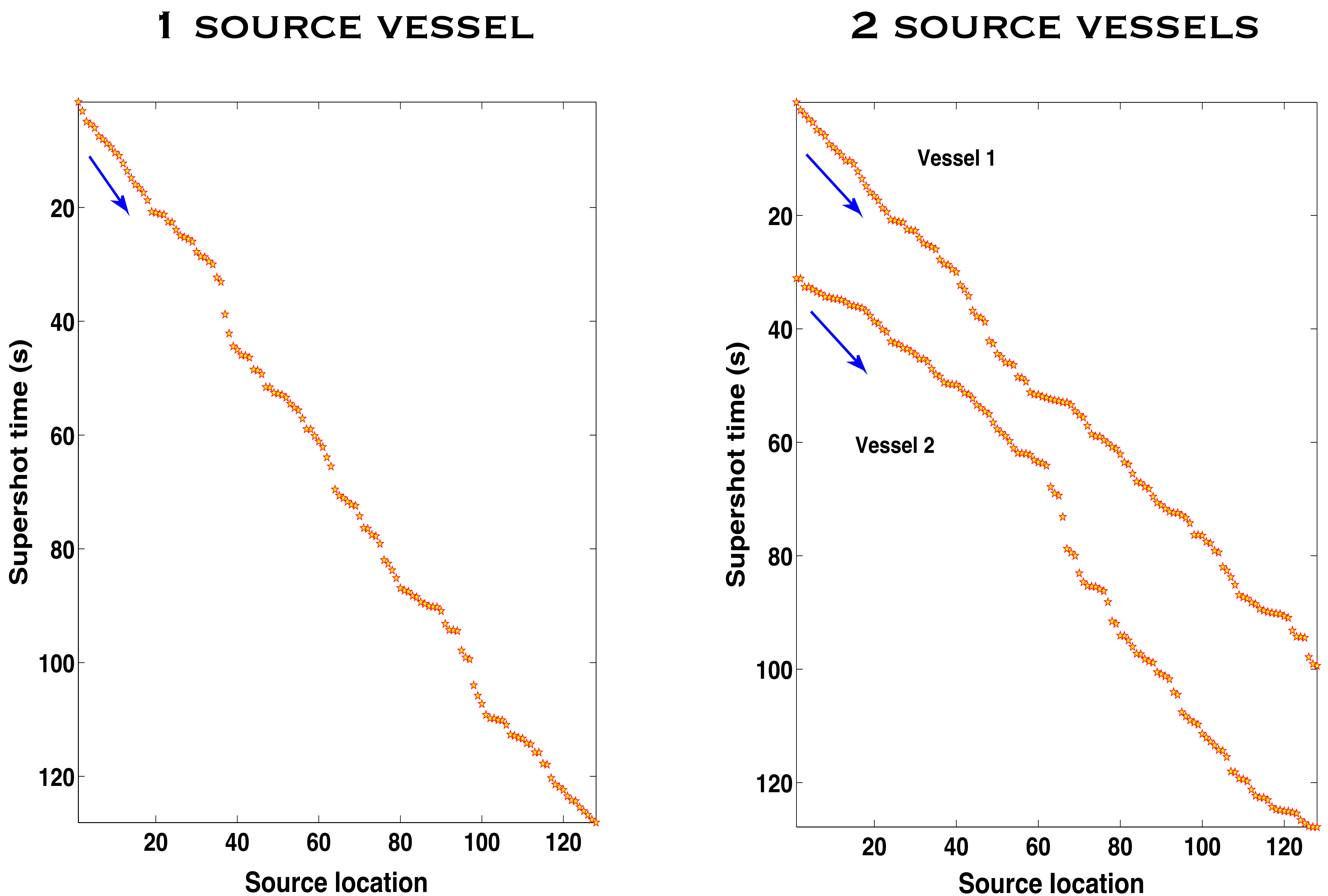
(a)



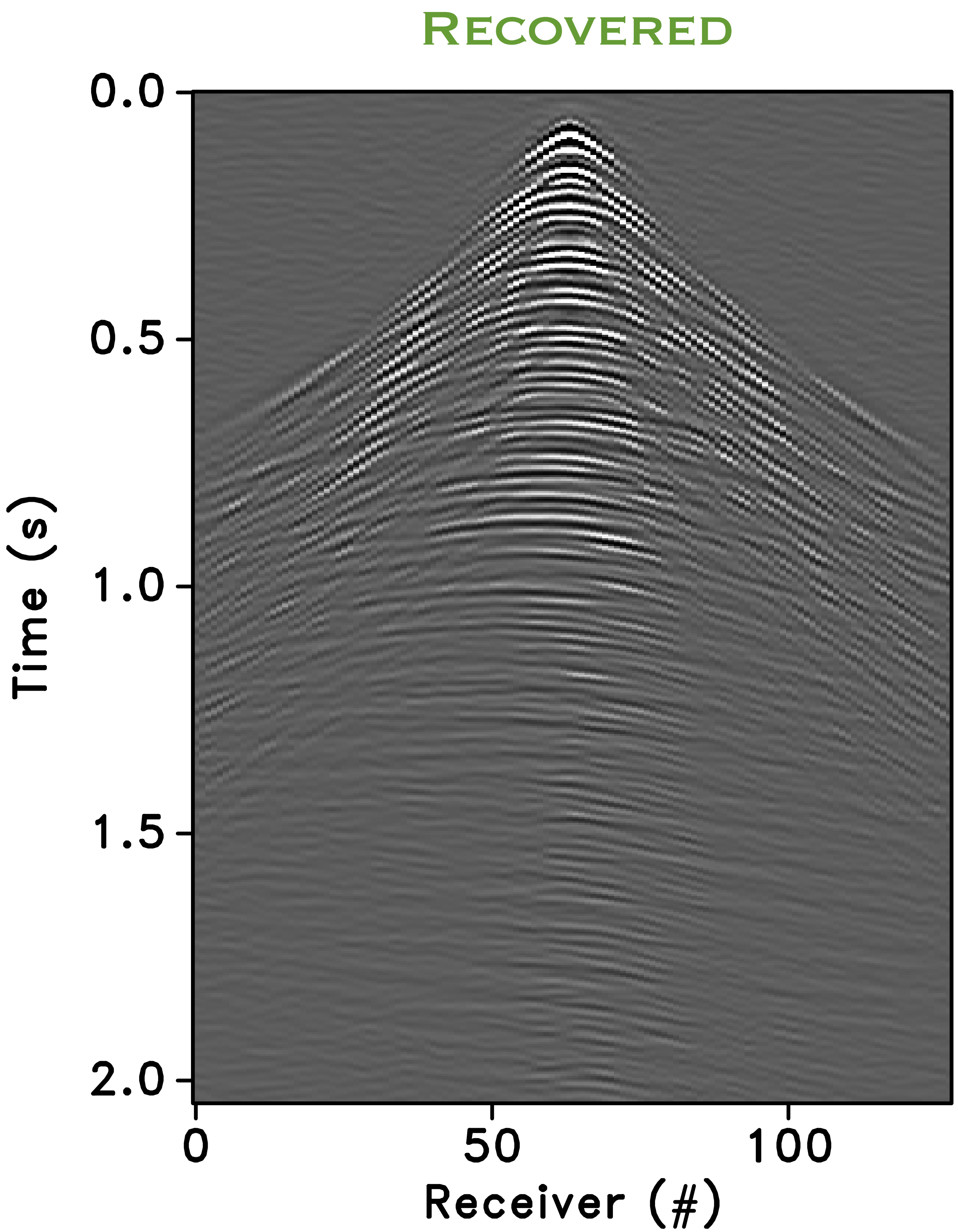
(b)

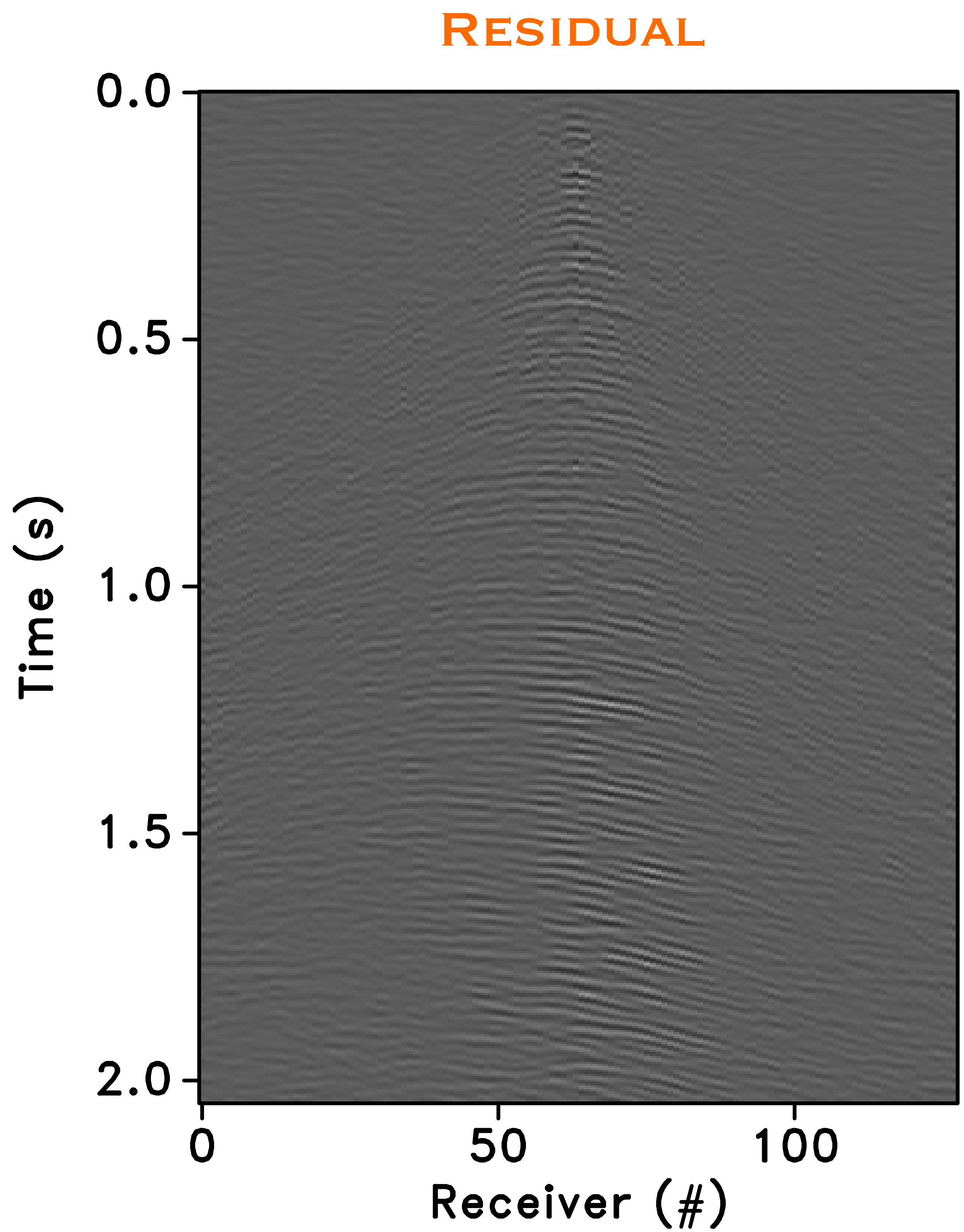
Figure 5: Comparison between the histograms of $\hat{\delta}_\Lambda$ from 1000 realizations of $\mathbf{A}_\Lambda$, the random time-shift sampling matrices $\mathbf{A} = \mathbf{RMS}^H$ restricted to a set Λ of size k, the size support of the largest transform coefficients of a real (Gulf of Suez) seismic image. The transform $\mathbf{S}$ is (a) the curvelet transform and (b) the nonlocalized 2D Fourier transform. The histograms show that randomized time-shifting coupled with the curvelet transform has better behaved RIP constant ($\hat{\delta}_\Lambda = \max\{1-\sigma_{\min}, \sigma_{\max}-1\} < 1$) and therefore promotes better recovery.

Random time-dithering

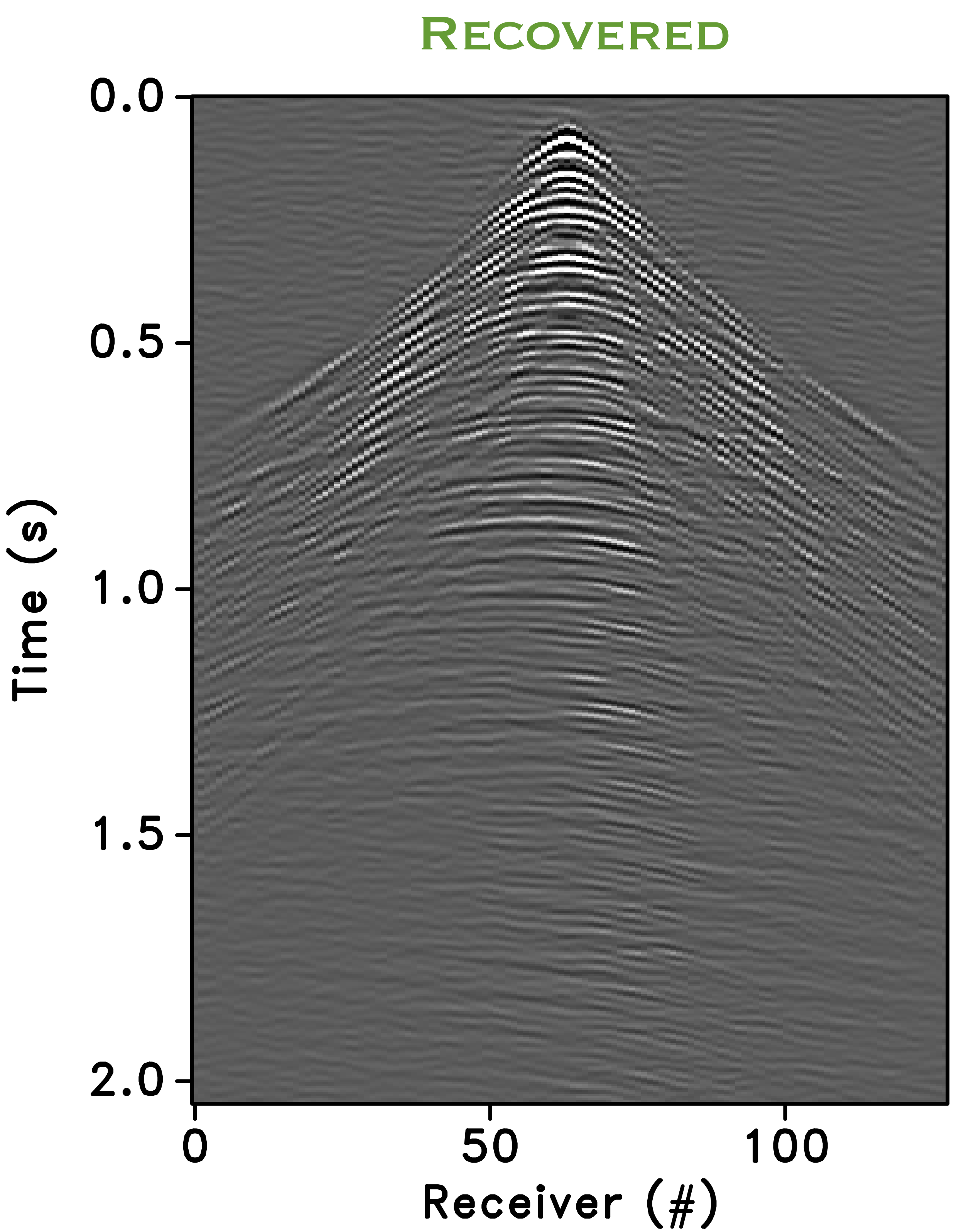


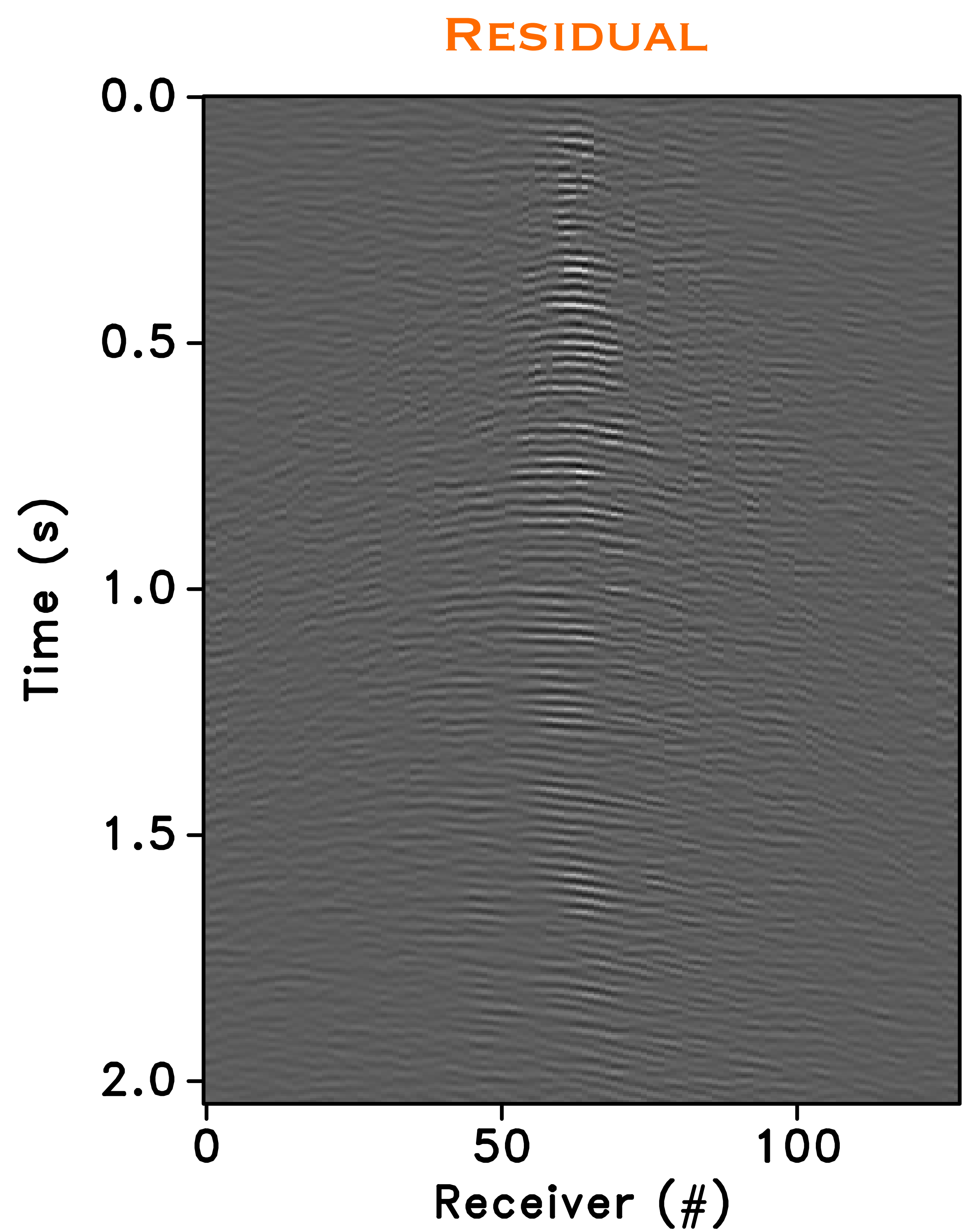
Random time-dithering with 1 source vessel
Recovery : 8.06 dB





Random time-dithering with 2 source vessels
Recovery : 10.3 dB





Challenges

Extension to 3D seismic (5-D data) exposes vulnerabilities

- redundancy of directional spasifying transforms
- cost of matvecs and # of matvecs for convex optimization

Explore a different kind of structure

- “low-rank” of matrix / tensor representations
- seismic data may not be low-rank but we have seen encouraging results

Nuclear Norm

- Given any matrix $X = USV^T$,

  the nuclear norm is $\|X\|_* = \sum(\mathrm{diag}(S))$.

- Just like the 1-norm approximates the 0-norm, so the *nuclear* norm approximates the rank.

- Therefore, to find a low rank solution, solve:

$$\min_X \|X\|_*$$

$$\text{such that } \|b - \mathcal{F}(X)\|_2 \leq \sigma \, .$$

Bring on the Pareto!

$$\min_X \|X\|_* \quad \text{such that } \|b - \mathcal{F}(X)\|_2 \leq \sigma \ .$$

- We can use SPGL1 to solve such problems if
  - It is easy to project onto $\mathbb{B}_*^\tau := \{X : \|X\|_* \leq \tau\}$
  - It is easy to evaluate the *dual* norm.
- *Dual* norm is simply *maximum* singular value (op norm)
- But just computing the nuclear norm requires SVDs. Fortunately, we can use a clever trick...

Factorization Approach

- The *Nuclear* norm has a convenient property:

$$\|X\|_* = \inf_{X=LR^*} \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right)$$

- We can work with L, R rather than X:

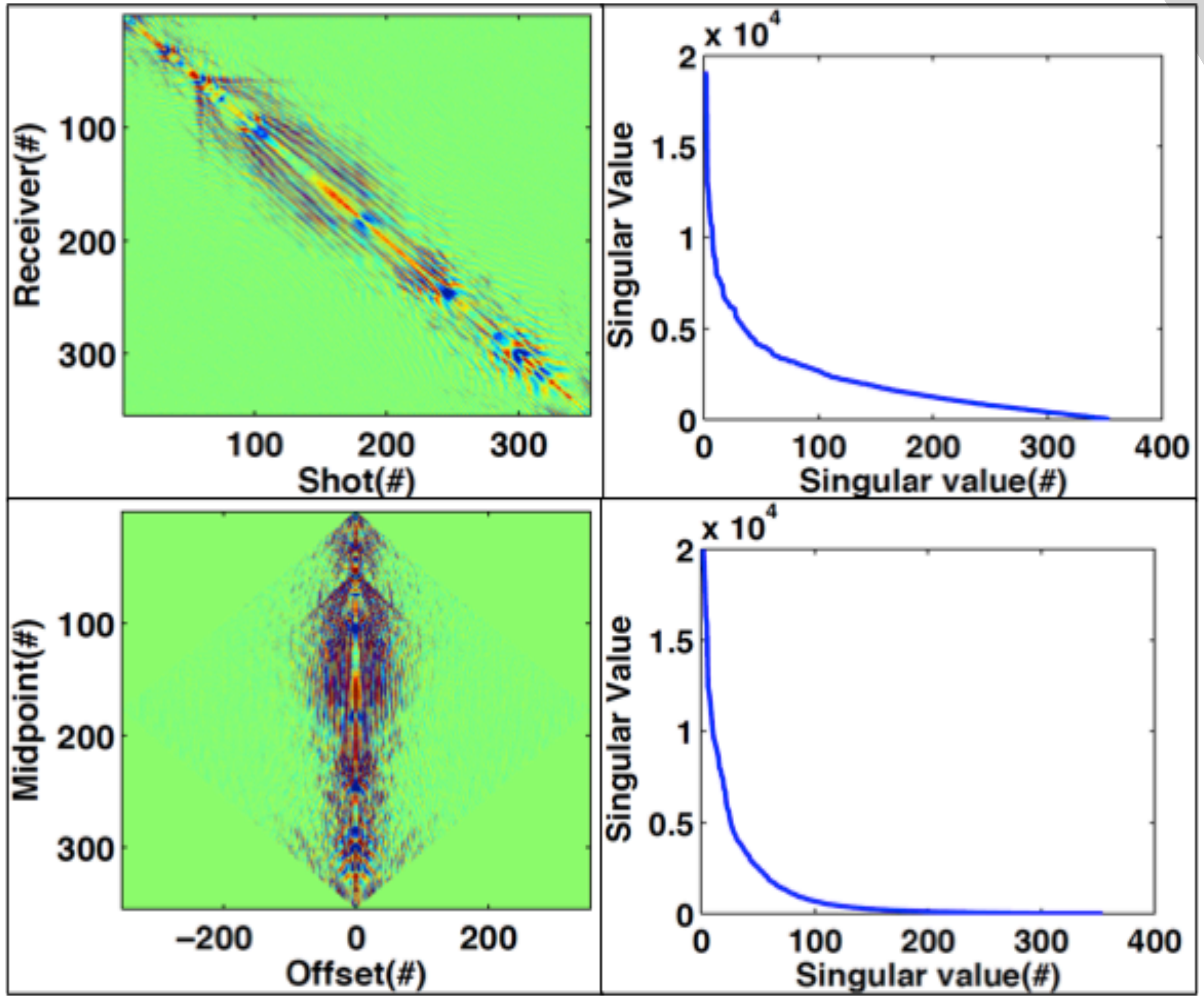
$$\min_{L,R} \frac{1}{2}\left(\|L\|_F^2 + \|R\|_F^2\right)$$

$$\text{such that } \|b - \mathcal{F}(LR^*)\|_2 \leq \sigma \ .$$

- Advantages: no SVD required; trivial projection; potential to use factors L, R downstream.

Rank Optimization in Midpoint-Offset

- Seismic data have faster singular value decay in midpoint-offset domain
- We recover 50% missing data by solving the rank optimization problem for high (70) and low (20) frequencies.
- nr = ns = 354.



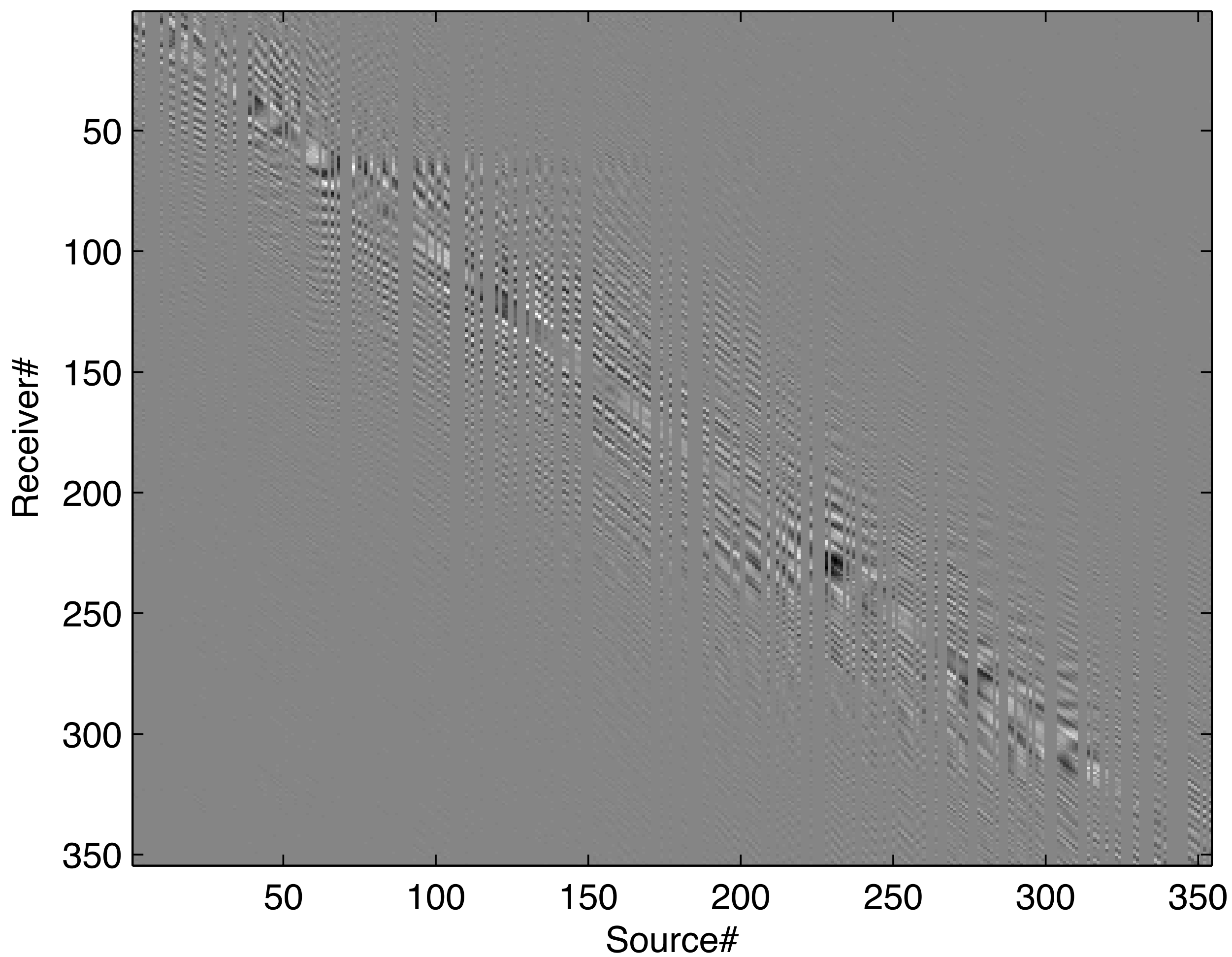
Complete data before and after transformation

Work flow:

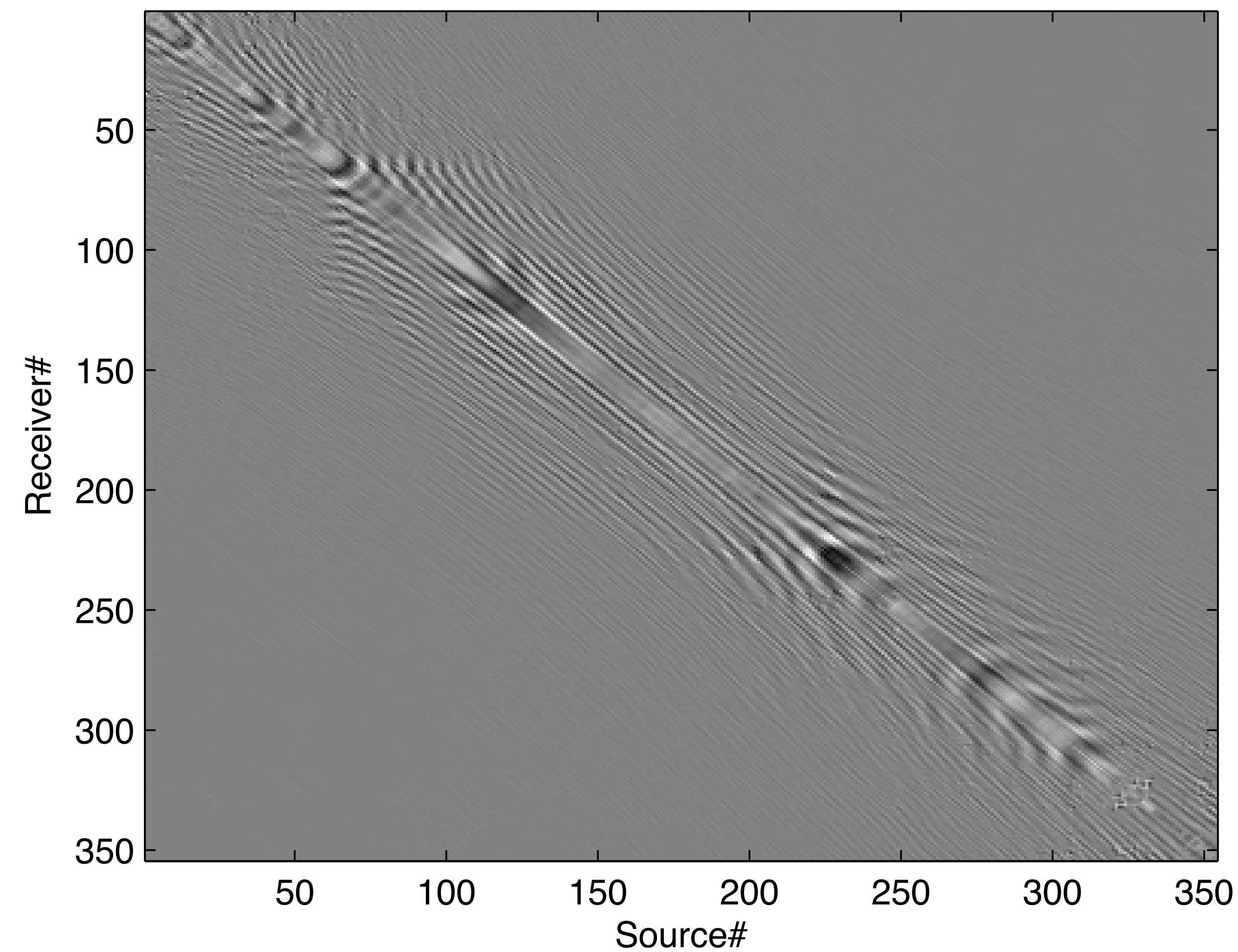
- Convert data with missing traces to M-O domain.
- Initialize L, R factors of pre-selected rank.
- Run rank optimization algorithm (SPGL1+).
- Form dense solution $X = LR^*$
- Convert solution back to source-receiver domain.

Gulf of Suez: Least Squares + Low Rank

Frequency Slice : 70 Hz, Rank : 20



50% Missing data, before interpolation

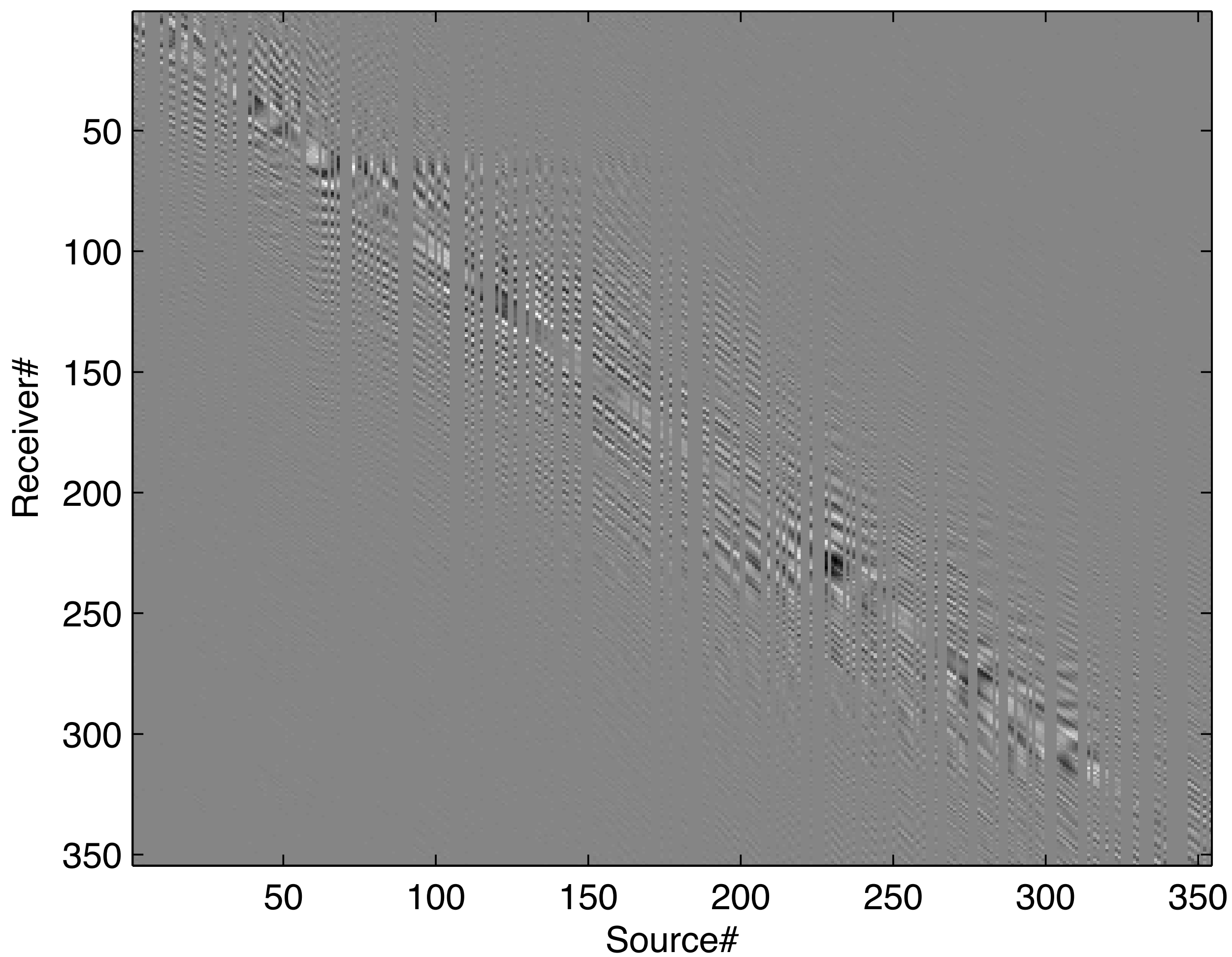


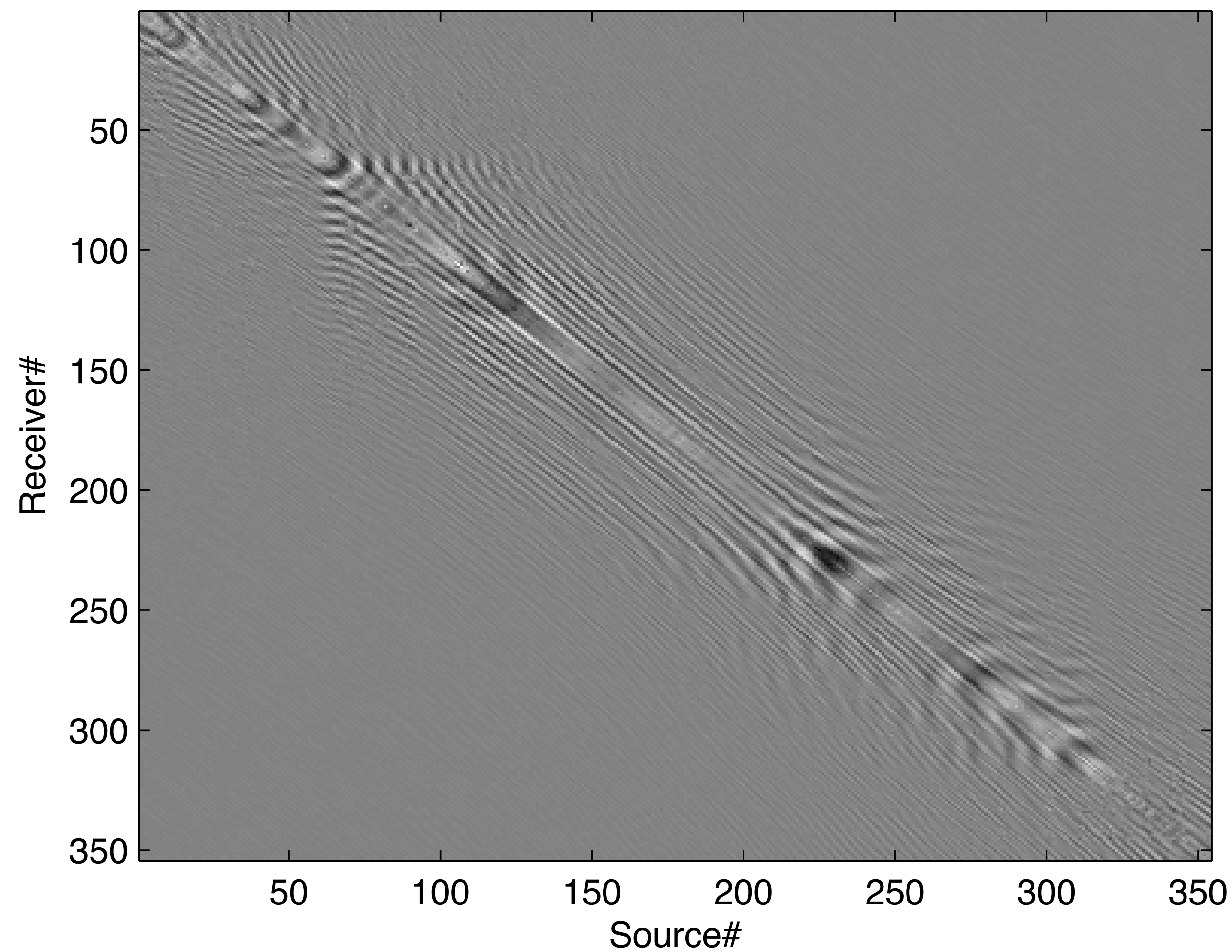
Data after interpolation, **SNR = 22.7 db**

- 150 SPGL1 iterations; sigma = 1e-6, nr = ns = 354.

Gulf of Suez: Least Squares + Low Rank

Frequency Slice : 70 Hz, Rank : 40





50% Missing data, before interpolation

Data after interpolation, **SNR = 29.3 db**

- 150 SPGL1 iterations; sigma = 1e-6, nr = ns = 354.

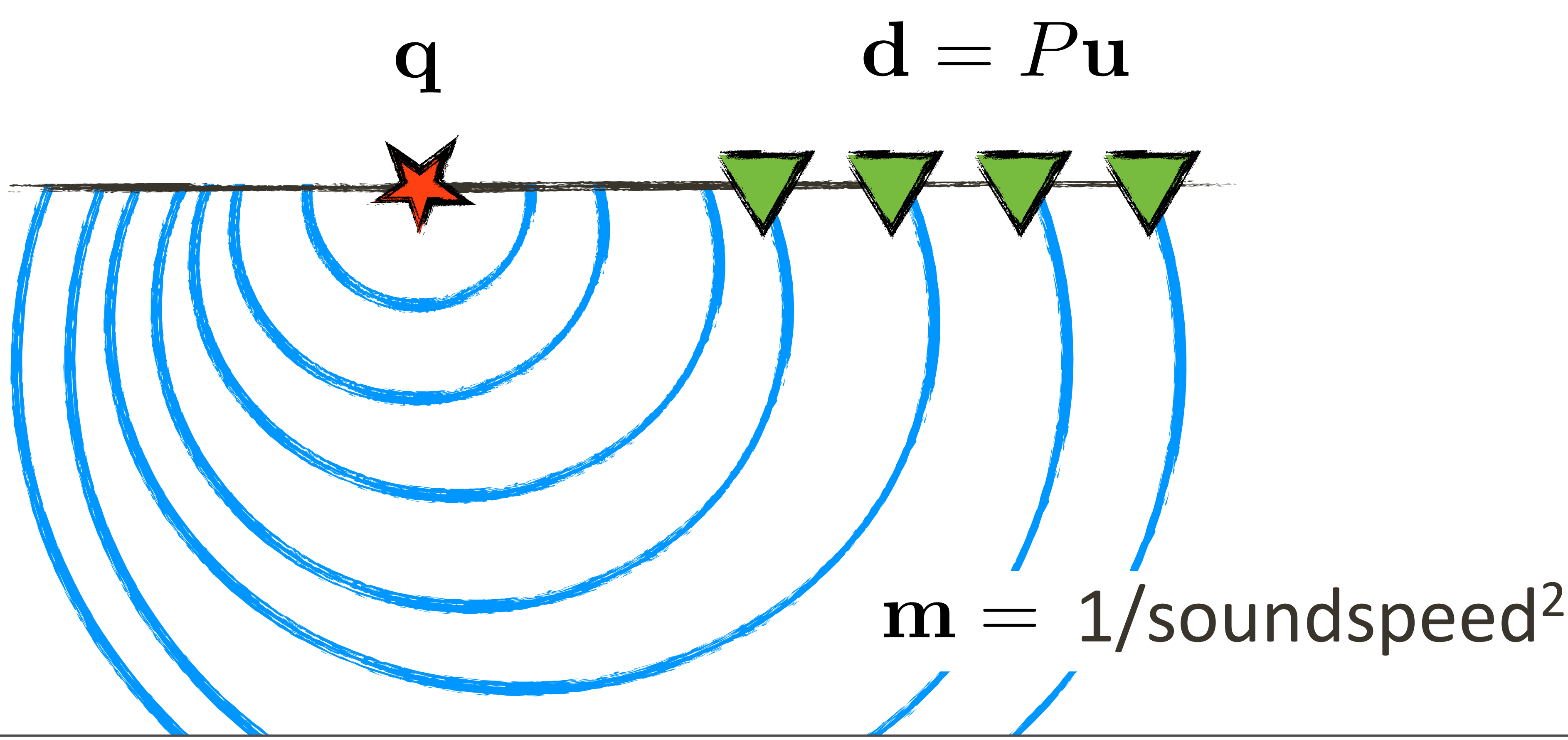
Wave-equation based inversion

PDE *constrained* inversion

- *Batching* techniques that exploit *separable* structure & *linearity* in the sources
- CS techniques to *reduce* size of GN subproblems & *linearity* in the sources
- AMP techniques to speed up convergence by using *redundancy* in data

Full-waveform inversion

We model the data in the *acoustic* approximation $(\omega^2\mathbf{m} + \nabla^2)\mathbf{u} = \mathbf{q}$



Full-waveform inversion

Realistic scale (3D):

- $\mathbf{m} \sim \mathcal{O}(10^9)$ unknowns
- $\mathbf{d} \sim \mathcal{O}(10^{15})$ measurements
- 3D Helmholtz equation is non-trivial to solve.

Batched optimization

$$\min_{\mathbf{m}} \Phi[\mathbf{m}] = \frac{1}{K}\sum_{i=1}^{K} \phi_i[\mathbf{m}]$$

Quasi-Newton approach

$$\mathbf{s}_k = -B_k \nabla\Phi[\mathbf{m}_k]$$

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \lambda_k \mathbf{s}_k$$

But: evaluation of *full* misfit and gradient is very expensive.

Full waveform inversion

The gradient can be calculated via the adjoint state method

$$\frac{\partial \phi_i}{\partial m_k} = \mathbf{u}_i^H \left(\frac{\partial A[\mathbf{m}]}{\partial m_k} \right)^H \mathbf{v}_i$$

$$A[\mathbf{m}]\mathbf{u}_i = \mathbf{q}_i$$

$$A[\mathbf{m}]^H \mathbf{v}_i = P^T(\mathbf{d}_i - F[\mathbf{m}]\mathbf{q}_i)$$

Optimization

The gradient is the *average*

$$\nabla\Phi = \frac{1}{K}\sum_{i=1}^{K}\nabla\phi_i$$

which we can approximate by

$$\nabla\Phi \approx \nabla\widetilde{\Phi} = \frac{1}{|\mathcal{I}|}\sum_{i\in\mathcal{I}}\nabla\phi_i$$

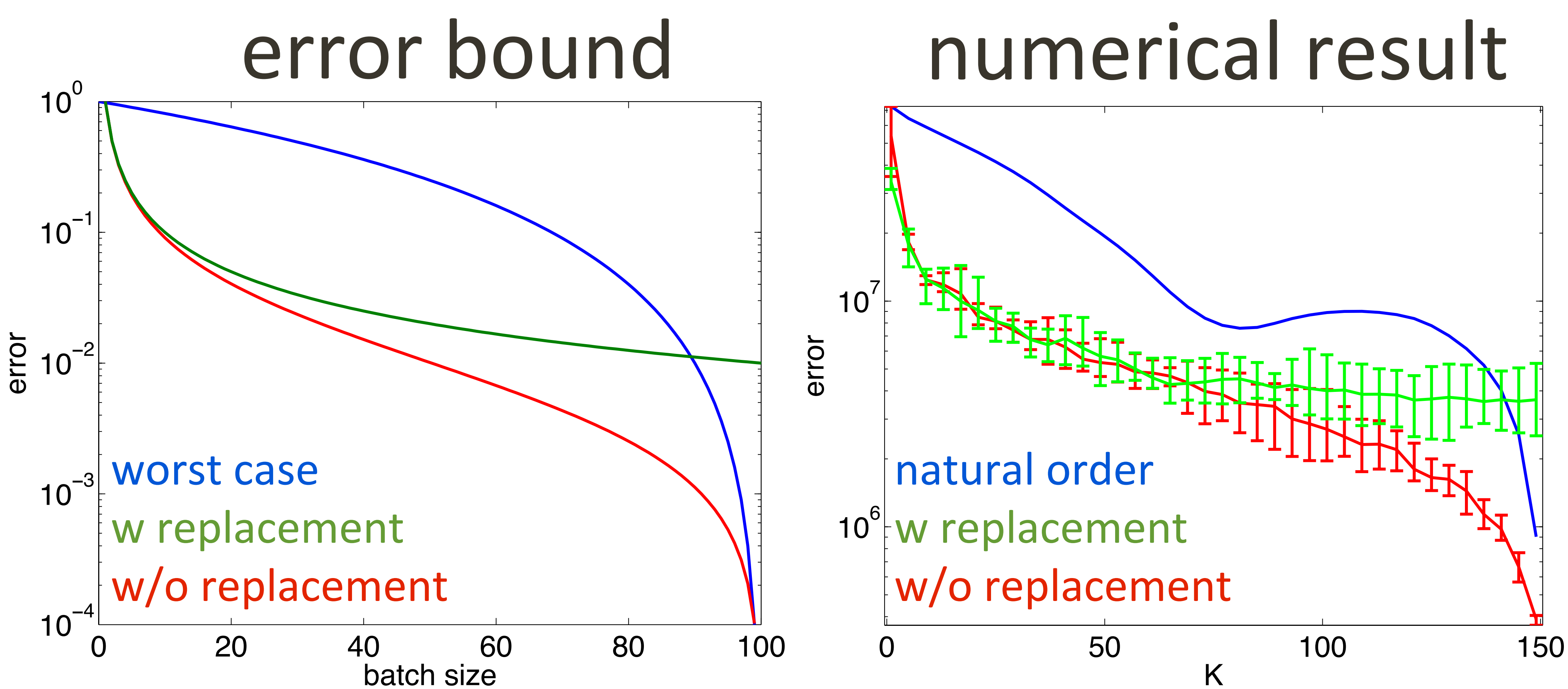
Optimization

Grow the sample by adding elements

- in a pre-scribed order
- chosen at random *without* replacement
- chosen at random *with* replacement

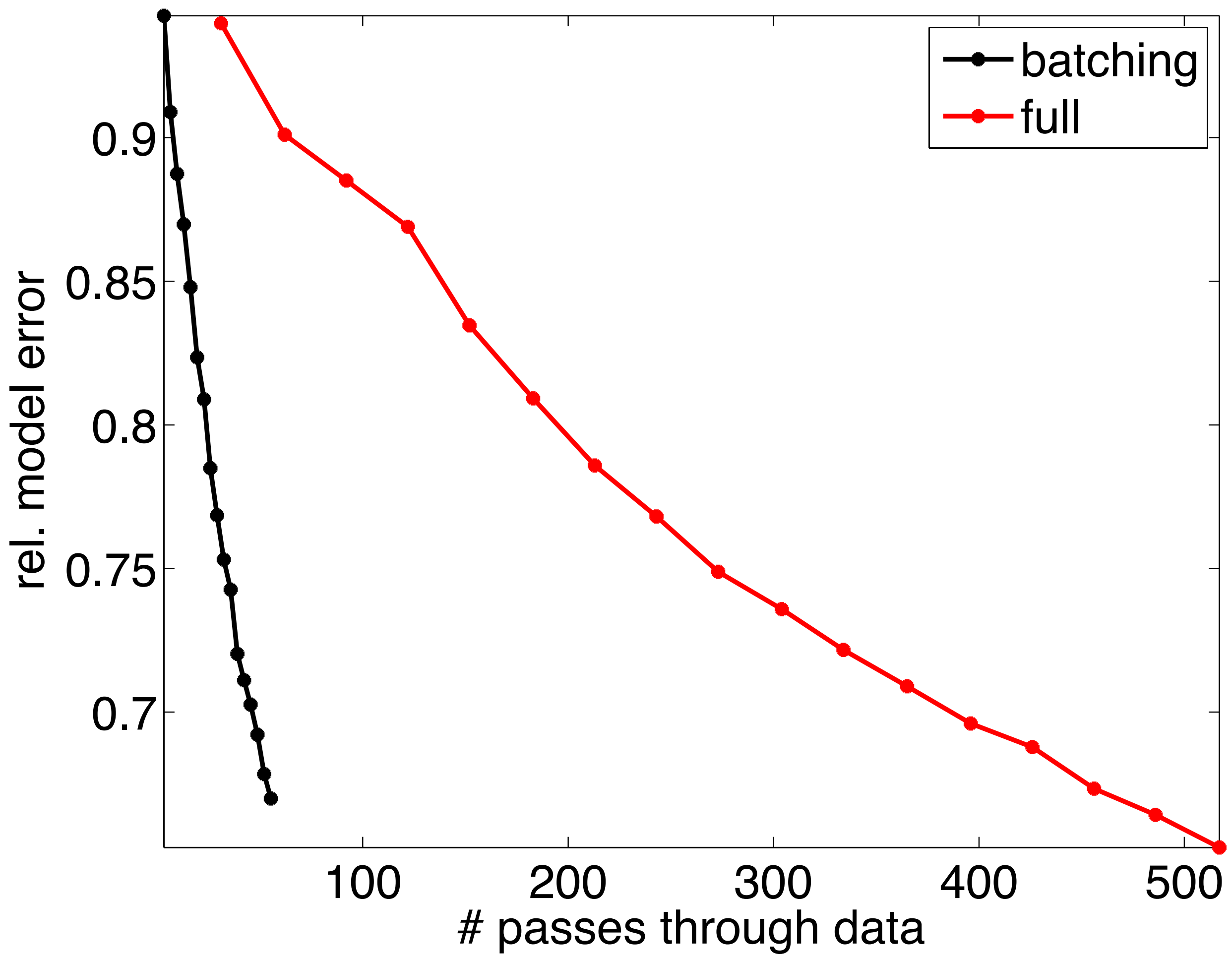
Optimization

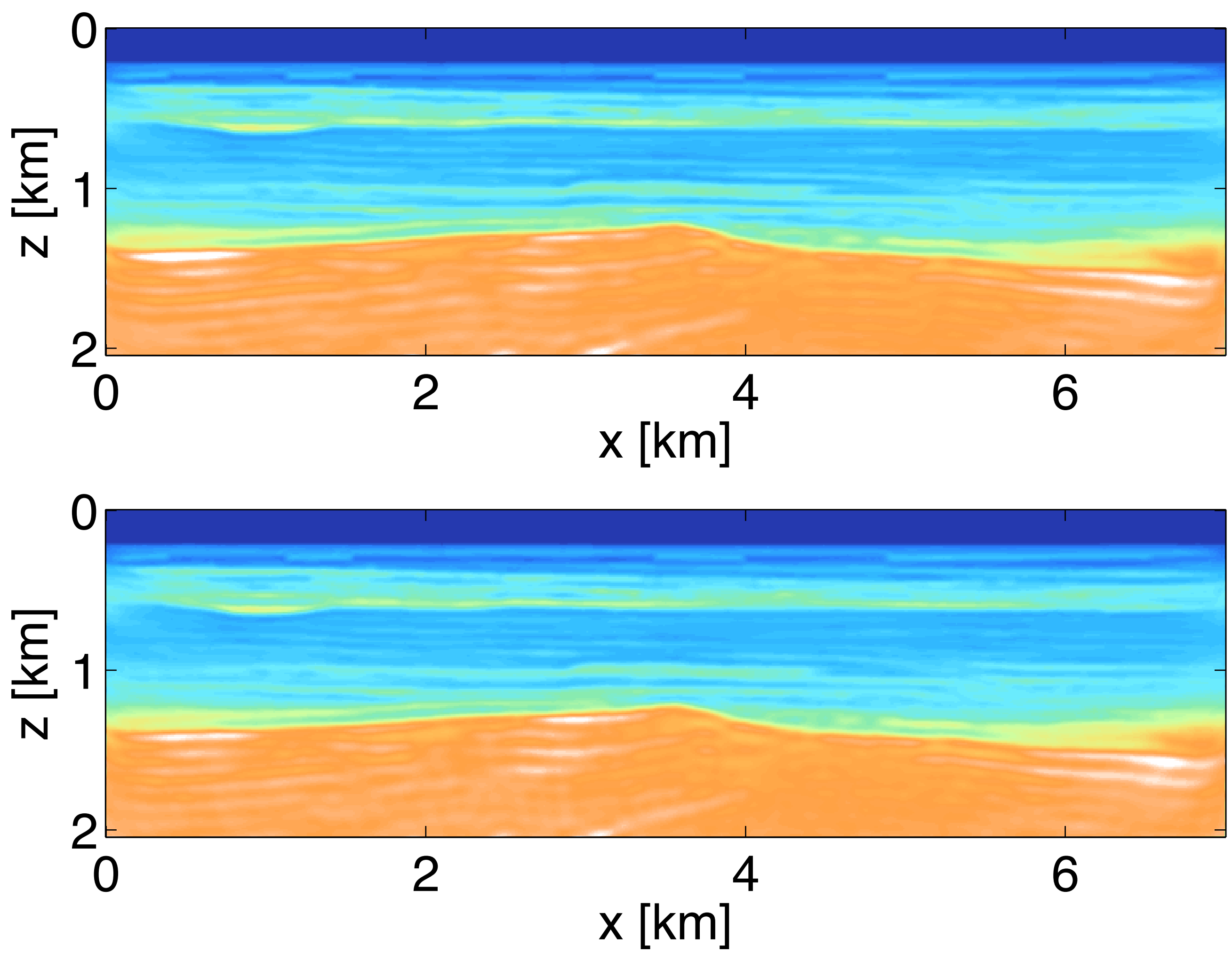
Error in the gradient



Optimization

10 x speedup





[van Leeuwen et al ’11]

Imaging & FWI

with compressive sensing

Work on *small* subsets of data and use *sparsity* promotion to control errors of Gauss-Newton updates

- works for *simultaneous* & *sequential* (marine) data

Use *separable* structure of FWI and use techniques from

- stochastic optimization & compressive sensing [Bertsekas, '96, Nemirovsky, '08, Candes et.al., '06, Donoho, '06]
- *approximate* message passing [Donoho et. al. '09, Montanari, '12]
- phase encoding [Krebs et.al., '09, Operto et. al., '09, Herrmann et.al., '08-10']

Random source-encoded imaging

Replace GN update with *all* data (overdetermined system)

$$\widetilde{\mathbf{x}}_{\mathrm{mig}} = \mathbf{A}^*\mathbf{b} \quad \text{approximating} \quad \underset{\mathbf{x}}{\text{minimize}} \ \frac{1}{2K}\sum_{i=1}^{K}\|\mathbf{b}_i - \mathbf{A}_i\mathbf{x}\|_2^2$$

with *K* large by *sparsity*-promoting GN (underdetermined)

$$\boxed{\underset{\mathbf{x}}{\text{minimize}} \ \|\mathbf{x}\|_1 \quad \text{subject to} \quad \underline{\mathbf{b}}_i = \underline{\mathbf{A}}_i\mathbf{x}, \quad i = 1\cdots K'}$$

with $K' \ll K$ and $\{\underline{\mathbf{b}}_i, \underline{\mathbf{A}}_i\}$ *supershots* & *linearized* Born *scattering* operators

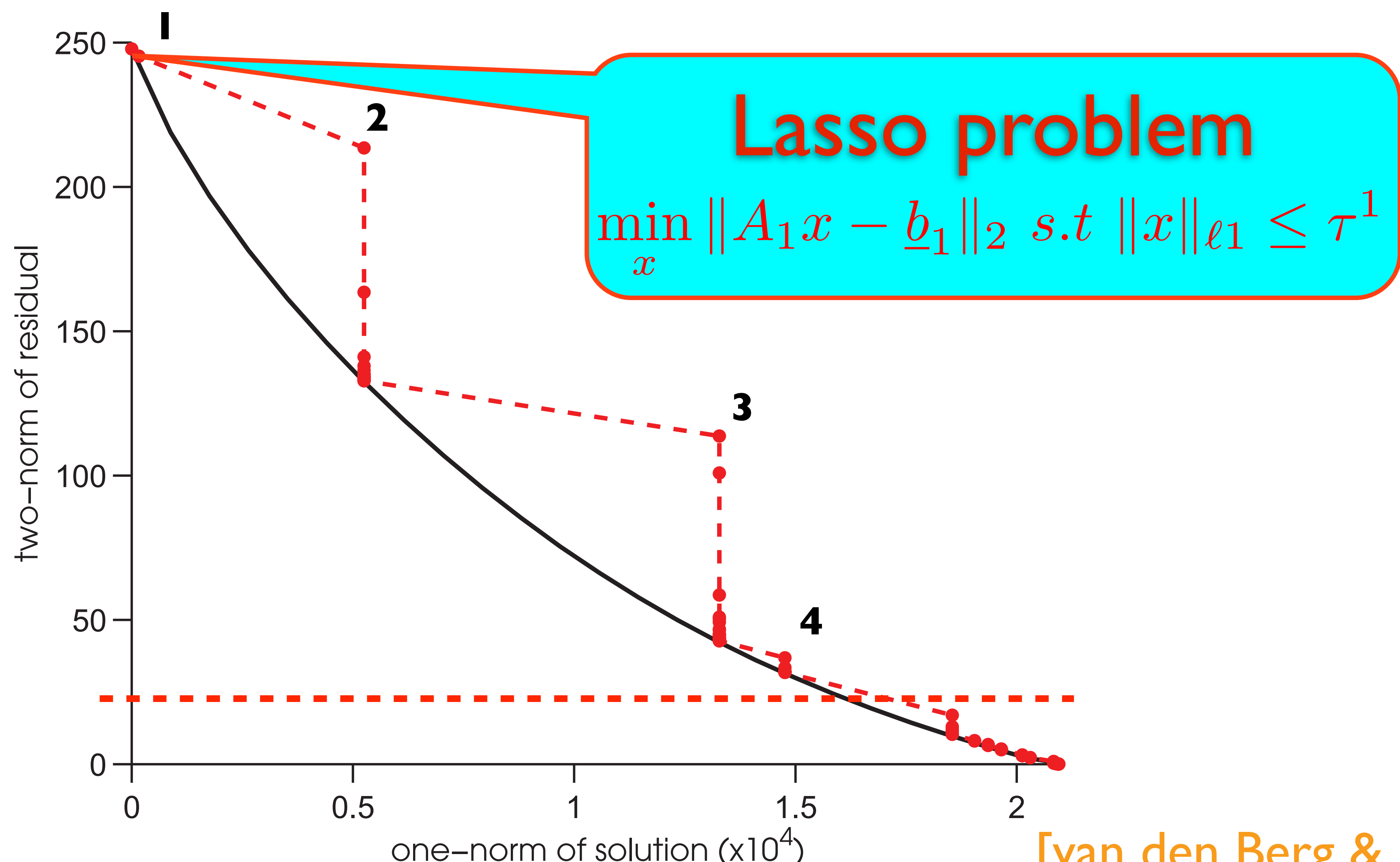
Continuation methods

Versatile large-scale *sparsity*-promoting solvers *limit* the number of *matrix-vector* multiplies by *cooling*, which

- slowly allows *components* to *enter* into the *solution*
- solves an *intelligent* series of LASSO *subproblems* for *decreasing* sparsity levels
- uses *convexity* & *smoothness* of Pareto curves with Newton root finding

Supercooled

spectral-projected gradients



[van den Berg & Friedlander, ’08]

[Hennefent et. al., ’08]

[Lin & FJH, ’09-]

Problems

One-norm solvers suffer from:

- *first-order* spectral-gradient methods need many *iterations*
- *second-order* quasi-Newton need to store *multiple* model vectors
- *correlation* buildup that slows down *convergence*

Can *insights* from *AMP* be used to *accelerate* current state-of-the art *one-norm* solvers?

Compressive imaging
[*with* message passing]

Select *independent* random source *encodings* after *each* LASSO *subproblem* is solved

- calculate corresponding *supershots*
- *redefine* Jacobian operator (and its *adjoint*)
(select *independent* simultaneous sources & supershots)

Promote *sparsity* in the *curvelet* domain

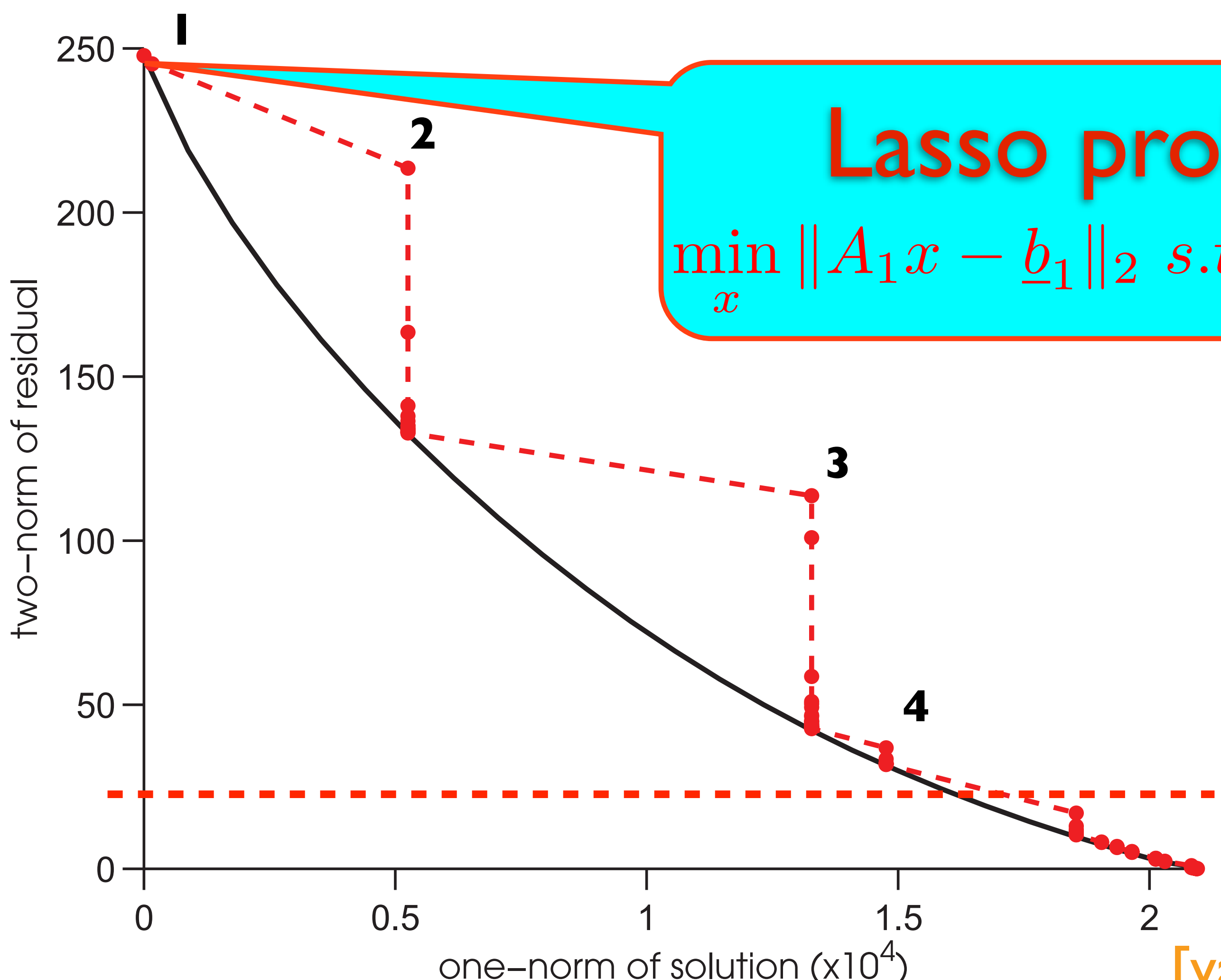
Supercooling

Break *correlations* between the model *iterate* and matrix **A** by *rerandomization*

- draw new *independent* $\{\mathbf{b}_t, \mathbf{A}_t\}$ after each LASSO subproblem is solved
- brings in "*extra*" information *without* growing the *system*
- ***minimal*** *extra* computational & memory cost

Supercooled

spectral-projected gradients



Lasso problem

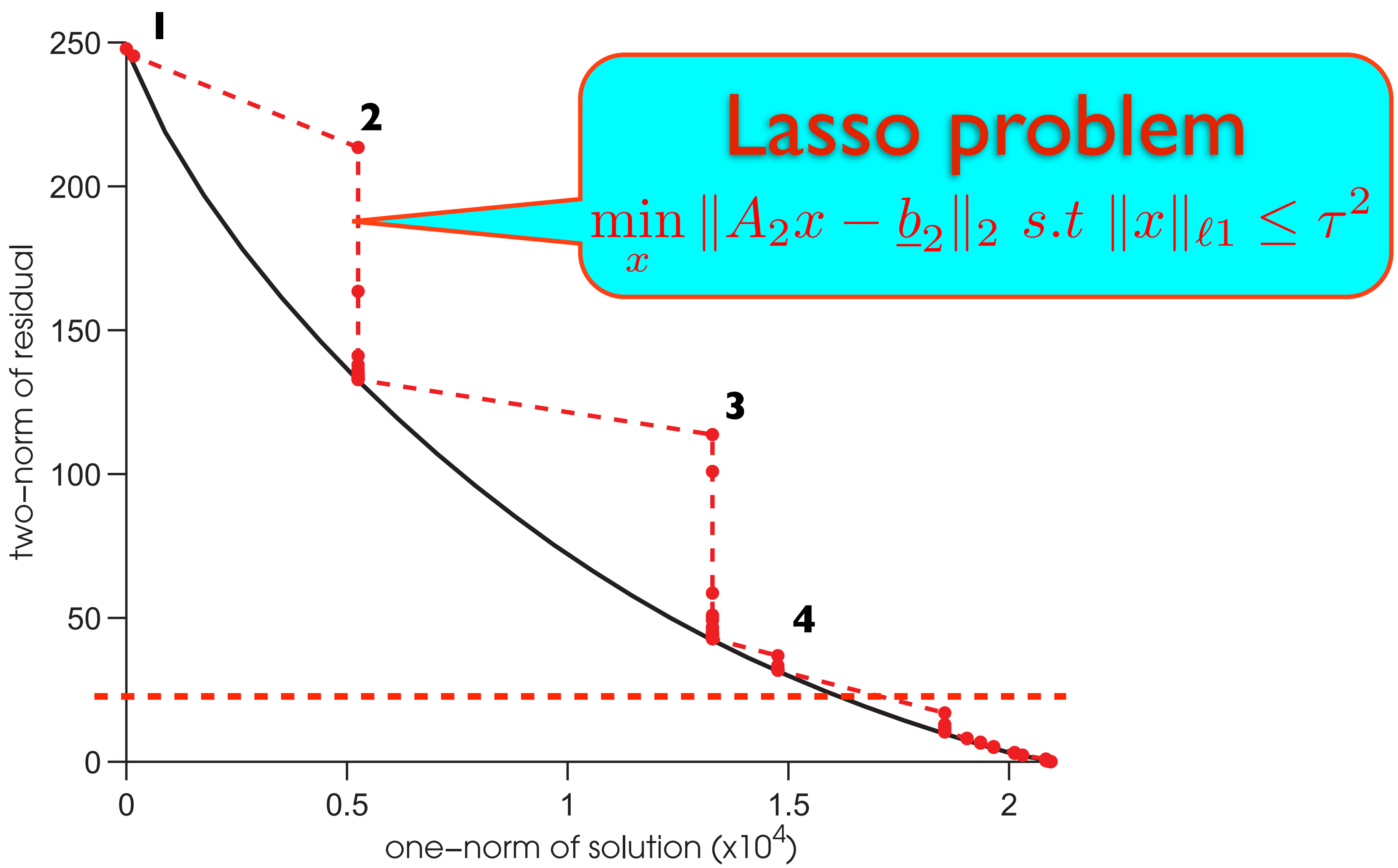
$$\min_{x} \|A_1 x - \underline{b}_1\|_2 \ s.t \ \|x\|_{\ell 1} \leq \tau^1$$

[van den Berg & Friedlander, '08]

[Hennefent et. al., '08]

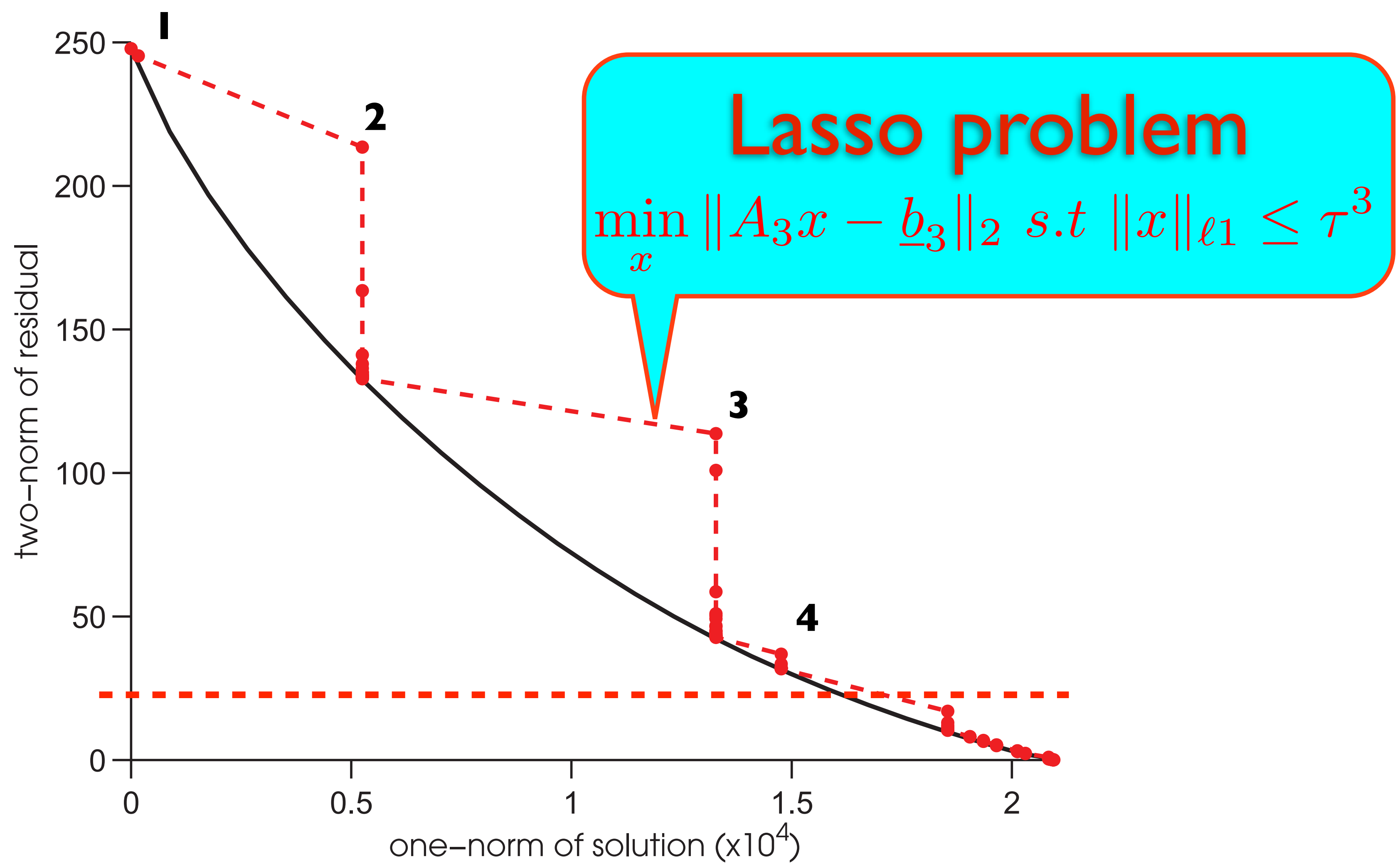
[Lin & FJH, '09-]

Supercooled
spectral-projected gradients



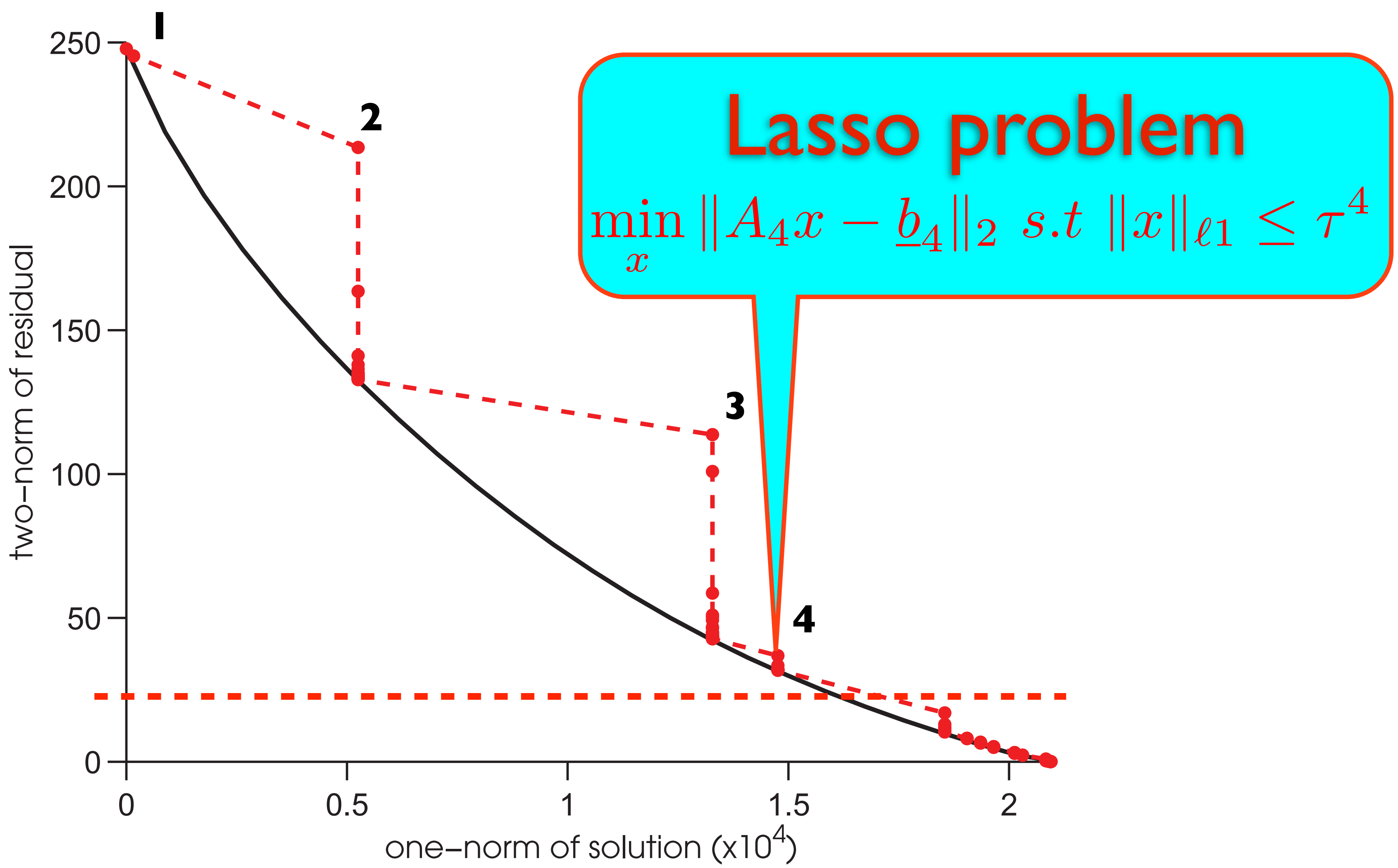
Supercooled

spectral-projected gradients



Supercooled

spectral-projected gradients



Compressive imaging
[*with* message passing]

Result: Estimate for the model $\mathbf{x}^{t+1}$

```
x^0, x̃ ⟵ 0 and t, τ^0 ⟵ 0 ;                                  // Initialize
while t ≤ T do
    W ⟵ W ∈ ℝ^{K×K'} with W_ij ~ N(0, 1/√K') ;               // Random encoding
    {b, q} ⟵ {DW, QW} ;                                      // Draw sim sources and data
    A ⟵ ∇F[m_0; q];                                          // New demigration operator
    x^{t+1} ⟵ spgl1(A, b, τ^t, σ = 0, x^t);                  // Reach Pareto
    τ^t ⟵ ‖x^{t+1}‖_1;                                       // New initial τ value
    t ⟵ t + ΔT ;                                             // Add # of iterations of spgl1
end
```

$\mathbf{x}^0, \widetilde{\mathbf{x}} \longleftarrow \mathbf{0}$ and $t, \tau^0 \longleftarrow 0$; // Initialize

while $t \leq T$ **do**

- $\mathbf{W} \longleftarrow \mathbf{W} \in \mathbb{R}^{K \times K'}$ with $W_{ij} \sim N(0, 1/\sqrt{K'})$; // Random encoding
- $\{\underline{\mathbf{b}}, \underline{\mathbf{q}}\} \longleftarrow \{\mathbf{DW}, \mathbf{QW}\}$; // Draw sim sources and data
- $\underline{\mathbf{A}} \longleftarrow \nabla\boldsymbol{\mathcal{F}}[\mathbf{m}_0; \underline{\mathbf{q}}]$; // New demigration operator
- $\mathbf{x}^{t+1} \longleftarrow \mathrm{spgl1}(\underline{\mathbf{A}}, \underline{\mathbf{b}}, \tau^t, \sigma = 0, \mathbf{x}^t)$; // Reach Pareto
- $\tau^t \longleftarrow \|\mathbf{x}^{t+1}\|_1$; // New initial τ value
- $t \longleftarrow t + \Delta T$; // Add # of iterations of spgl1

end

Algorithm 1: Supercooled sparsity-promoting migration.

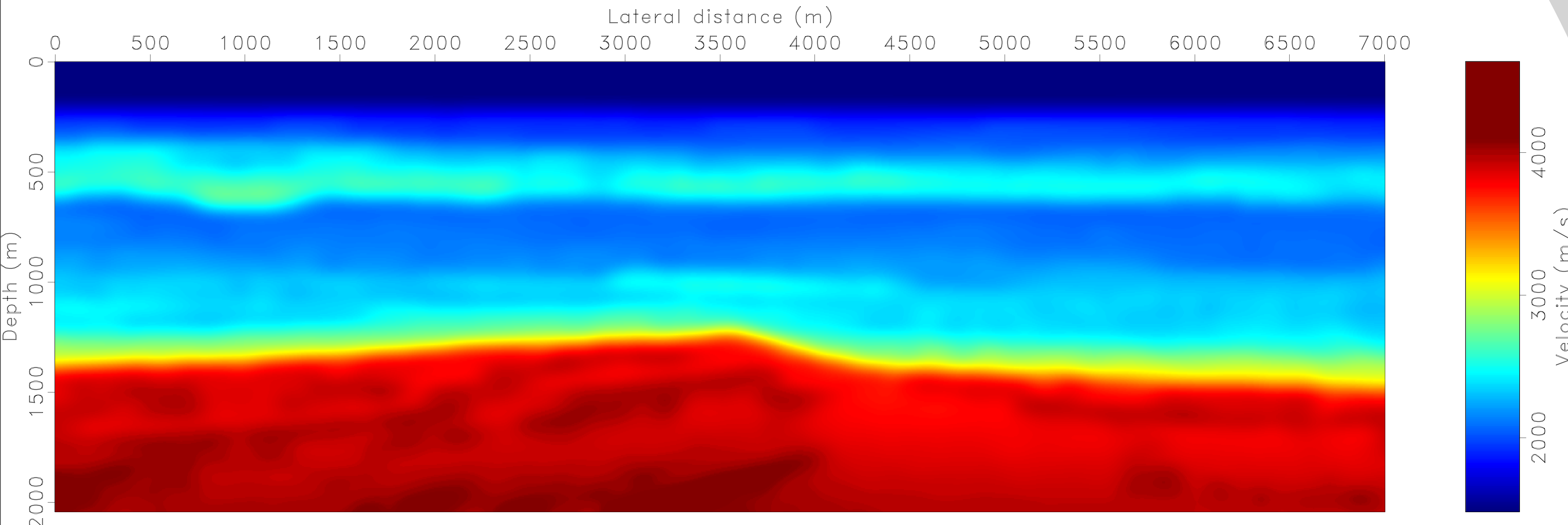
Imaging results

Time-harmonic Helmholtz:

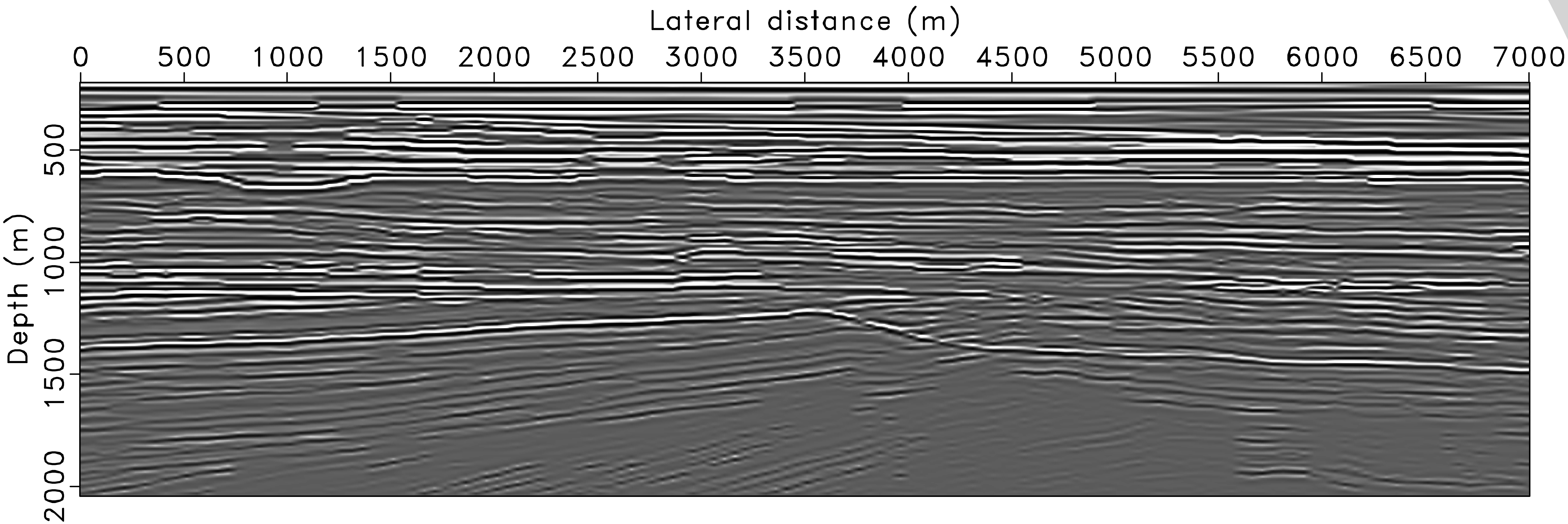
- 409 X 1401 with mesh size of 5m
- 9 point stencil [C. Jo et. al., '96]
- absorbing boundary condition with damping layer with thickness proportional to wavelength
- solve wavefields on the fly with direct solver

SLIM

Imaging results
[background model]



Migration results
[*true* perturbation]

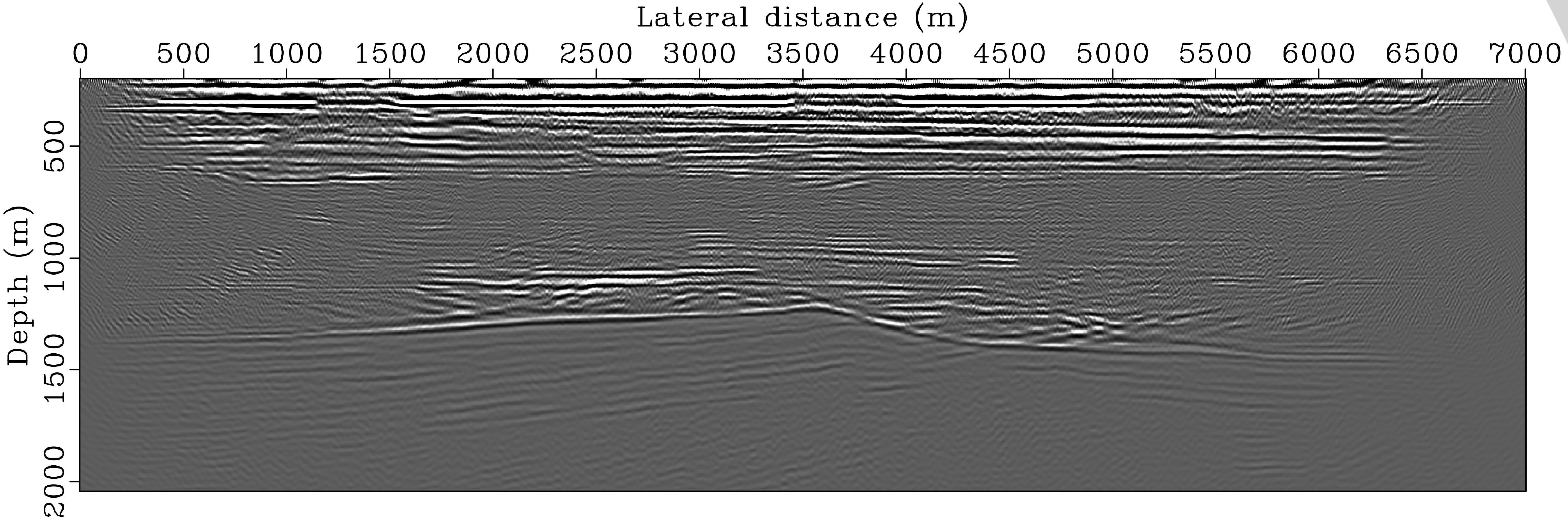


Imaging results

Split-spread surface-free 'land' acquisition:

- 350 sources with sampling interval 20m
- 701 receivers with sampling interval 10m
- maximal offset 7km (3.5 X depth of model)
- Ricker wavelet with central frequency of 30Hz
- recording time for each shot is 3.6s

Migration results
[migration with *all* data]



Imaging results

Reduced setup:

- 10 *random* frequencies *(versus 300 frequencies)* (20Hz-50Hz)
- 3 random *simultaneous* shots (*versus 350 sequential shots*)

Significant dimensionality reduction of

$$\boxed{\frac{K'}{K} = 0.0003}$$

[Herrmann & Li, 2011]

Imaging results

Least-squares migration with *randomized supershots:*

$$\delta\widetilde{\mathbf{m}} = \mathbf{S}^* \operatorname*{arg\,min}_{\delta\mathbf{x}} \|\delta\mathbf{x}\|_{\ell_2} \quad \text{subject to} \quad \|\delta\underline{\mathbf{d}} - \overbrace{\nabla\mathcal{F}[\mathbf{m}_0;\underline{\mathbf{Q}}]}^{\text{demigration}}\mathbf{S}^*\delta\mathbf{x}\|_2 \leq \sigma$$

$$\begin{aligned} \delta\mathbf{x} &= \text{Sparse curvelet-coefficient vector} \\ \mathbf{S}^* &= \text{Curvelet synthesis} \\ \underline{\mathbf{Q}} &= \text{Simultaneous sources} \\ \delta\underline{\mathbf{d}} &= \text{Super shots} \end{aligned}$$

Imaging results

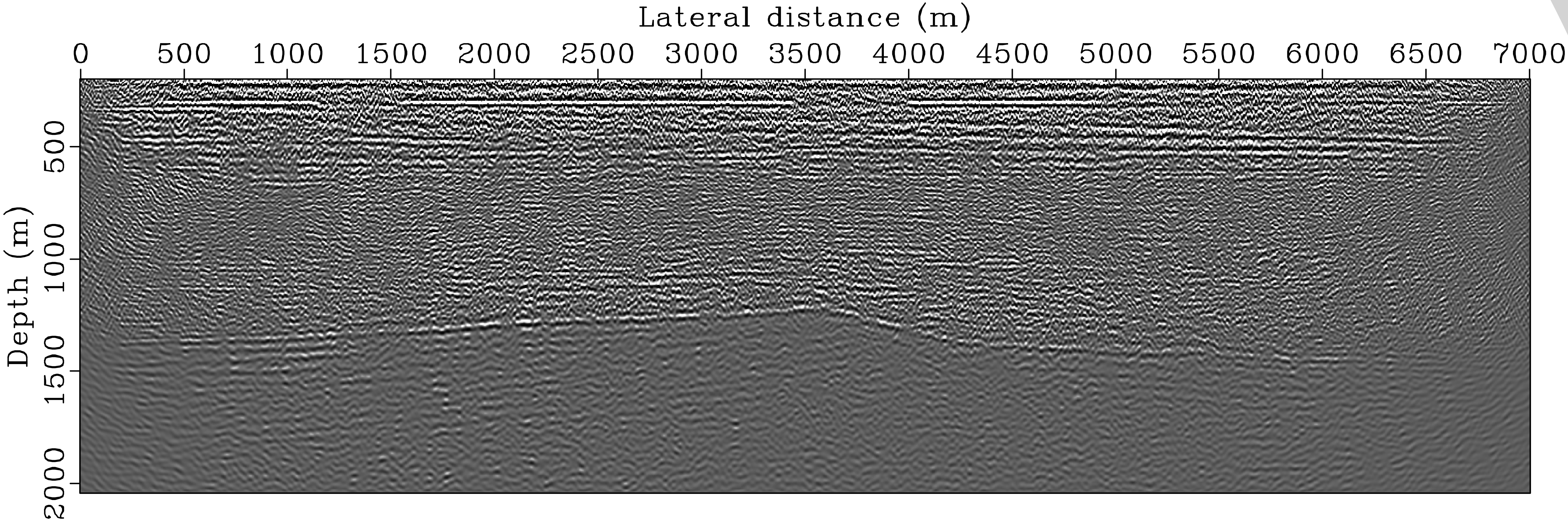
Sparsity-promoting migration with *randomized supershots:*

$$\delta\widetilde{\mathbf{m}} = \mathbf{S}^* \underset{\delta\mathbf{x}}{\arg\min} \|\delta\mathbf{x}\|_{\ell_1} \quad \text{subject to} \quad \|\delta\underline{\mathbf{d}} - \overbrace{\nabla\mathcal{F}[\mathbf{m}_0;\underline{\mathbf{Q}}]}^{\text{demigration}}\mathbf{S}^*\delta\mathbf{x}\|_2 \leq \sigma$$

$$\begin{aligned} \delta\mathbf{x} &= \text{Sparse curvelet-coefficient vector} \\ \mathbf{S}^* &= \text{Curvelet synthesis} \\ \underline{\mathbf{Q}} &= \text{Simultaneous sources} \\ \delta\underline{\mathbf{d}} &= \text{Super shots} \end{aligned}$$

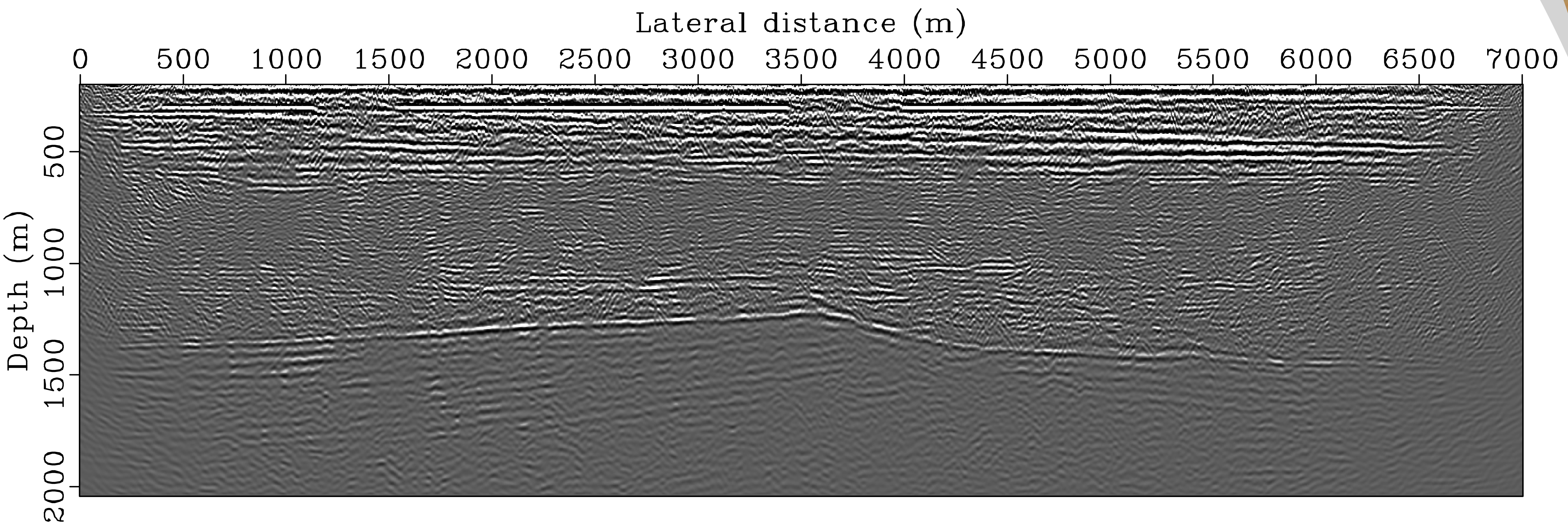
Migration results

[ℓ_2 *without* renewals]



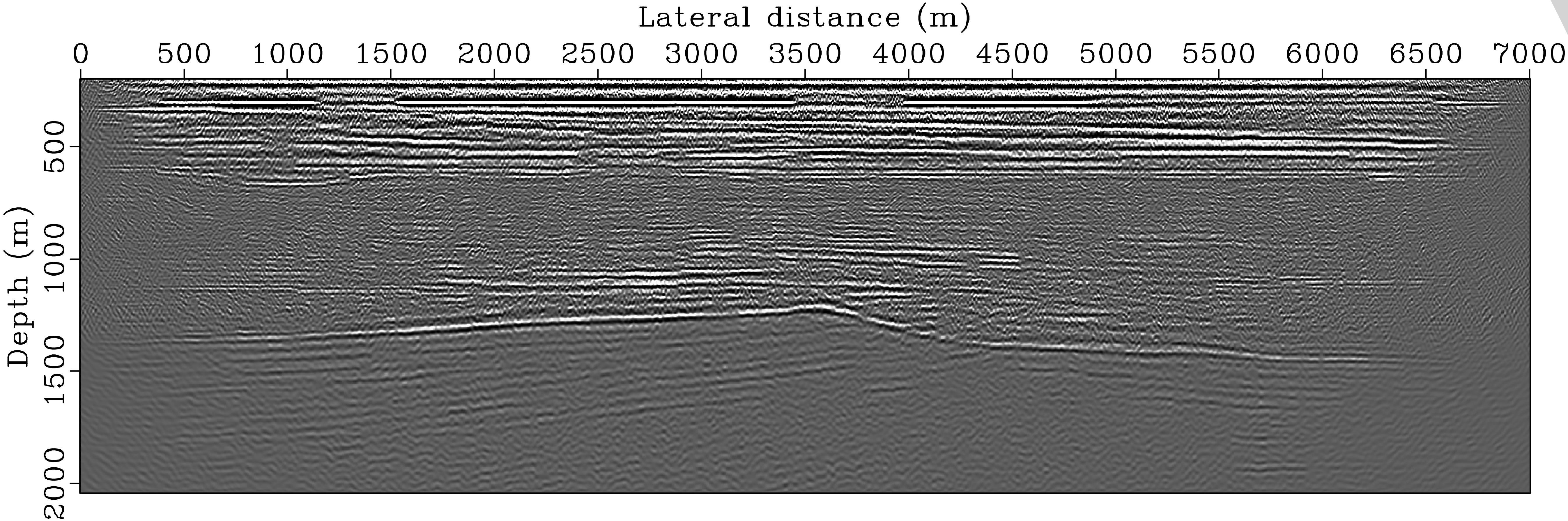
Imaging results

[ℓ_1 *without* renewals]



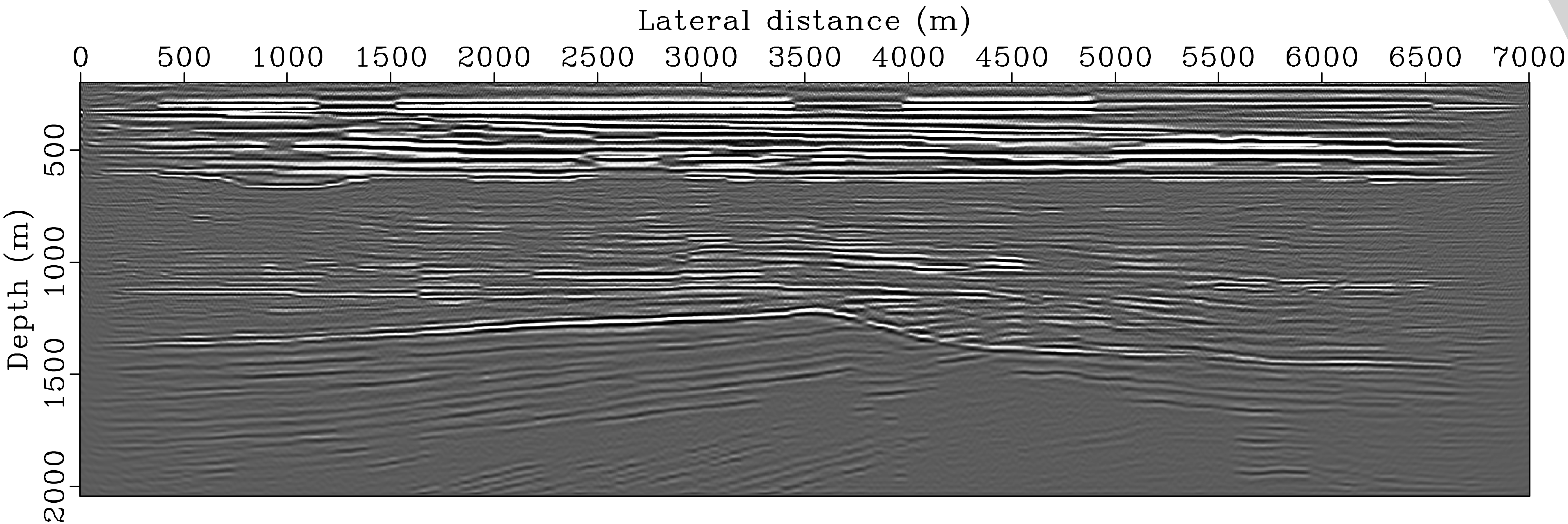
Migration results

[ℓ_2 *with* renewals]

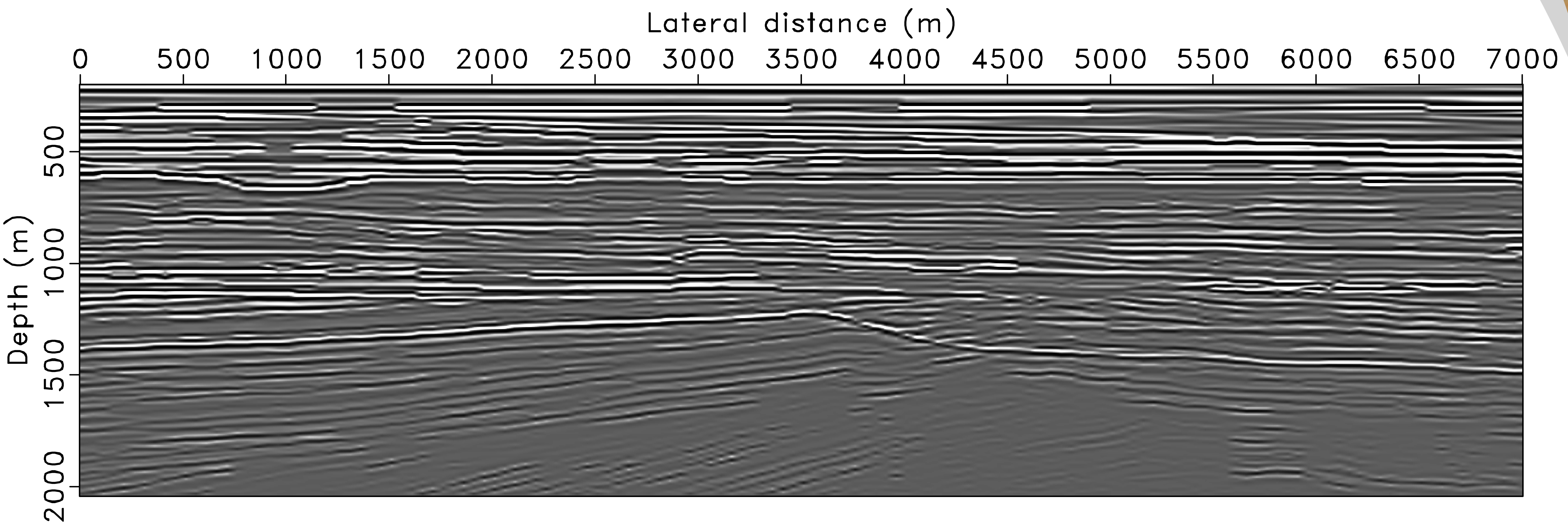


Migration results

[ℓ_1 *with* renewals]

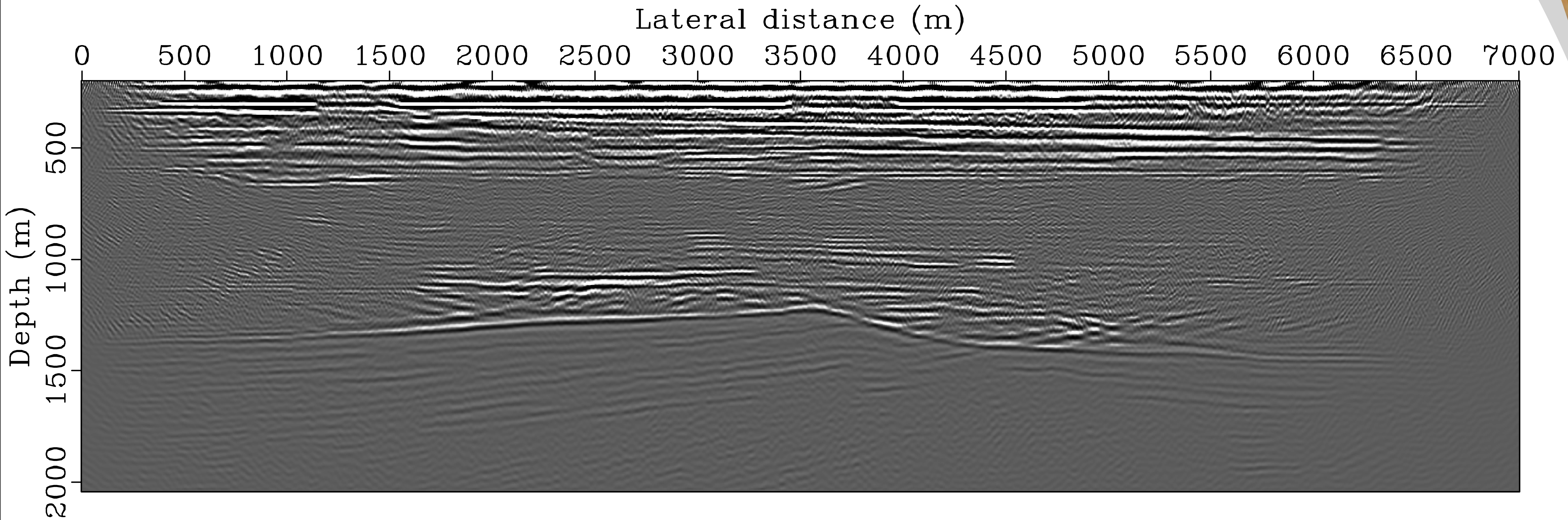


Migration results
[*true* perturbation]



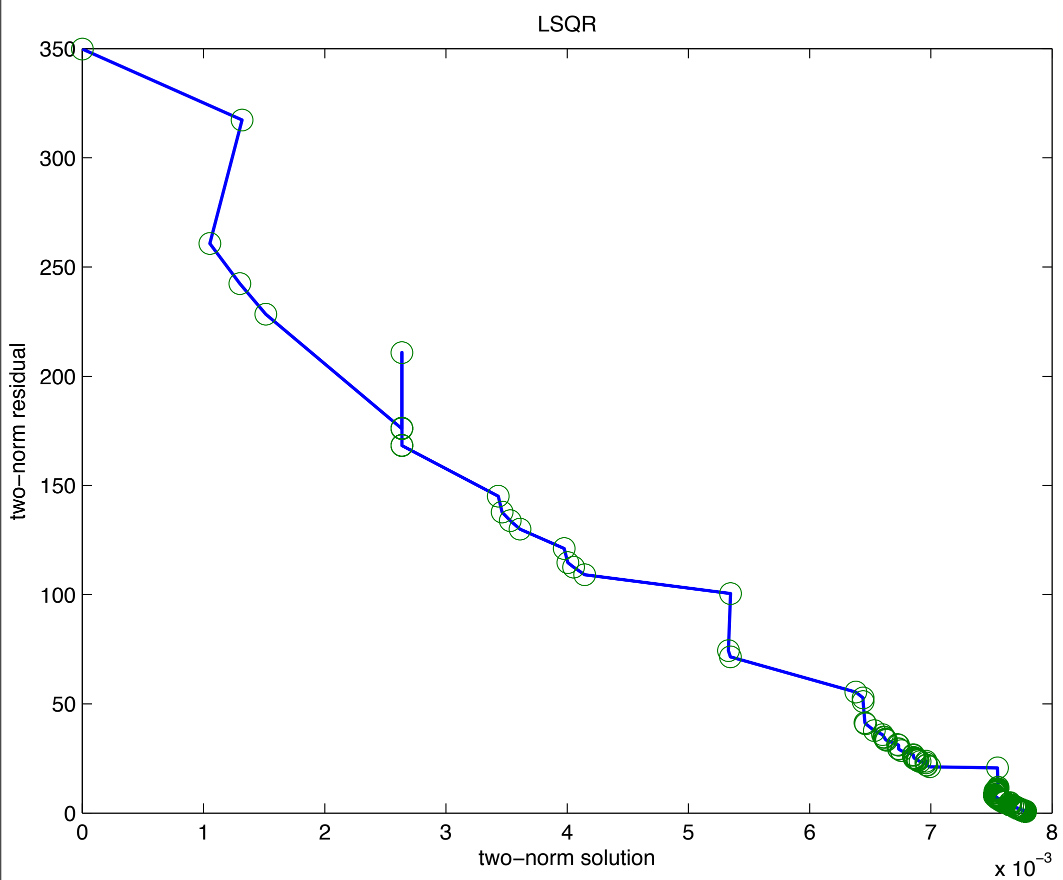
Migration results

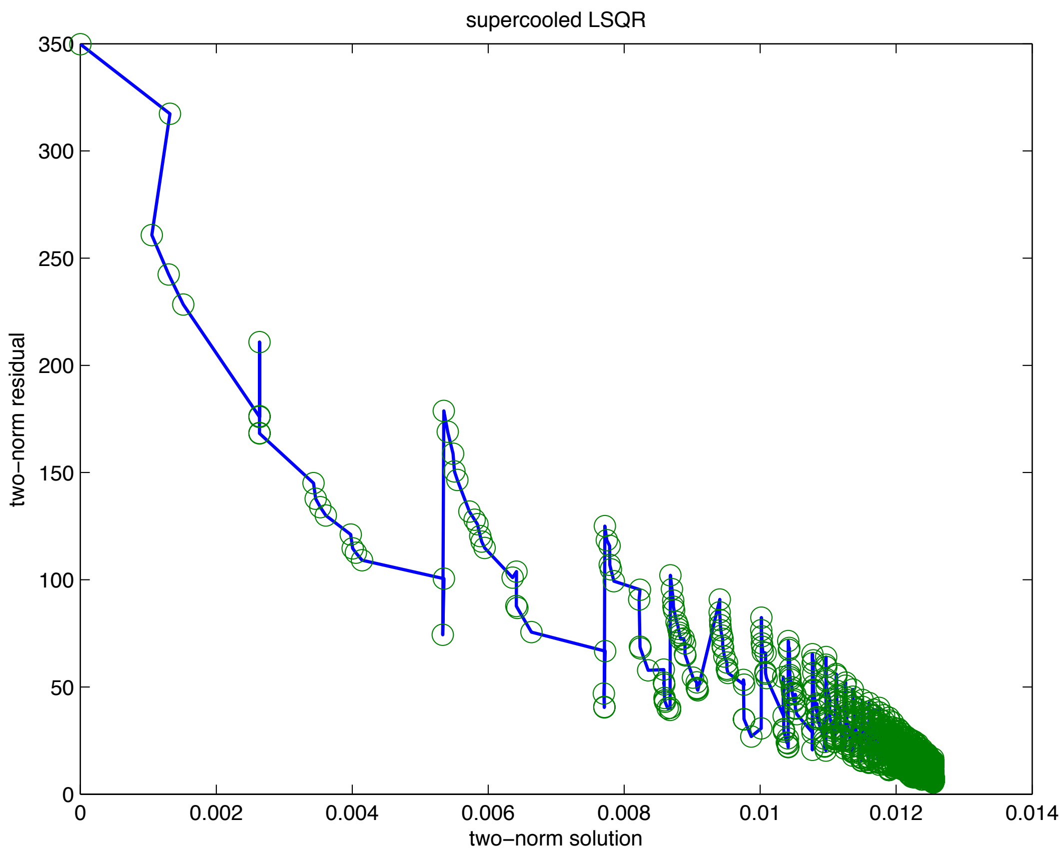
[migration with *all* data]



Migration results

[solution paths ℓ_2]



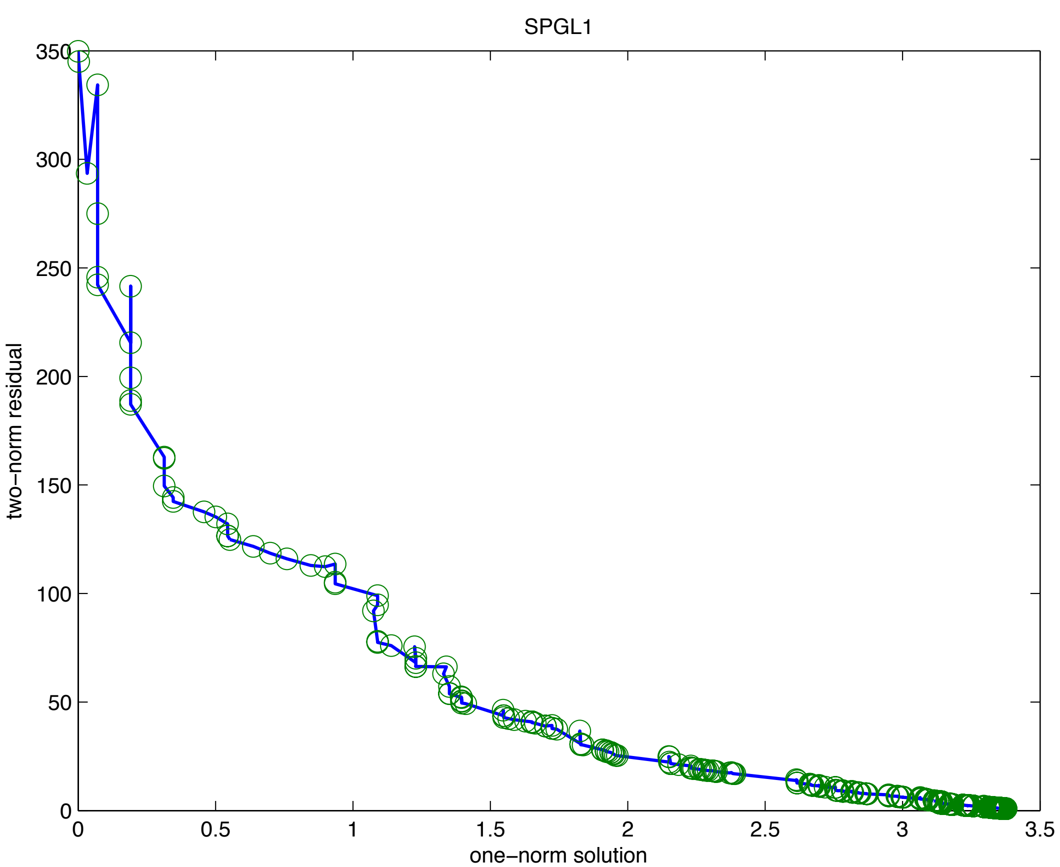


without renewals

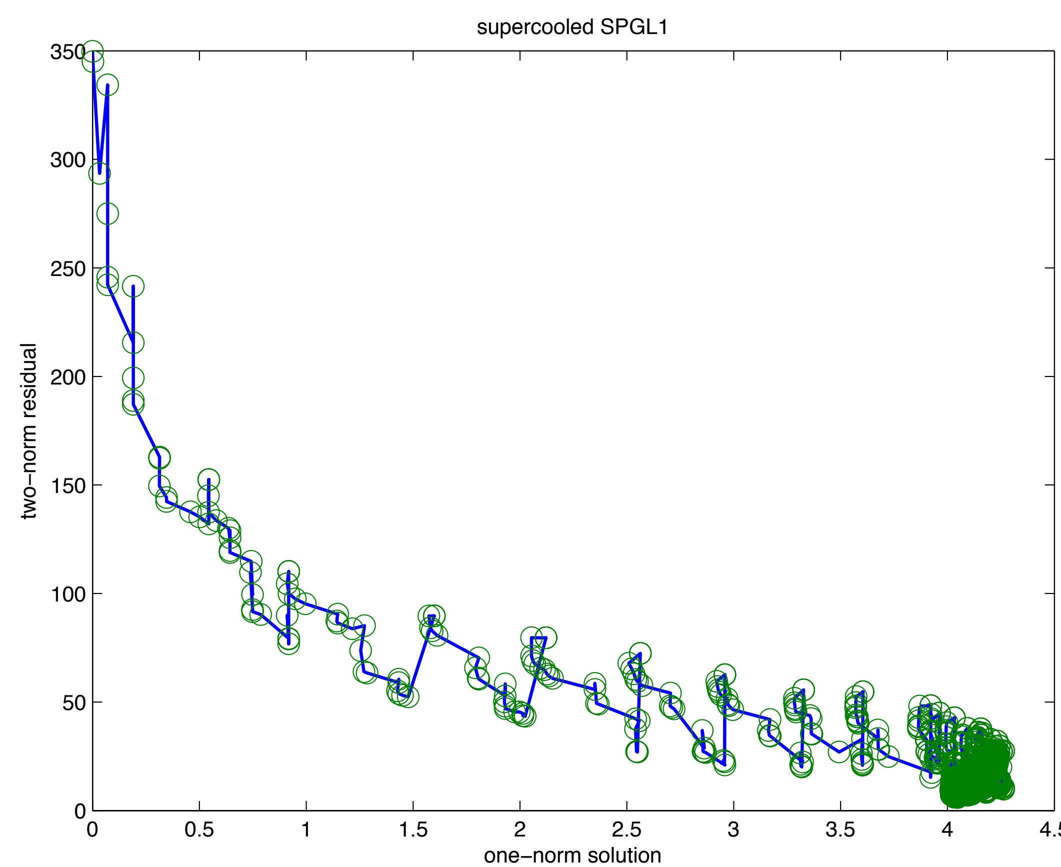
with renewals

Migration results

[solution paths ℓ_1]

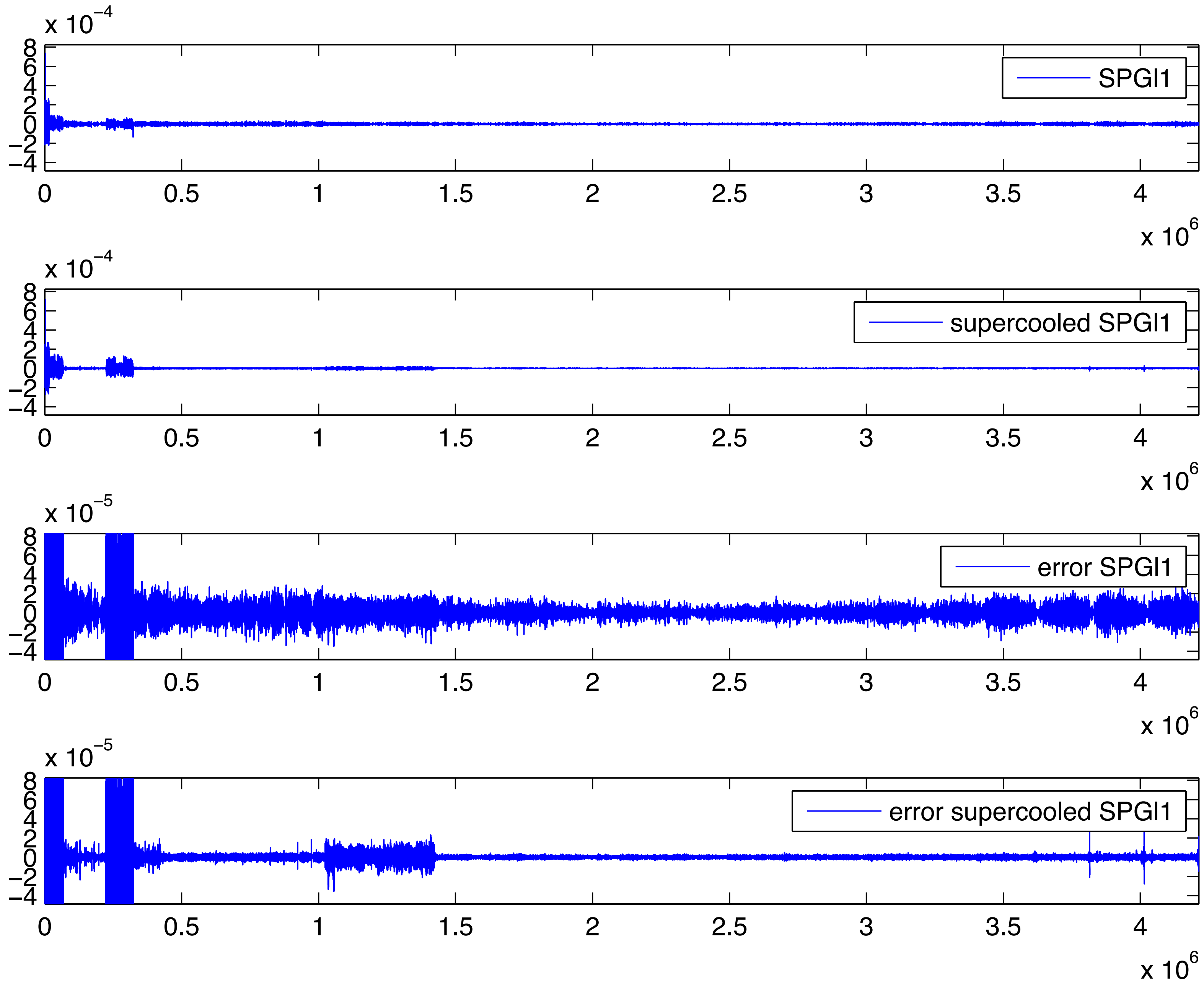


without renewals



with renewals

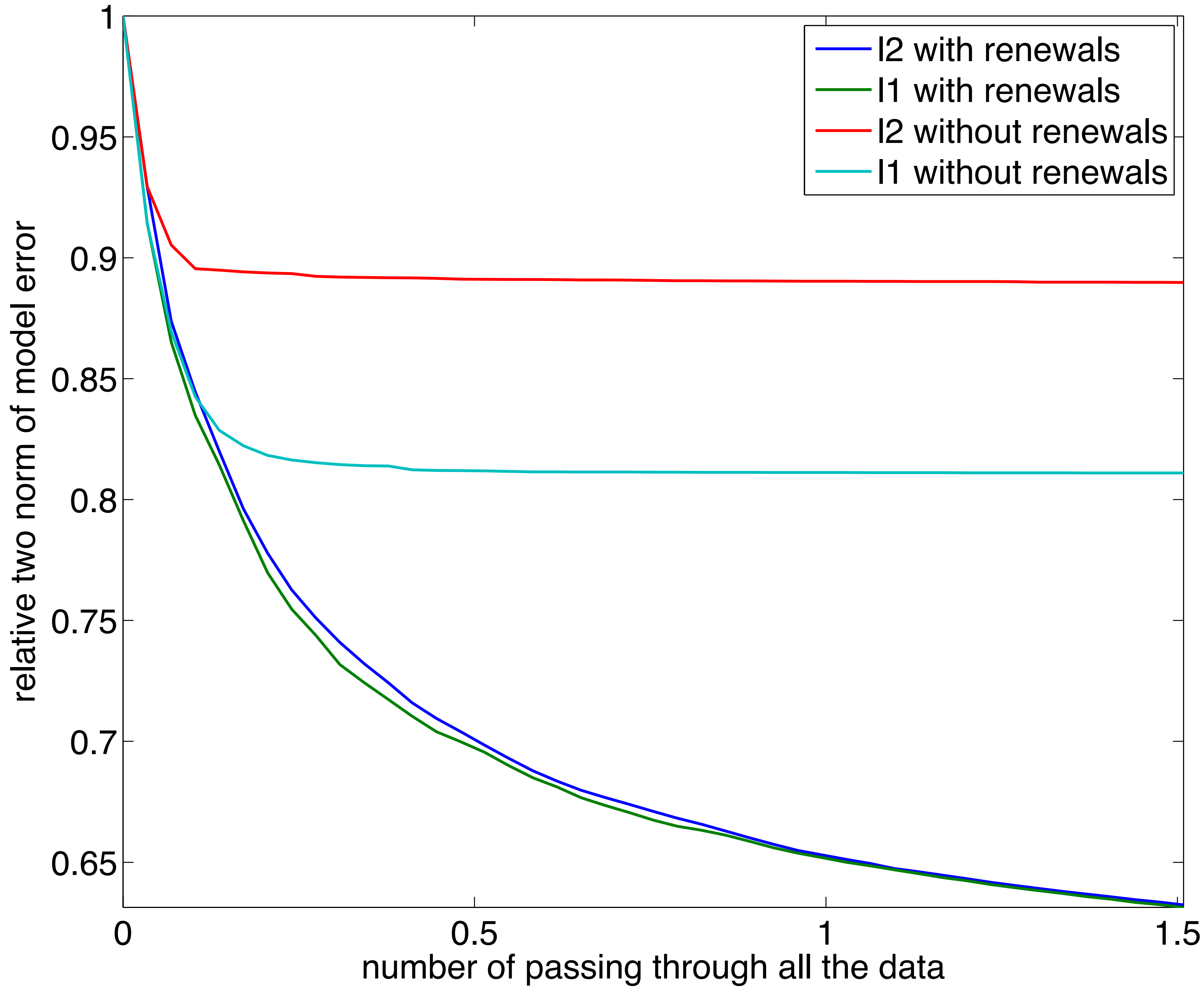
Imaging results



10 X

10 X

Migration results
[model errors]



Why does this work?

Physicist's perspective:

We are dealing with *extremely* large *systems* that *mix* for

- *large* enough system *sizes* and long enough *times*
- *large* enough *complexity* in the *velocity* model

Linear systems start to behave like 'Gaussian' matrices

- show 'phase-transitions' for *simple* recovery *algorithms*
- *approximations* become *better* when systems get *larger*

Approximate message passing

Add a *term* to *iterative* soft *thresholding*, i.e.,

$$\mathbf{x}^{t+1} = \eta_t\left(\mathbf{A}^*\mathbf{r}^t + \mathbf{x}^t\right)$$

$$\mathbf{r}^t = \mathbf{b} - \mathbf{A}\mathbf{x}^t + \frac{\|\mathbf{x}^{t+1}\|_0}{n}\mathbf{r}^{t-1}$$

Holds for

- *normalized* Gaussian matrices $\mathbf{A}_{ij} \in n^{-1/2}N(0,1)$
- large-scale limit and for specific thresholding *strategy*

[Montanari, '12]

SLIM

Approximate message passing

Statistically equivalent to

$$\boxed{\begin{aligned}\mathbf{x}^{t+1} &= \eta_t\left(\mathbf{A}_t^*\mathbf{r}^t + \mathbf{x}^t\right)\\ \mathbf{r}^t &= \mathbf{b}_t - \mathbf{A}_t\mathbf{x}^t\end{aligned}}$$

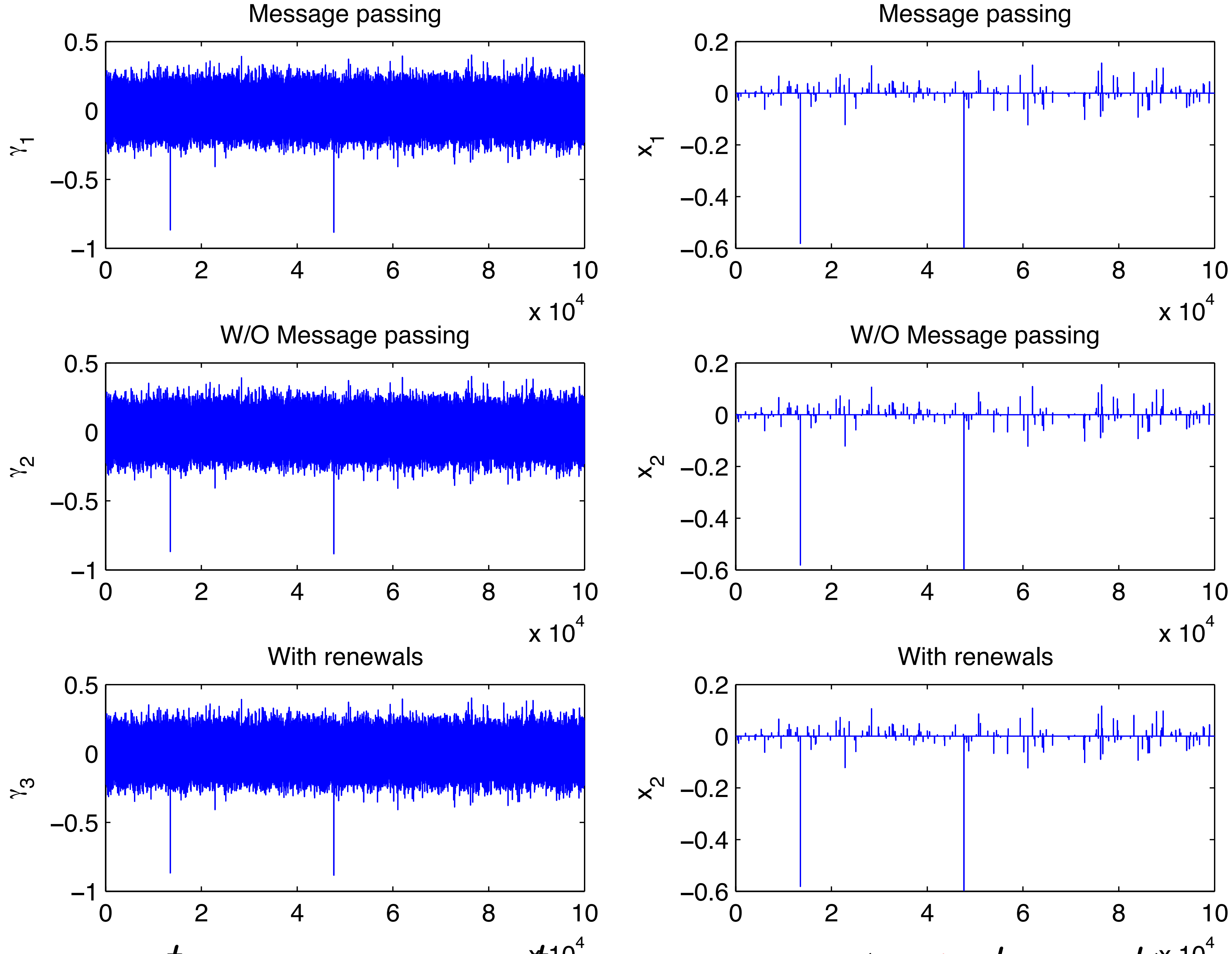
by drawing *new independent* pairs $\{\mathbf{b}_t, \mathbf{A}_t\}$ for *each* iteration

Changes the story *completely*

- breaks *correlation* buildup
- *faster* convergence

Iteration t=1

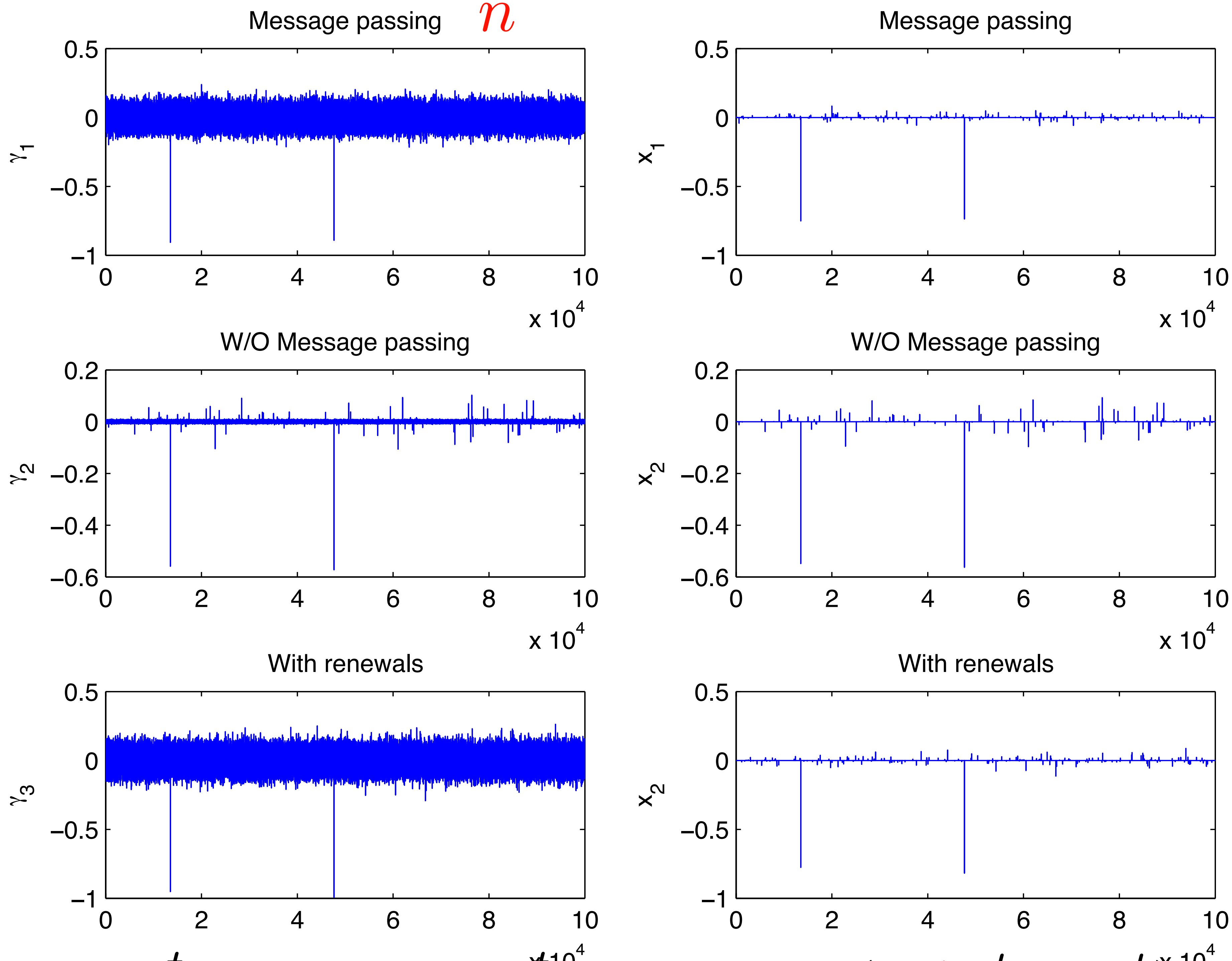
$$\mathbf{r}^t = \mathbf{b} - \mathbf{A}\mathbf{x}^t + \frac{\|\mathbf{x}^{t+1}\|_0}{} \mathbf{r}^{t-1} \qquad \eta_t(\mathbf{A}^*\mathbf{r}^t + \mathbf{x}^t)$$



$$\mathbf{r}^t = \mathbf{b}_t - \mathbf{A}_t\mathbf{x}^t \qquad \eta_t(\mathbf{A}_t^*\mathbf{r}^t + \mathbf{x}^t)$$

Iteration t=2

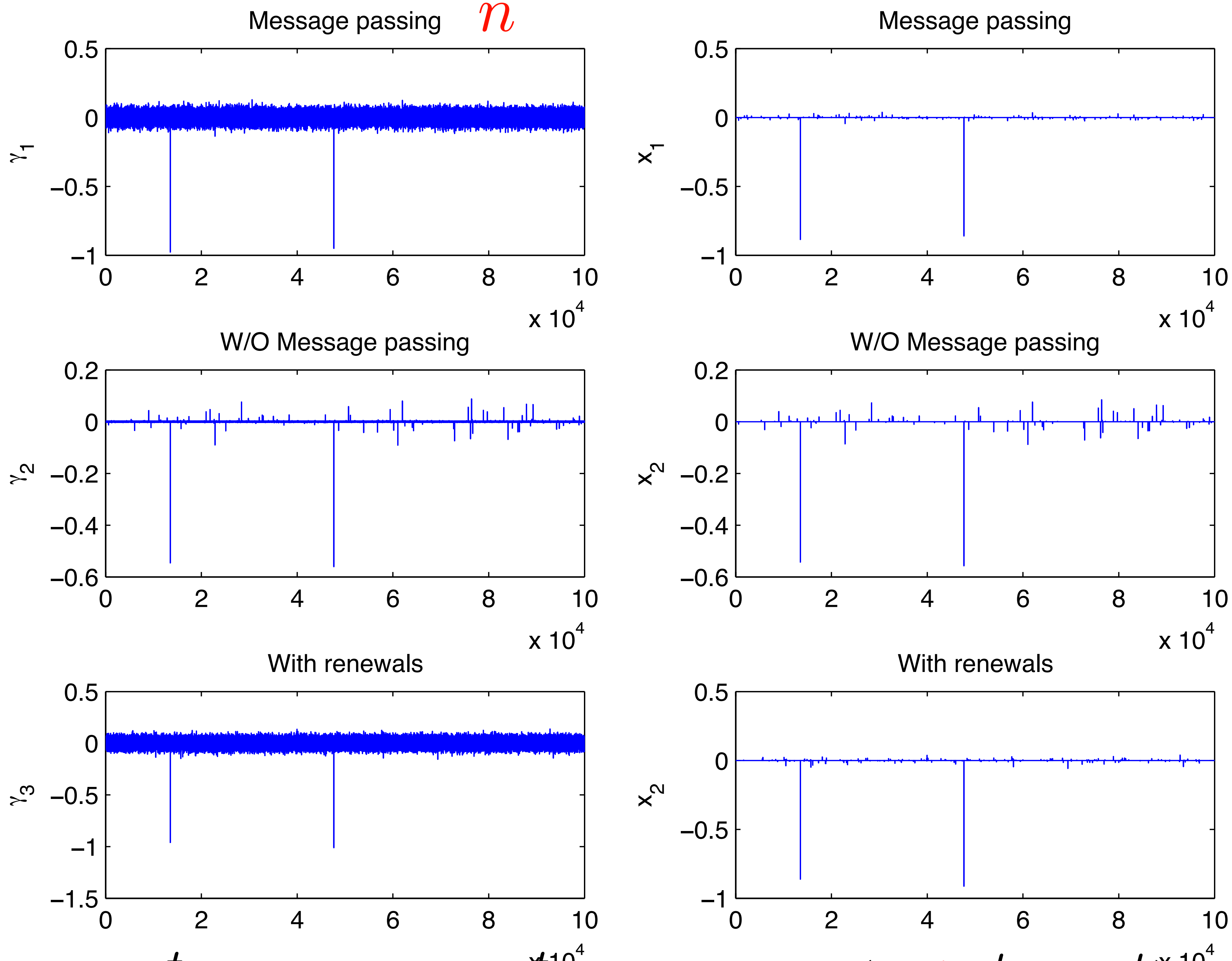
$$\mathbf{r}^t = \mathbf{b} - \mathbf{A}\mathbf{x}^t + \frac{\|\mathbf{x}^{t+1}\|_0}{n}\mathbf{r}^{t-1} \qquad \eta_t(\mathbf{A}^*\mathbf{r}^t + \mathbf{x}^t)$$



$$\mathbf{r}^t = \mathbf{b}_t - \mathbf{A}_t\mathbf{x}^t \qquad \eta_t(\mathbf{A}_t^*\mathbf{r}^t + \mathbf{x}^t)$$

Iteration t=3

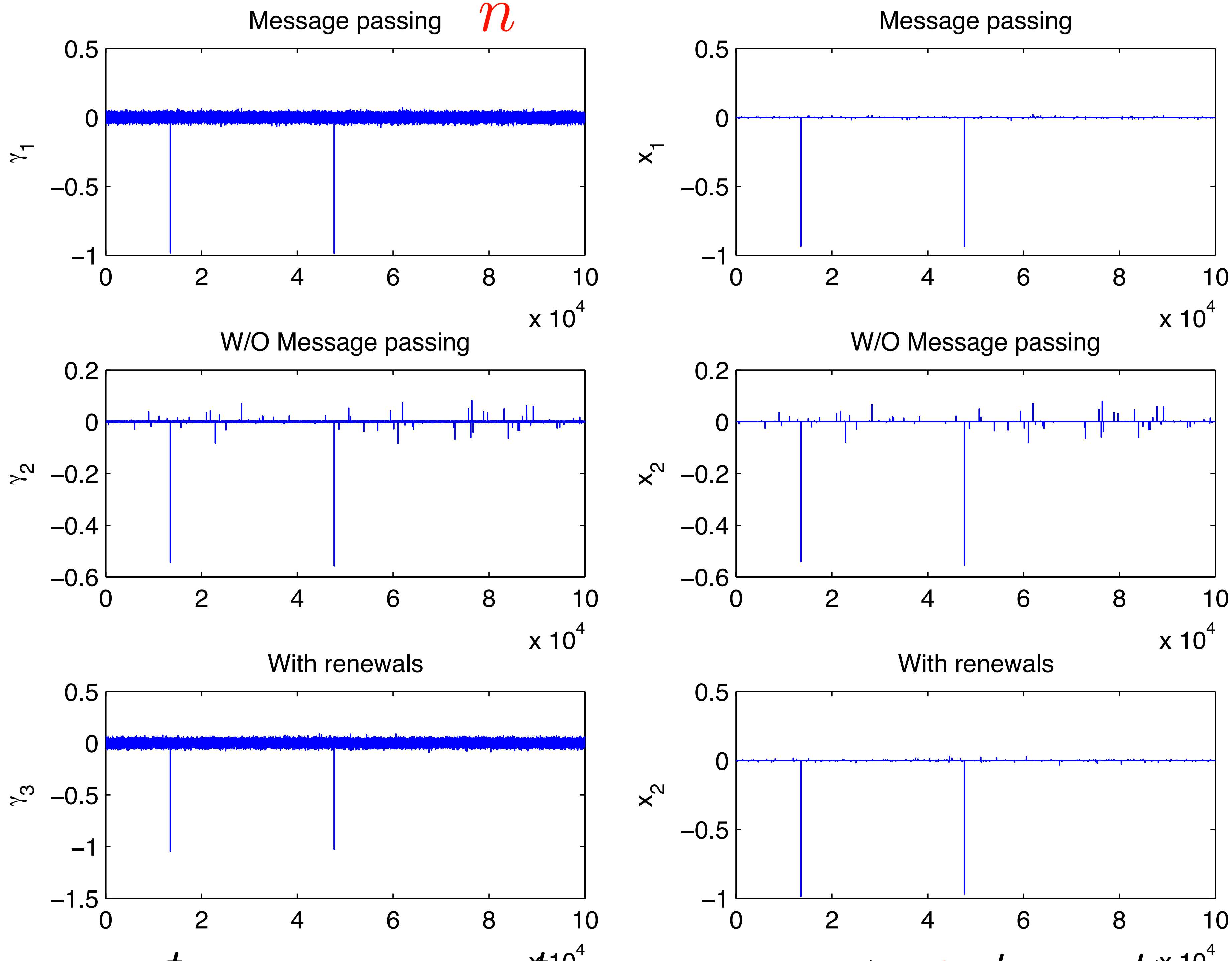
$$\mathbf{r}^t = \mathbf{b} - \mathbf{A}\mathbf{x}^t + \frac{\|\mathbf{x}^{t+1}\|_0}{n}\mathbf{r}^{t-1} \qquad \eta_t(\mathbf{A}^*\mathbf{r}^t + \mathbf{x}^t)$$



$$\mathbf{r}^t = \mathbf{b}_t - \mathbf{A}_t\mathbf{x}^t \qquad \eta_t(\mathbf{A}_t^*\mathbf{r}^t + \mathbf{x}^t)$$

Iteration t=4

$$\mathbf{r}^t = \mathbf{b} - \mathbf{A}\mathbf{x}^t + \frac{\|\mathbf{x}^{t+1}\|_0}{n}\mathbf{r}^{t-1} \qquad \eta_t(\mathbf{A}^*\mathbf{r}^t + \mathbf{x}^t)$$



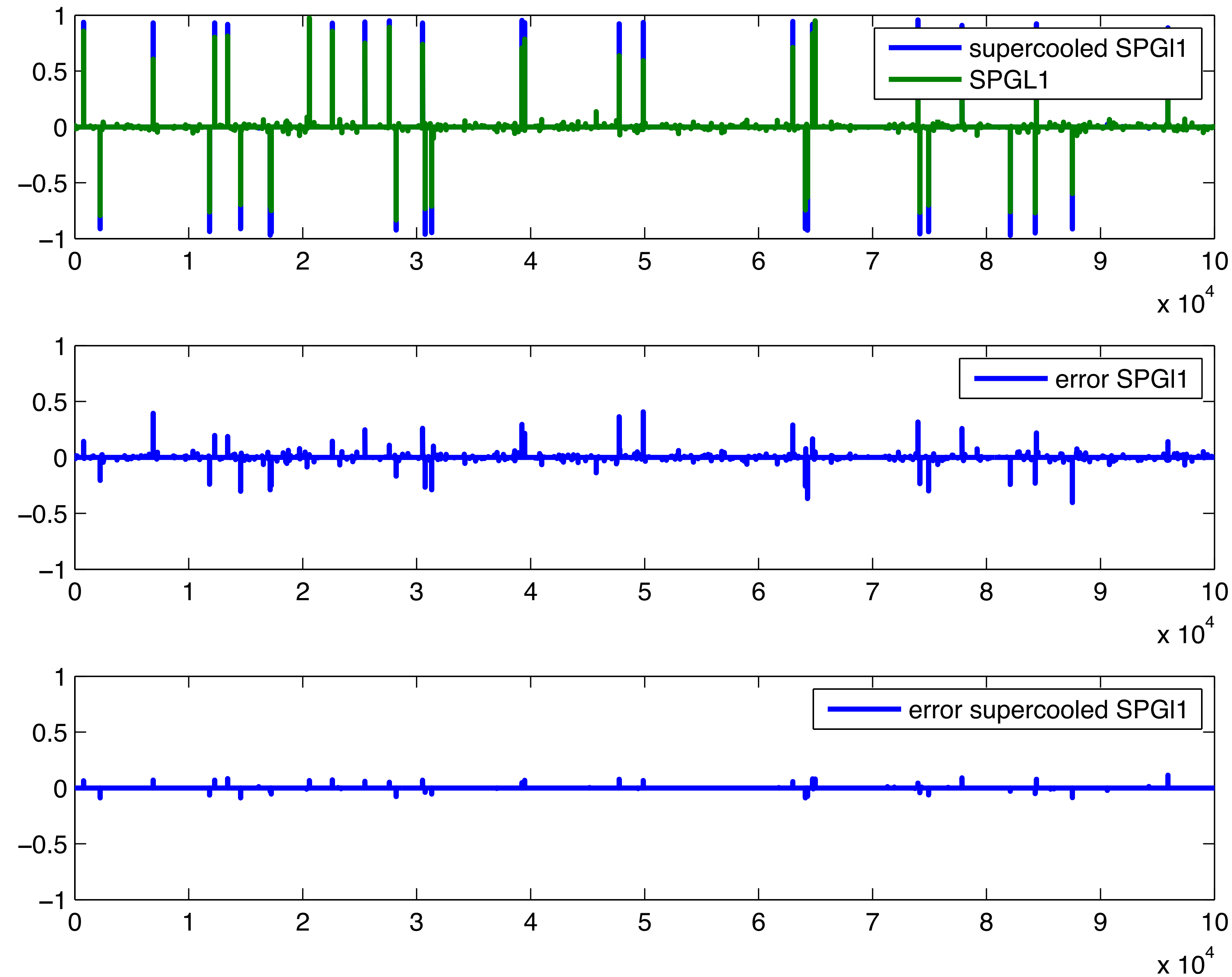
$$\mathbf{r}^t = \mathbf{b}_t - \mathbf{A}_t\mathbf{x}^t \qquad \eta_t(\mathbf{A}_t^*\mathbf{r}^t + \mathbf{x}^t)$$

Observations

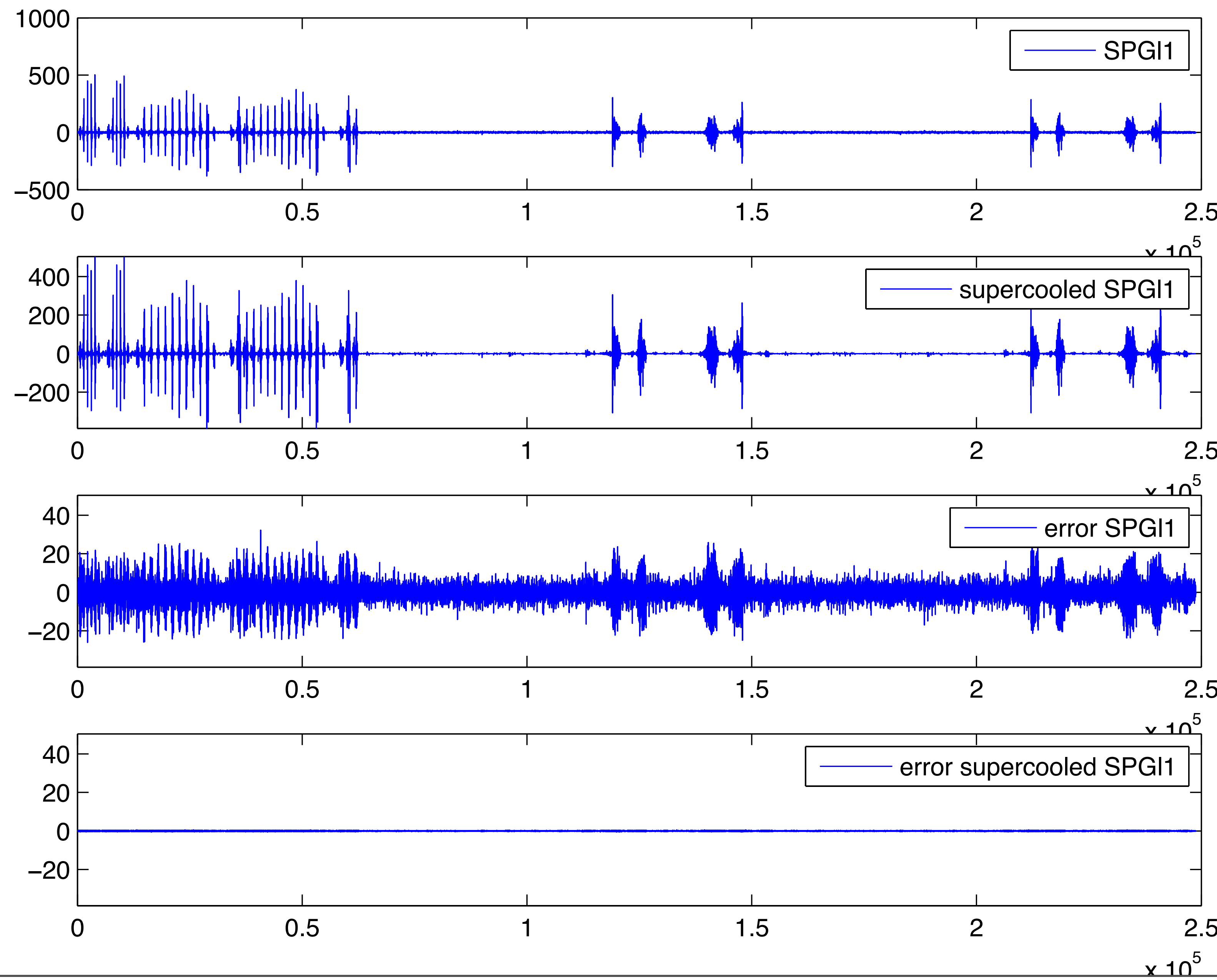
Message-pass term has the same *effect* as drawing *independent* experiments $\{\mathbf{b}_t, \mathbf{A}_t\}$

- '*Gaussian*' matrices
- *delicate* normalization and *thresholding* strategy
- *renders* proposed method *impractical*
- can lead to *dramatically* improved convergence

Sparse example
[n=500; N=10000; k=35; T=50]



Ideal 'Seismic' example [n/N=0.13;N=248759;T=500]

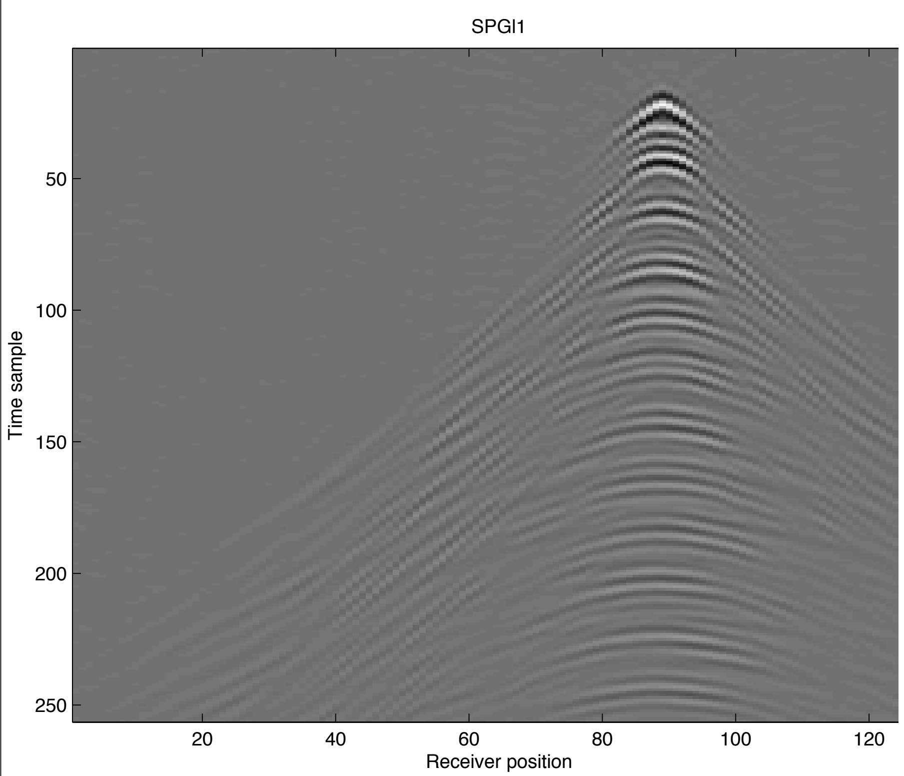


10 X

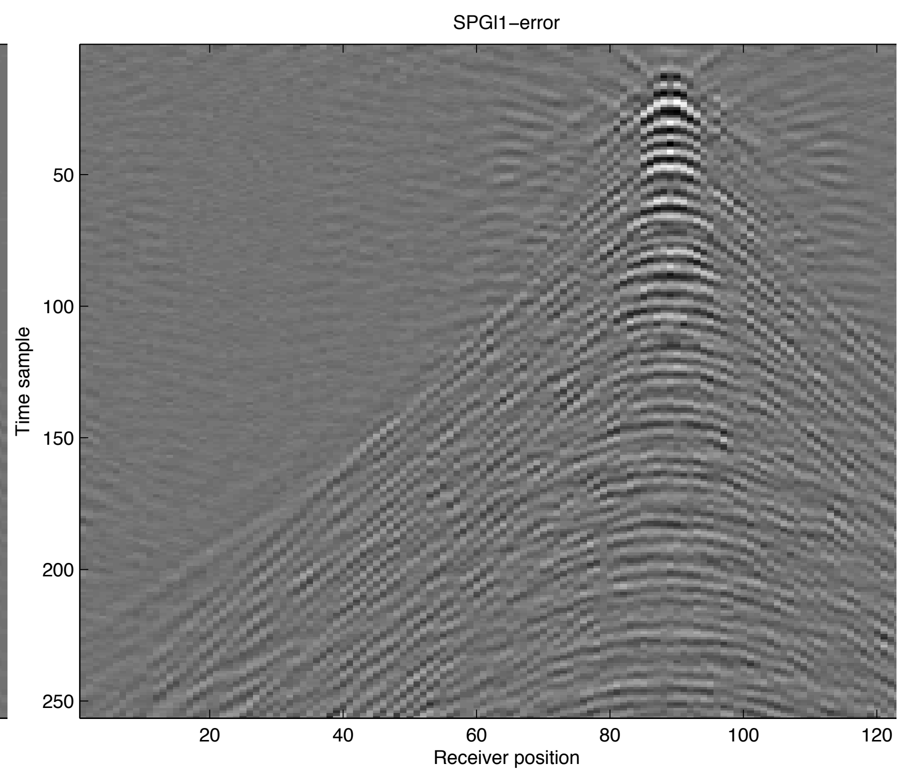
10 X

Ideal ‘Seismic’ example [n/N=0.13;N=248759;T=500]

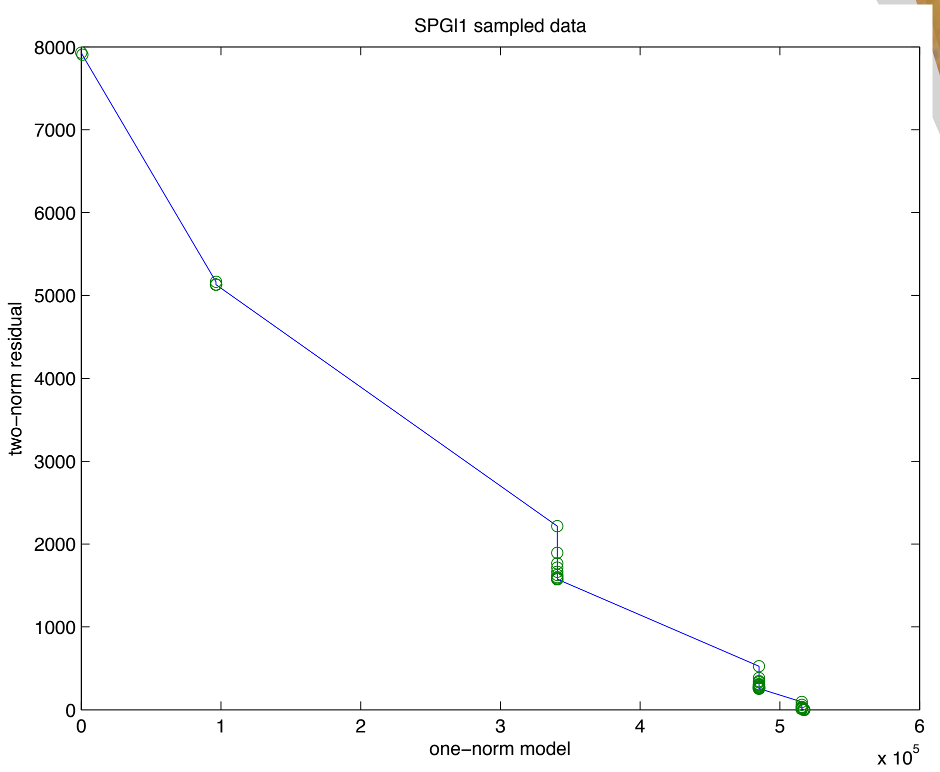
10 X



recovery



error

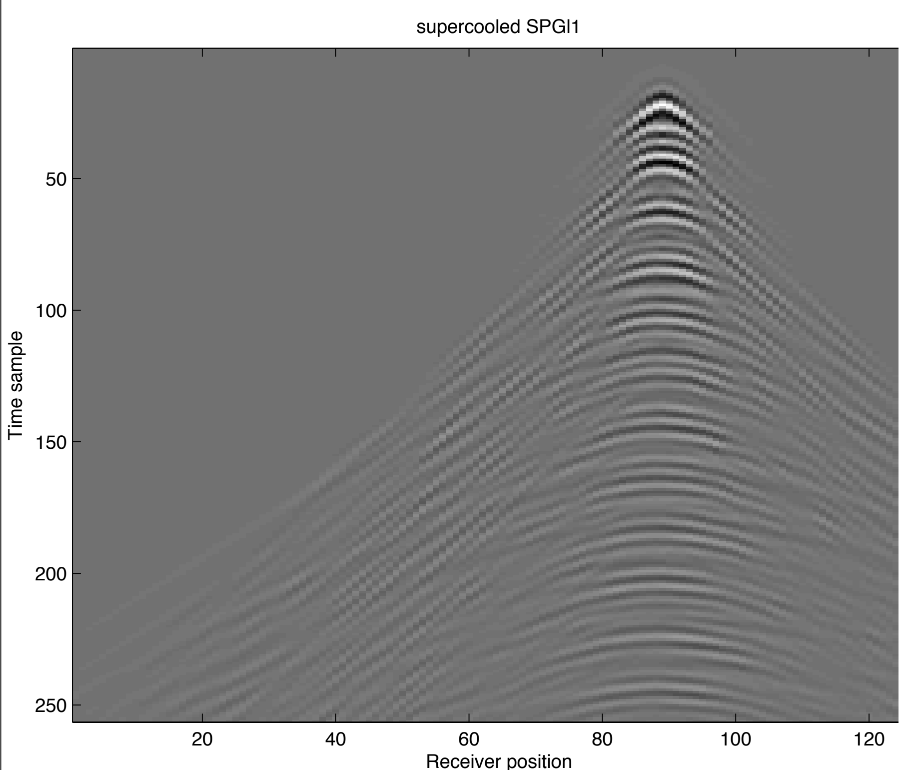


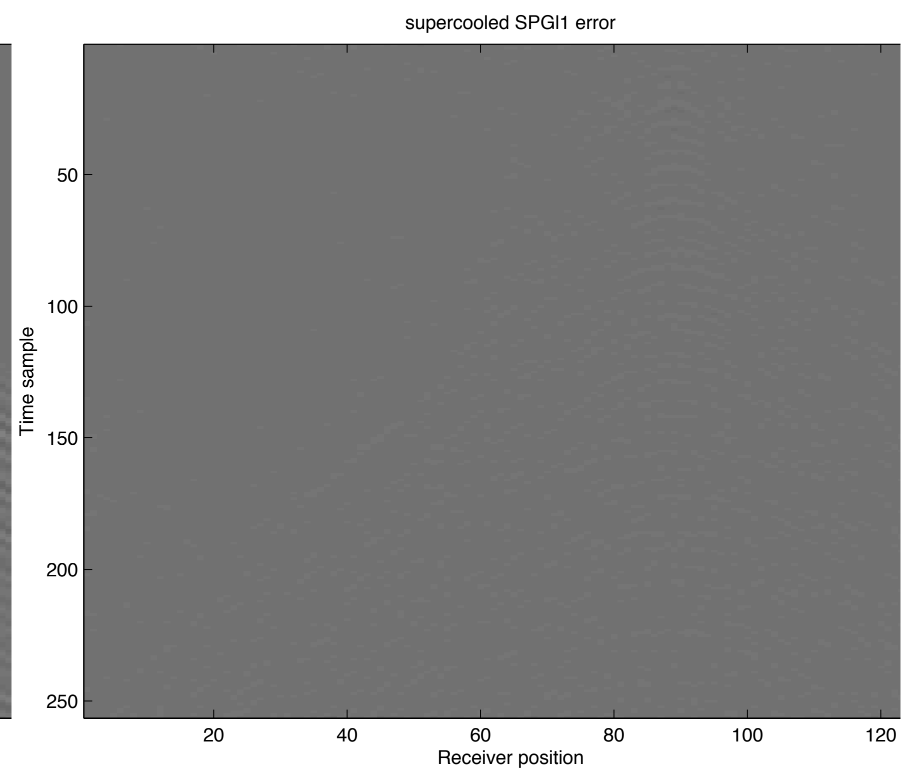
solution *path*

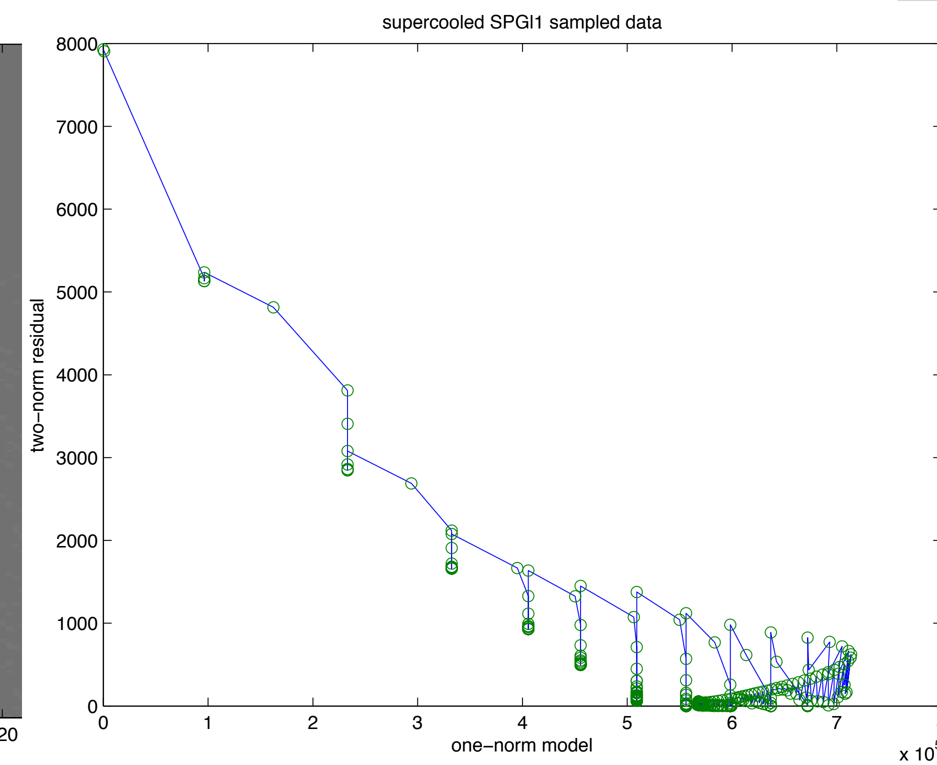
Cooled

Ideal 'Seismic' example [n/N=0.13;N=248759;T=500]

10 X



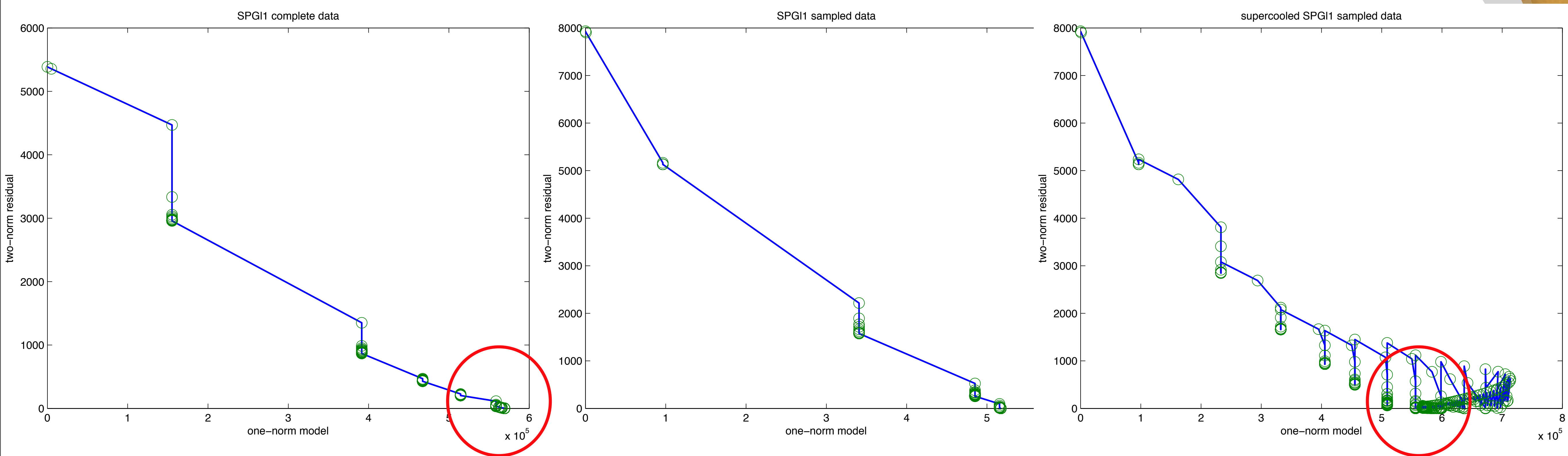




recovery *error* solution *path*

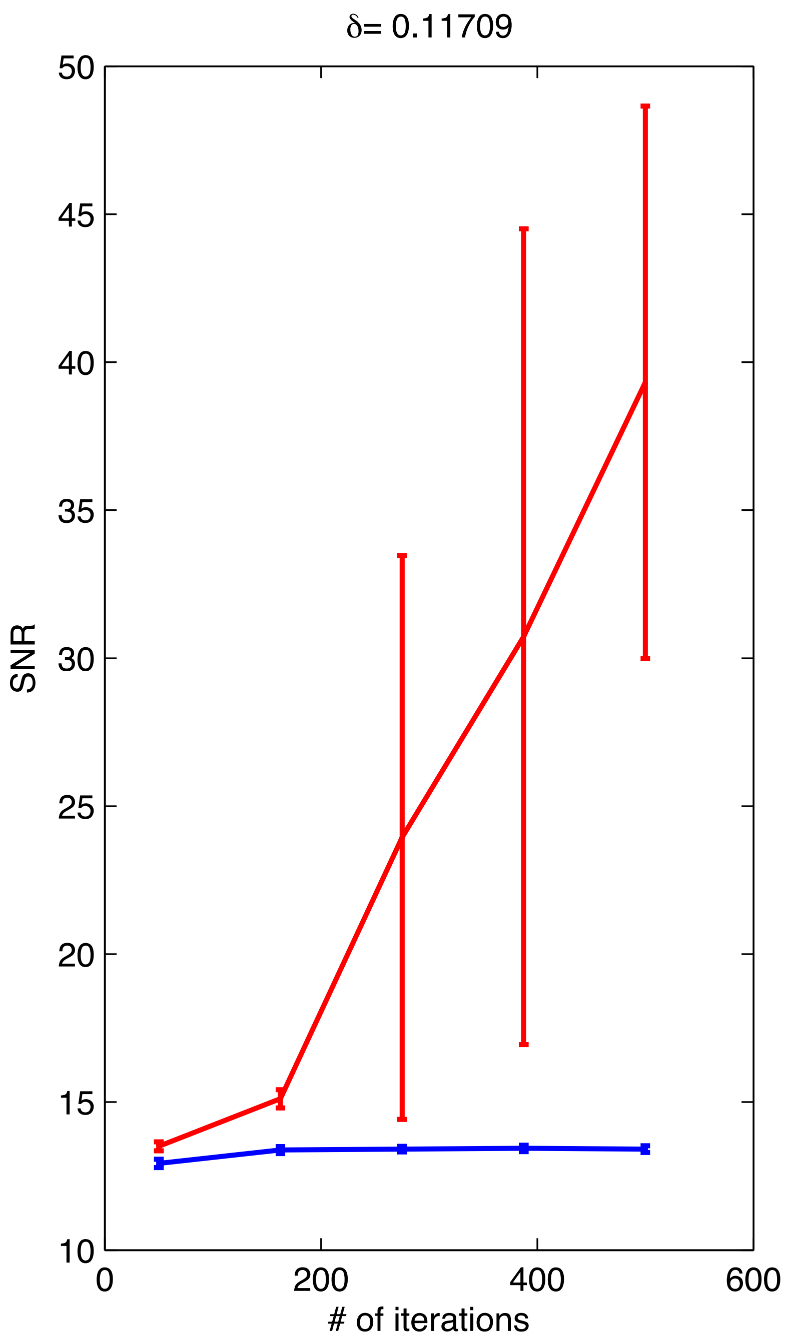
Supercooled

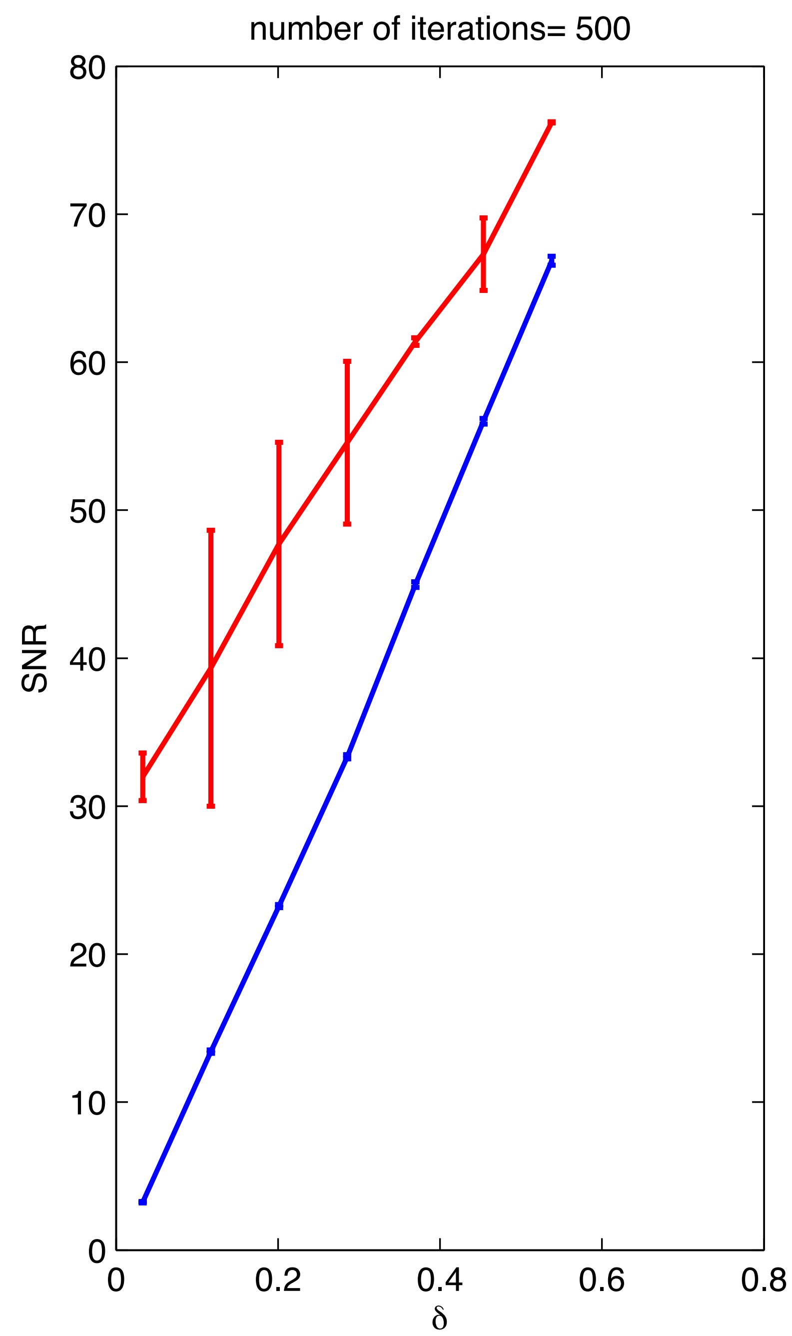
Solution paths



Independent redraws of $\{\mathbf{b}_t, \mathbf{A}_t\}$ lead to improved *recovery*

MCC experiments





[Romero et. al., 2000;]

[Montanari, 2012]

[Herrmann & Li, 2012]

Observations

Independent redraws of $\{\mathbf{b}_t, \mathbf{A}_t\}$ get rid of *small* difficult to remove *interferences*

- working *only* with *subsets* of the *data*

But, aren't we *fooling* ourselves since *proposed* method

- *defeats* the *premise* of *compressive* sampling

Or, are there *data-rich* applications for this method?

- e.g. *efficient* imaging with *random* source *encoding*

Conclusions

Message passing improves *image* quality

- computationally *feasible* one-norm *regularization*

Message *passing* via *rerandomization*

- *small* system *size* with *small* IO and memory *imprints*

Possibility to exploit *new* computer *architectures* that employ *model* space *parallelism* to *speed up* wavefield *simulations*...

SLIM

FWI results

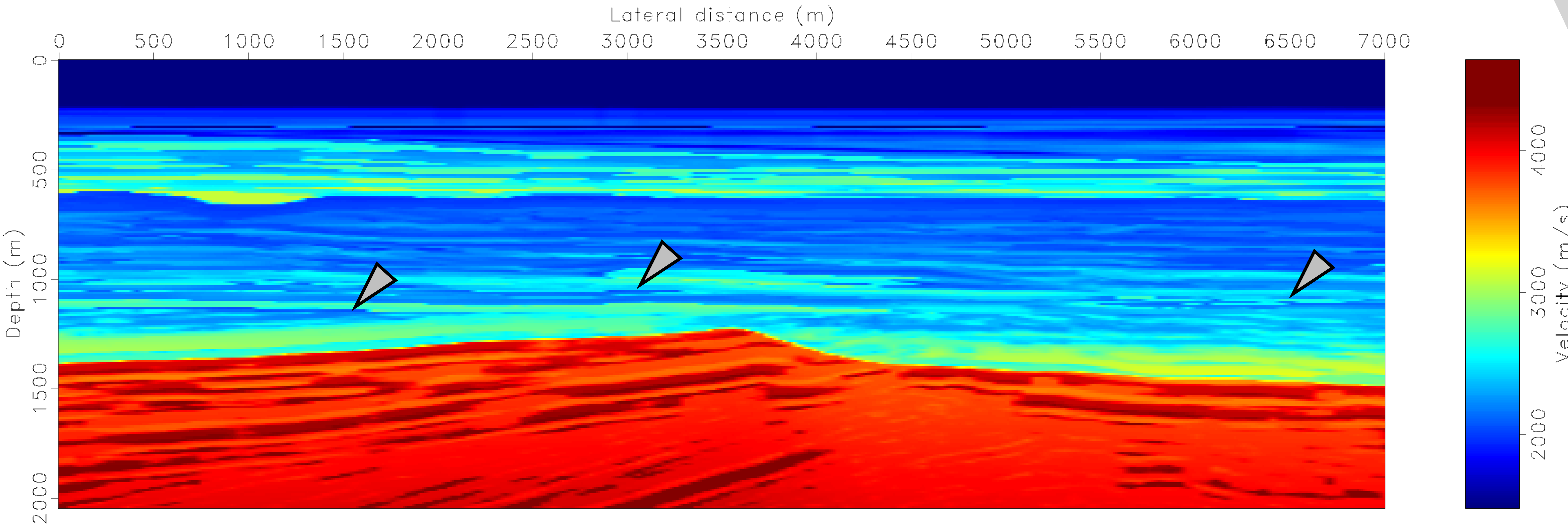
FWI:

- 10 overlapping frequency bands with 10 frequencies (2.9Hz-25Hz)
- 10 Gauss-Newton steps for each frequency band (solved with max 20 spectral-projected gradient iterations)

[Bunks '95; Pratt '98]

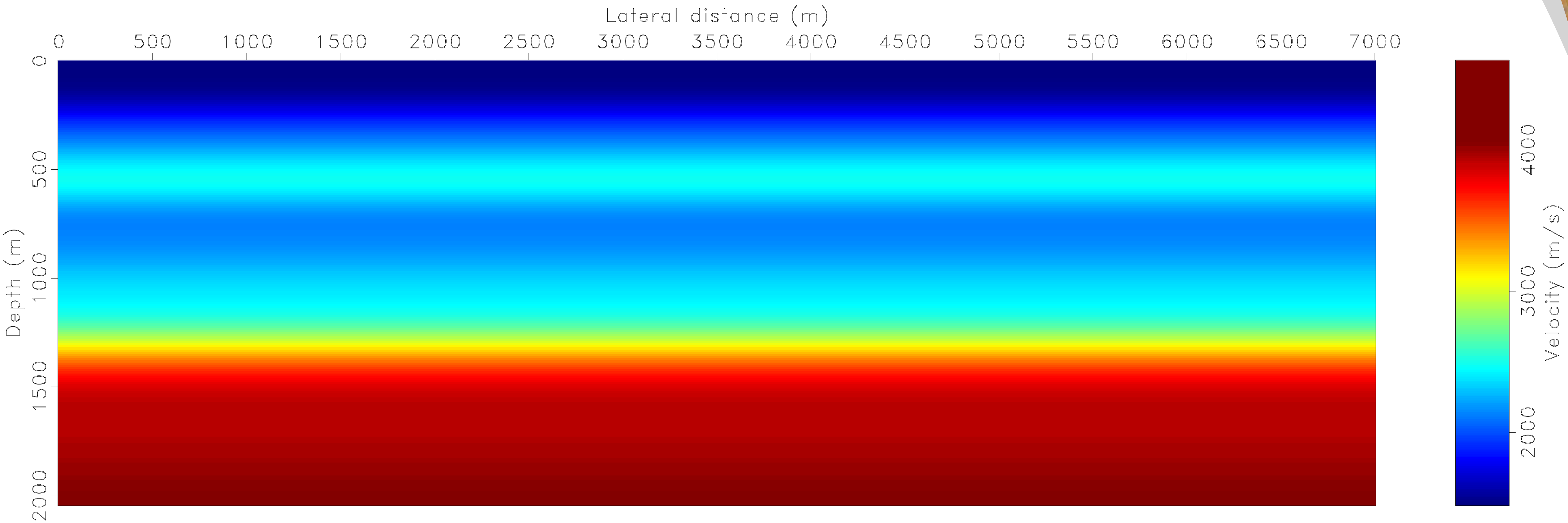
Results GN-FWI

True model



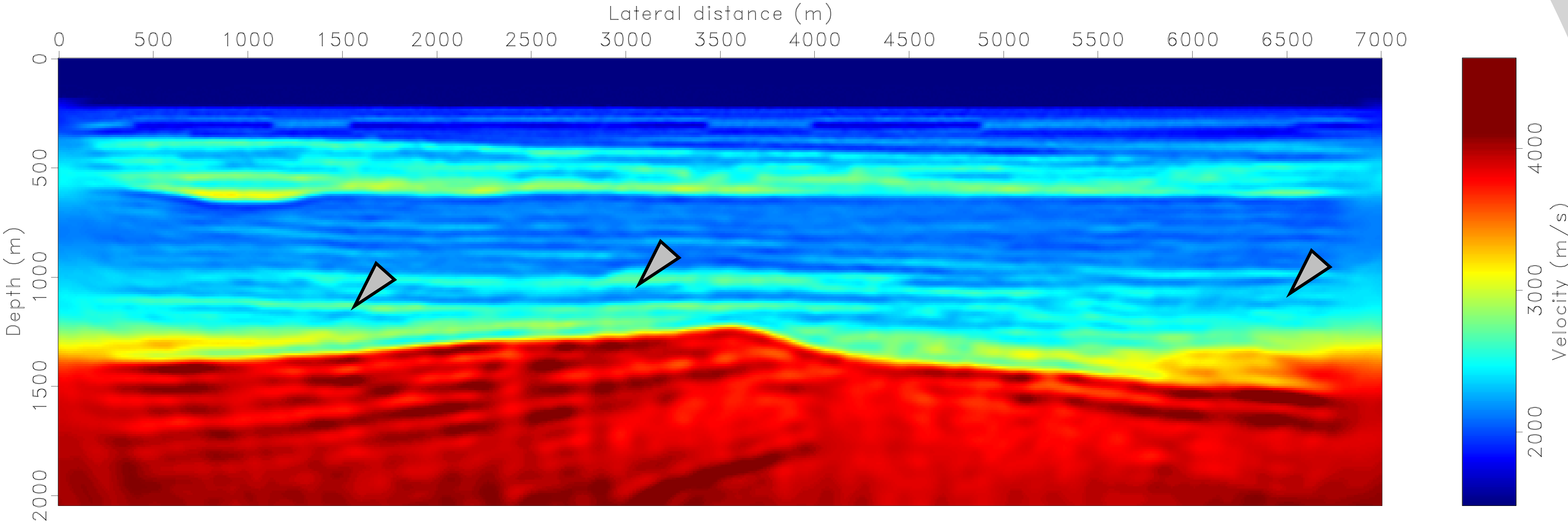
Results GN-FWI

Initial model



Results GN-FWI

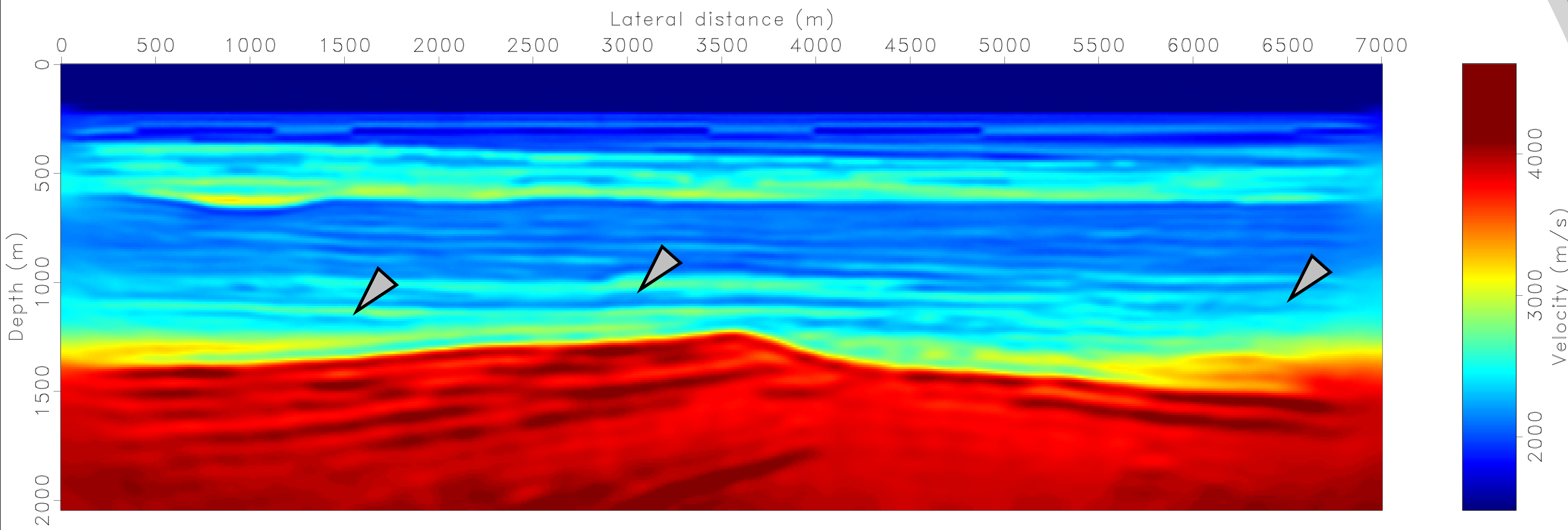
Modified GN 7 sim. shots *without renewals*



25 times speedup compared to full GN

Results GN-FWI

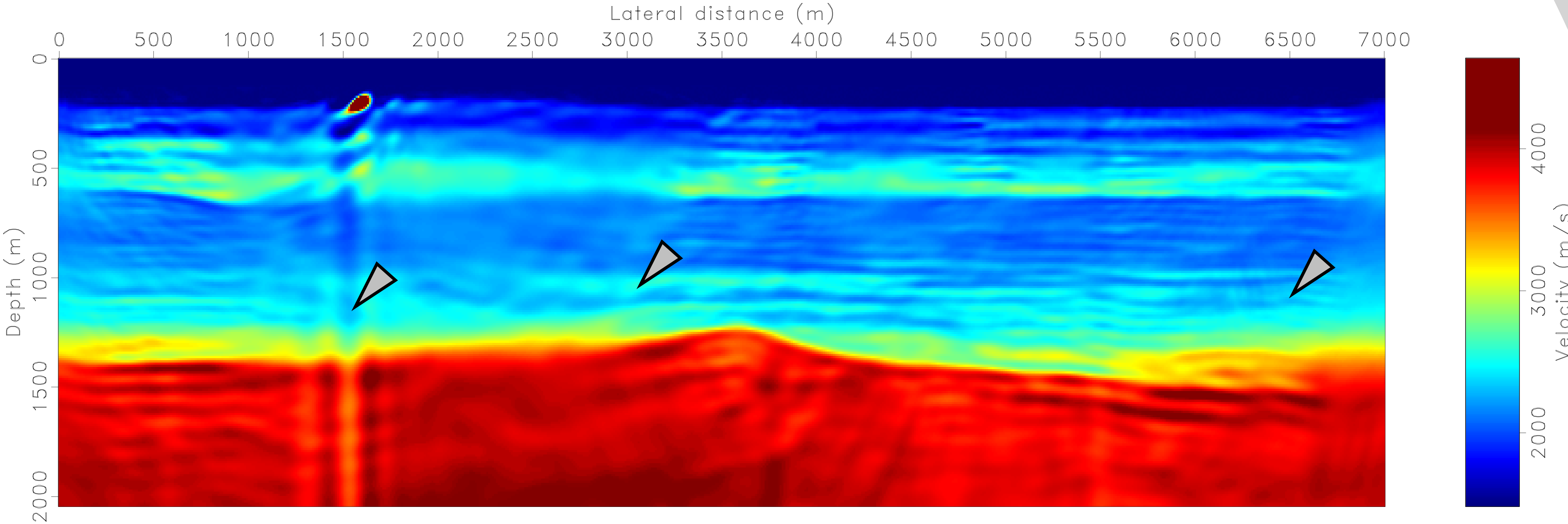
Modified GN 7 sim. shots *with renewals*



25 times speedup compared to full GN

Results GN-FWI

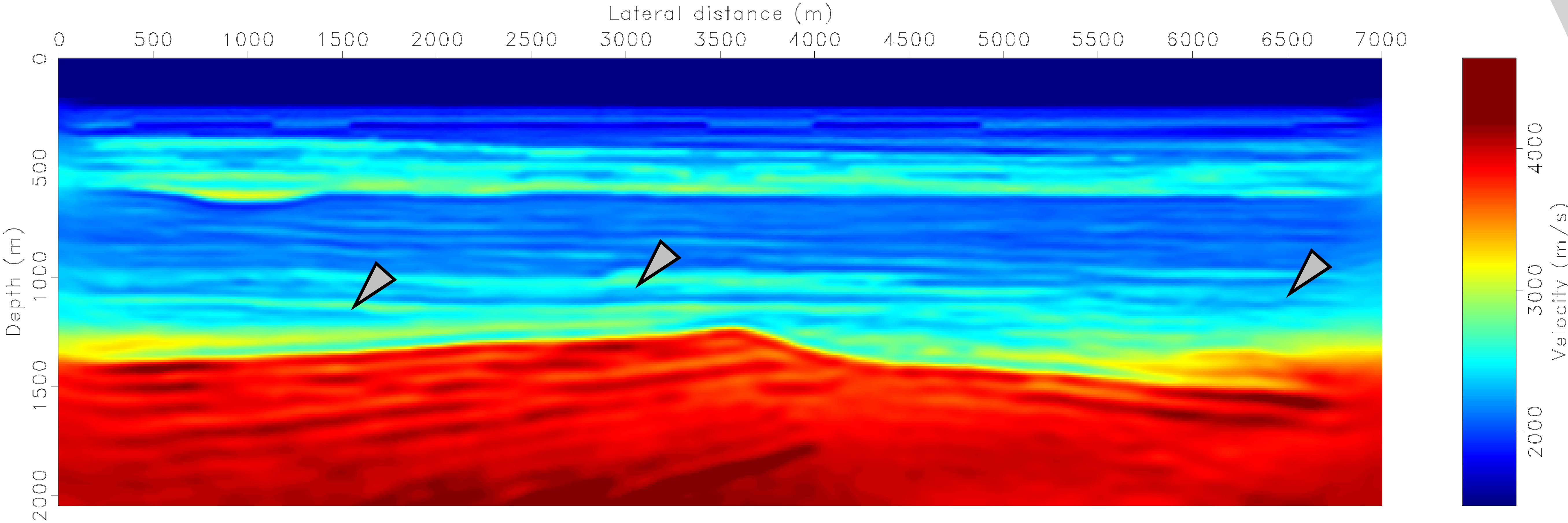
Modified GN 7 seq. shots *without renewals*



25 times speedup compared to full GN

Results GN-FWI

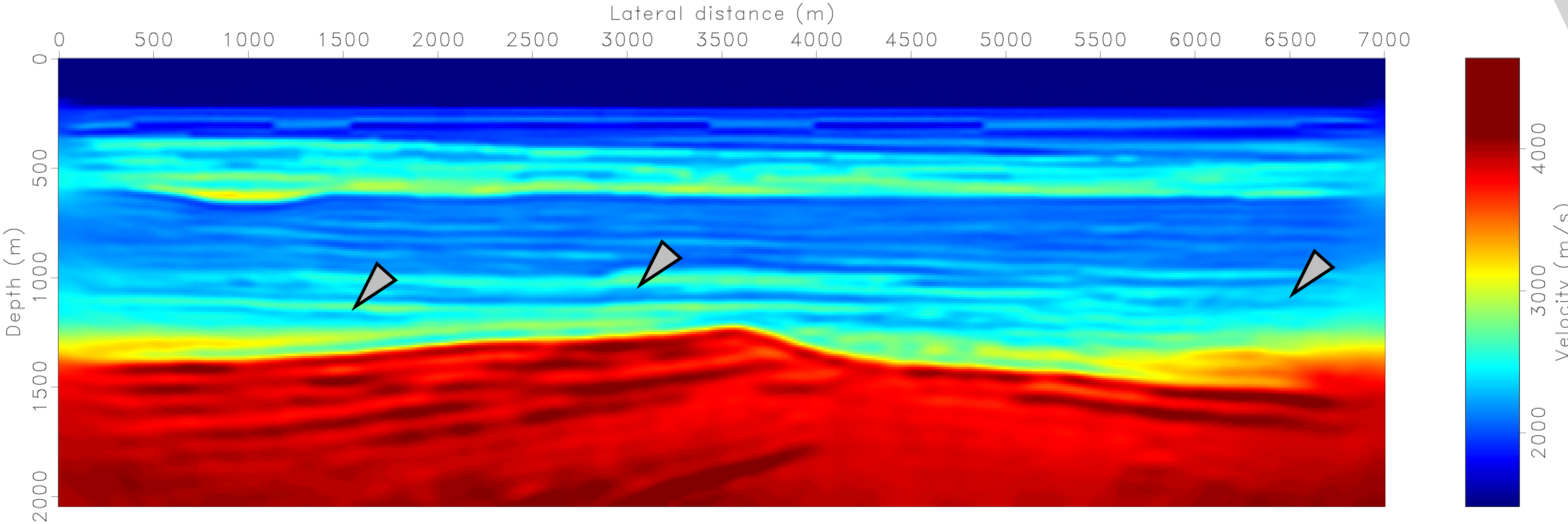
Modified GN 7 seq. shots *with renewals*



25 times speedup compared to full GN

Results GN-FWI

Modified GN 7 sim. shots *with renewals*



25 times speedup compared to full GN