SINBAD's research program

Felix J. Herrmann & Tristan van Leeuwen

SLIM

Seismic Laboratory for Imaging and Modeling
the University of British Columbia

What do we do at SLIM...?

Main research areas

Randomized seismic acquisition design

- *improved* quality at *reduced* cost
- fundamental *new* insights in (simultaneous) acquisition

Robust & dimensionality-reduced full-waveform inversion

- *removal* of computational *burden* & memory *imprint*
- *high-quality* inversions from *randomized* subsets of data

SLIM

Main research areas

Sparsity inducing imaging *with* surface-related multiples

- *improved* image quality by *leveraging*
  - relation *between* primaries and multiples
  - additional *sparsity* in the *image* domain
- efficiency via *randomized* dimensionality reduction

Key technologies

Stochastic optimization & compressive sensing

- sim. source acquisition, phase encoding, randomized batching etc.

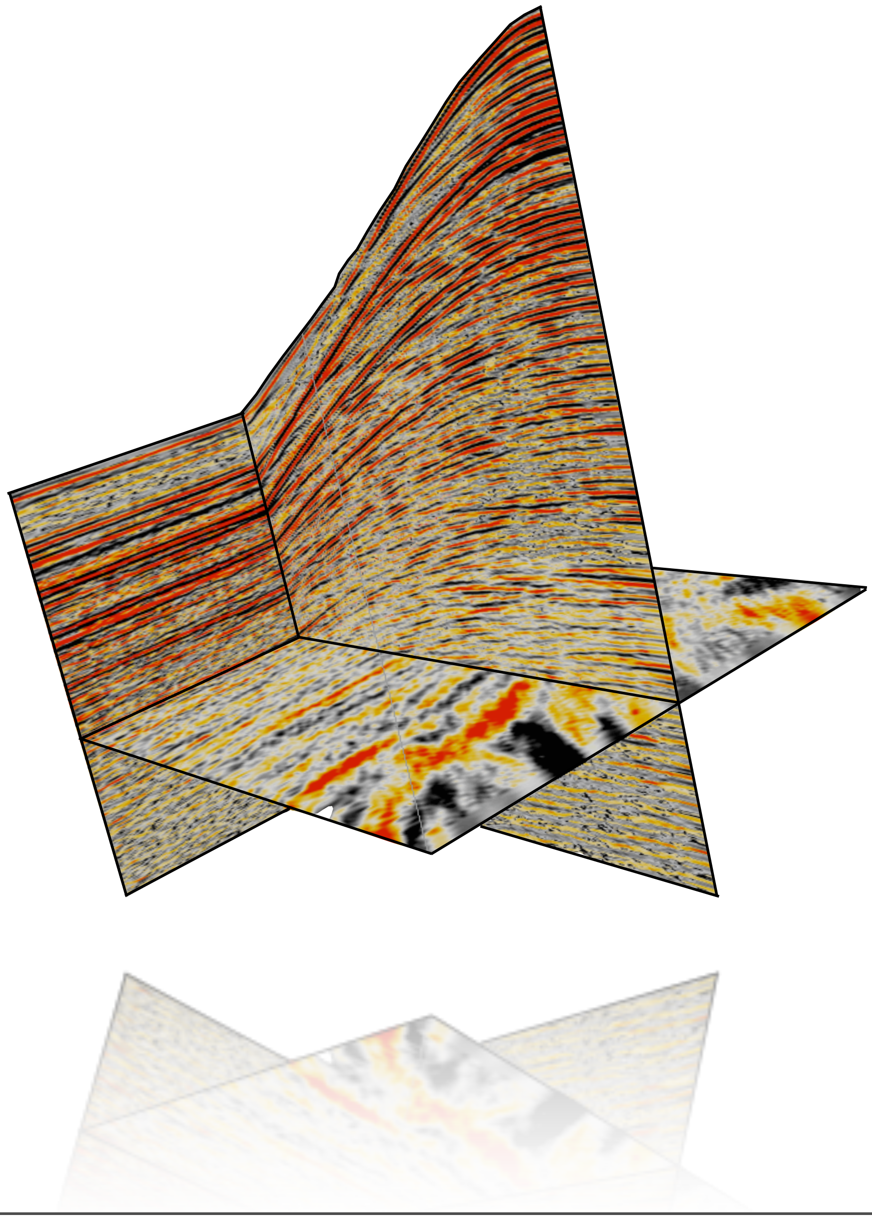
Sparsifying transforms

- wavelets, curvelets etc.

Large-scale (convex) optimization & robust statistics

- one-norm minimization, semi-stochastic optimization, student t minimization etc.

Randomized seismic acquisition design



SLIM

Key goals

Efficient & *high-quality* acquisition

- *more* information from *fewer* data by *adapting* insights from compressive sensing

SLIM

Key strategy

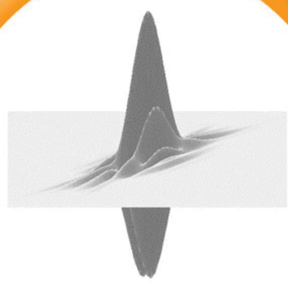
Randomization of acquisition

- *randomized* source/receiver *locations*
- *randomized* time *shifts* in marine
- phase encoding on land or computer

Turn coherent interferences (aliases & source crosstalk) in Gaussian "noise"

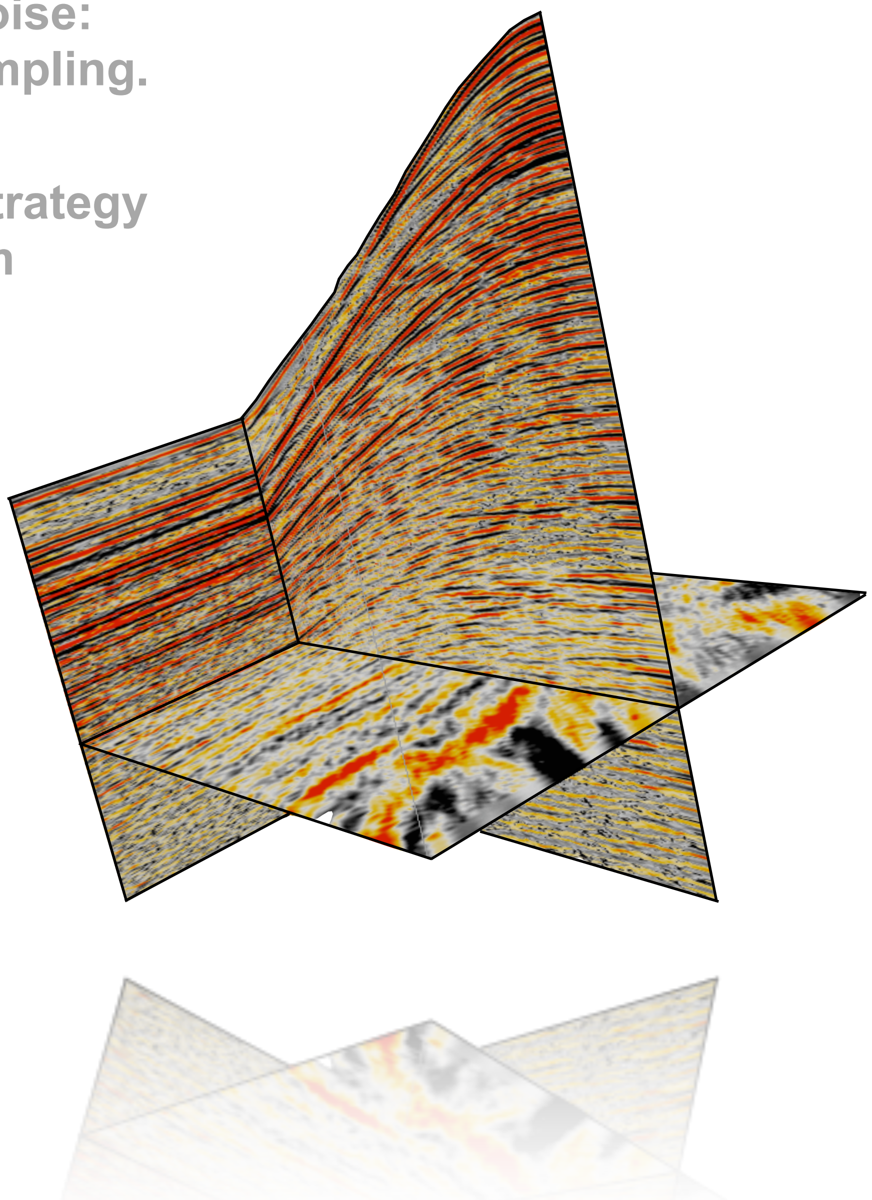
Use transform-domain (e.g. curvelets) *sparsity* promotion to *remove* the *noise*...

Randomized coil sampling

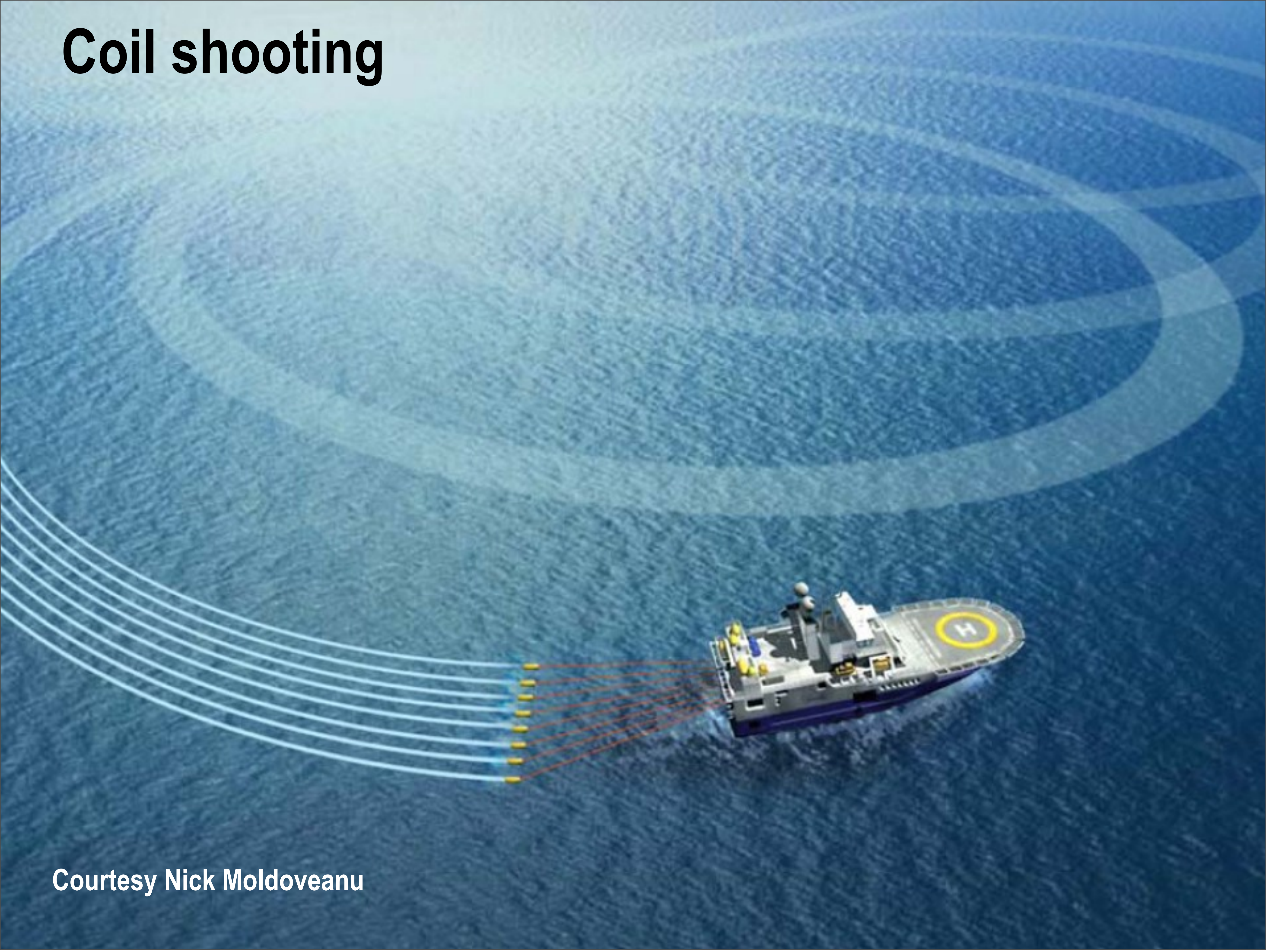


Hennefent, G. and Herrmann, F. J. Simply denoise: wavefield reconstruction via jittered under-sampling. Geophysics, Vol. 73, No. 3, pp. V19–V28, 2008.

Nick Moldoveanu. Random sampling: A new strategy for marine acquisition. SEG Technical Program Expanded Abstracts, 29(1):51–55, 2010.



Coil shooting



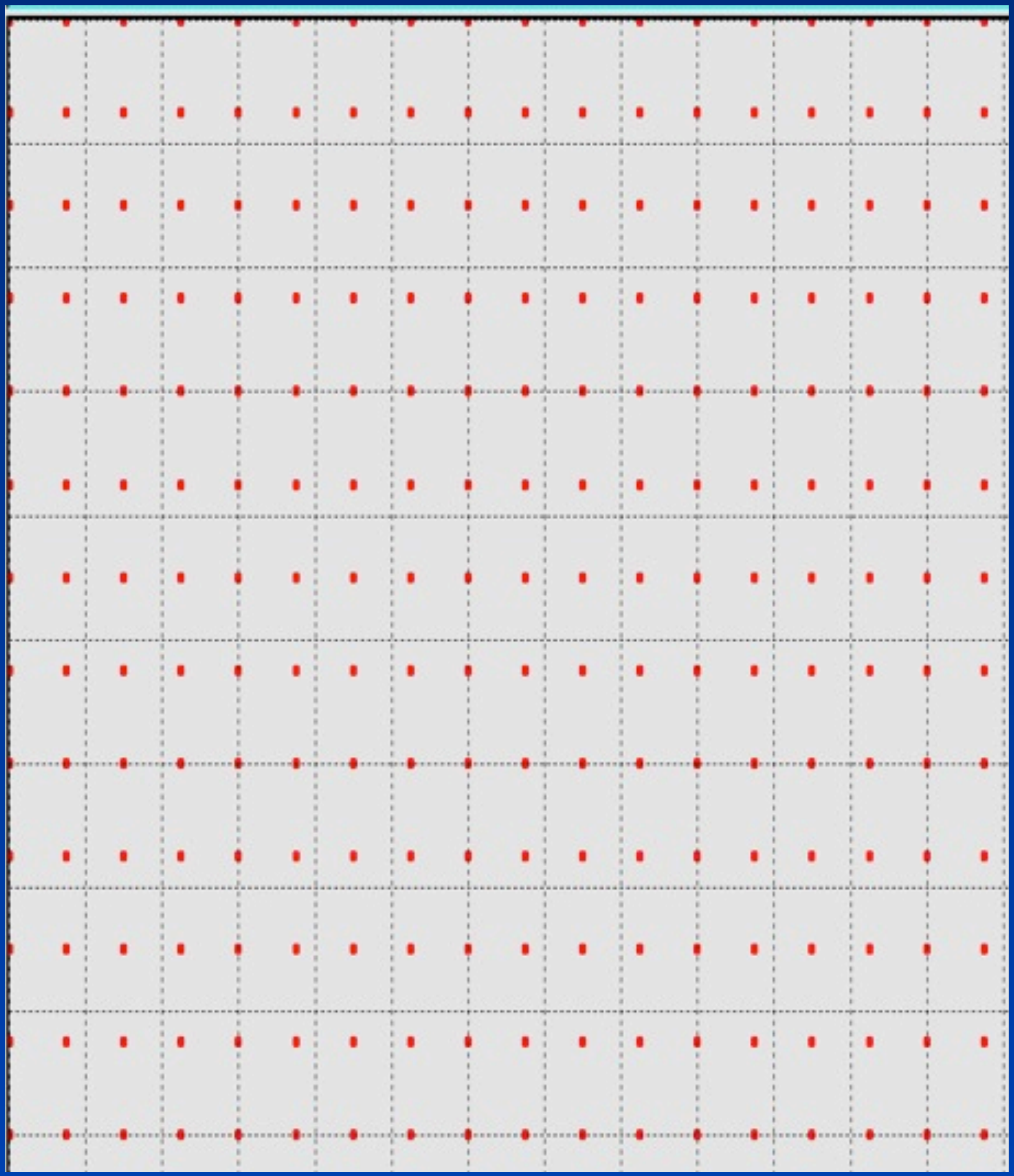
Courtesy Nick Moldoveanu

Coil center grid design

Courtesy Nick Moldoveanu

Coil center grid design

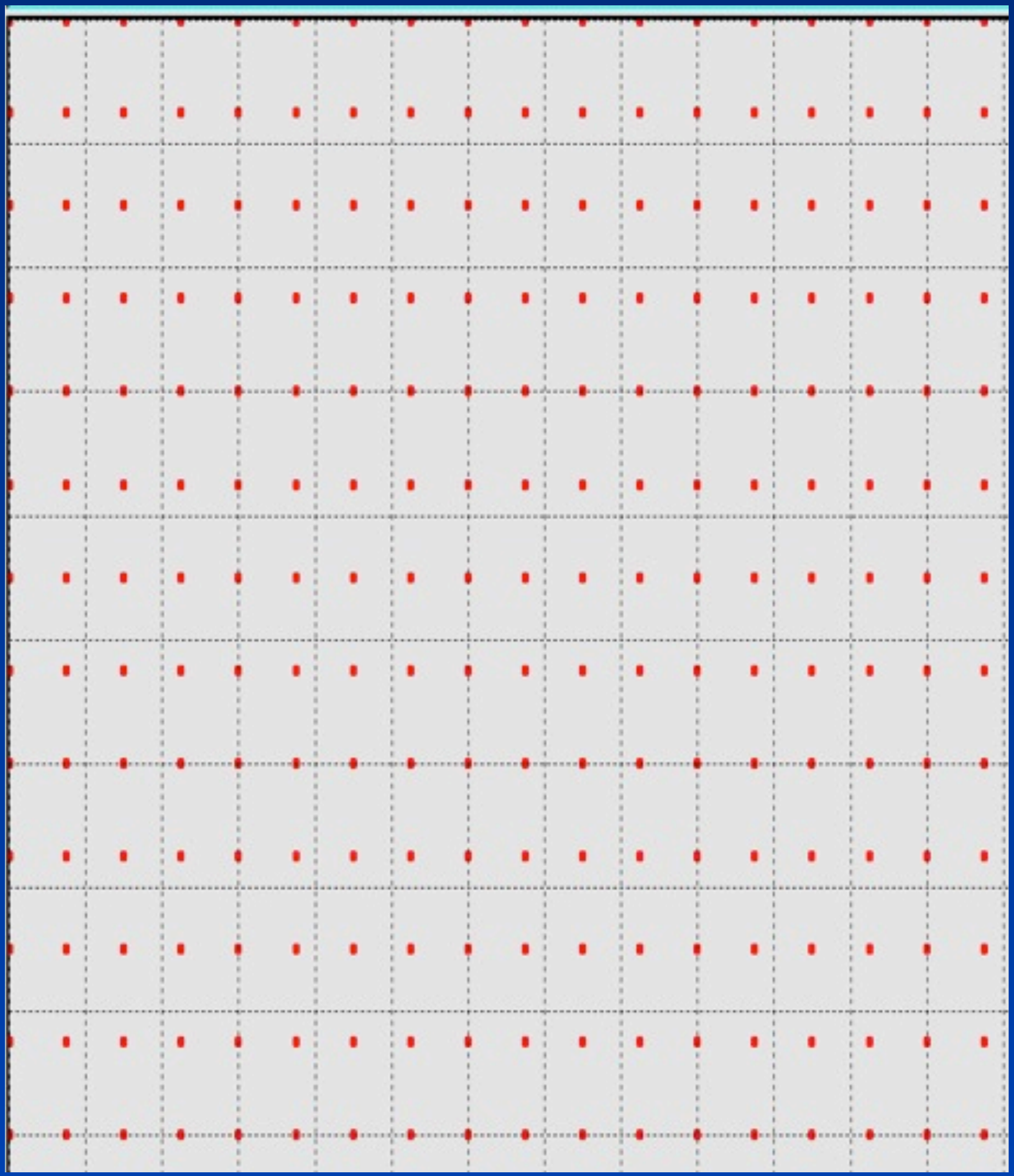
Regular center distribution



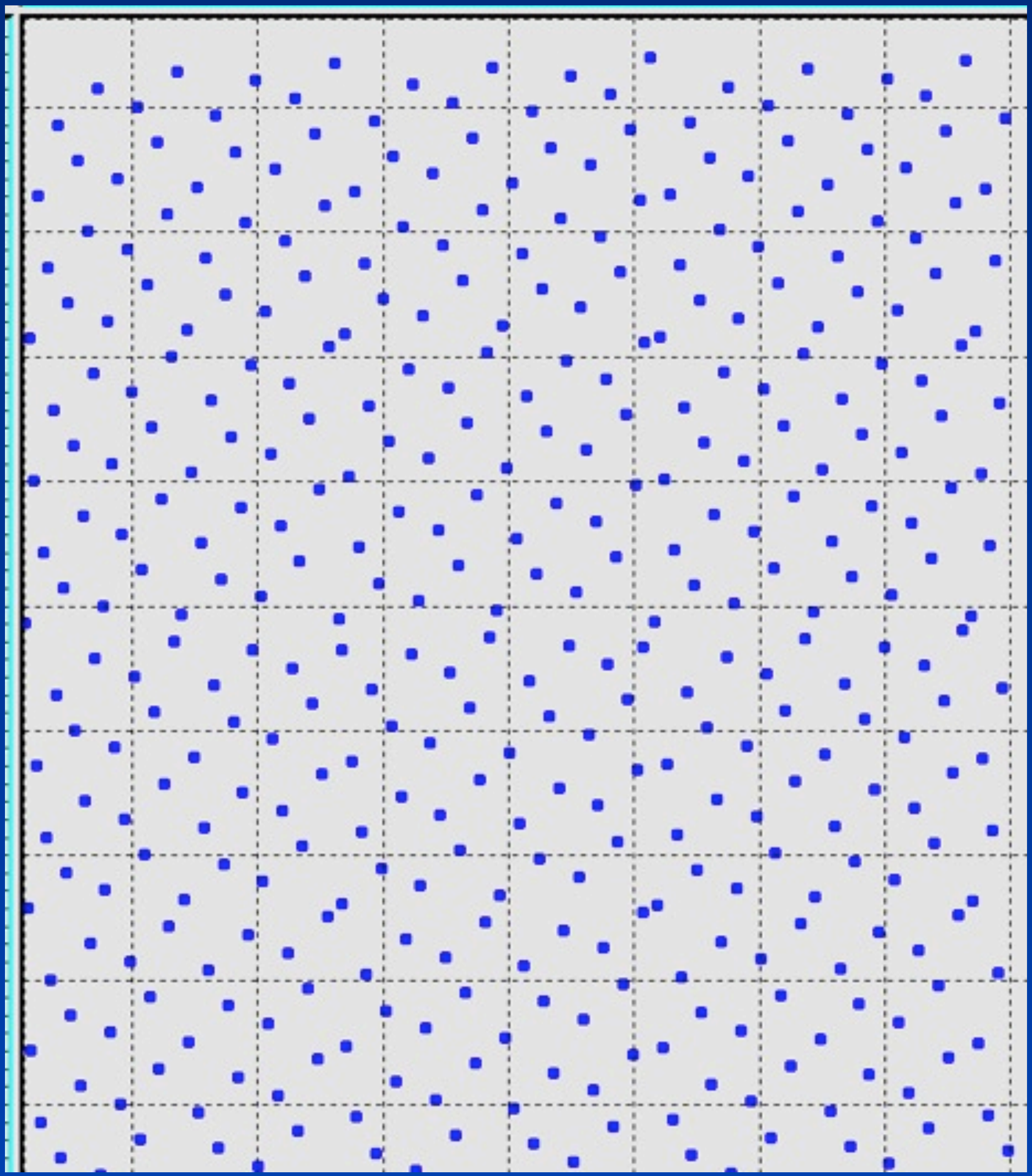
Courtesy Nick Moldoveanu

Coil center grid design

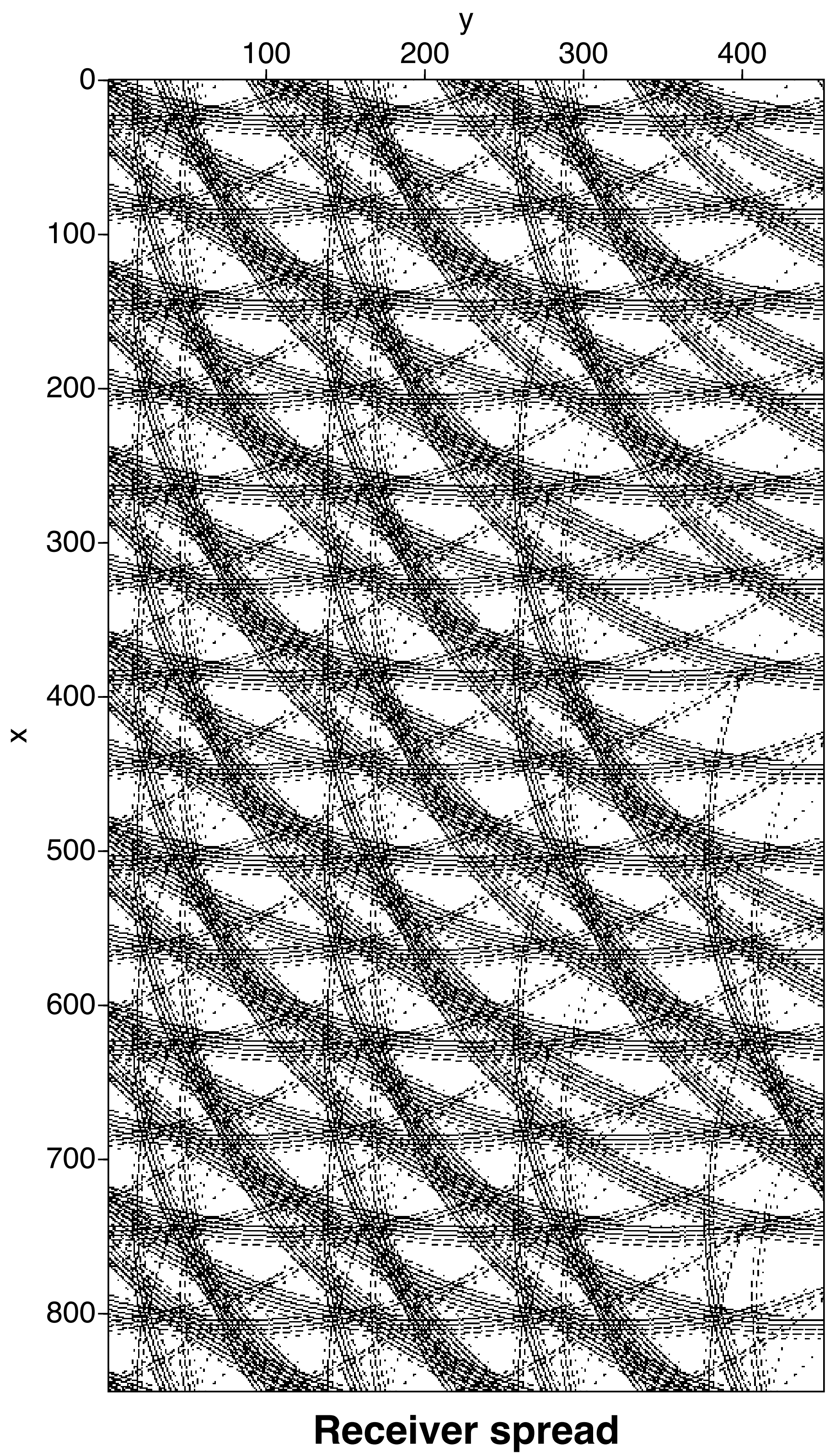
Regular center distribution



Random center distribution

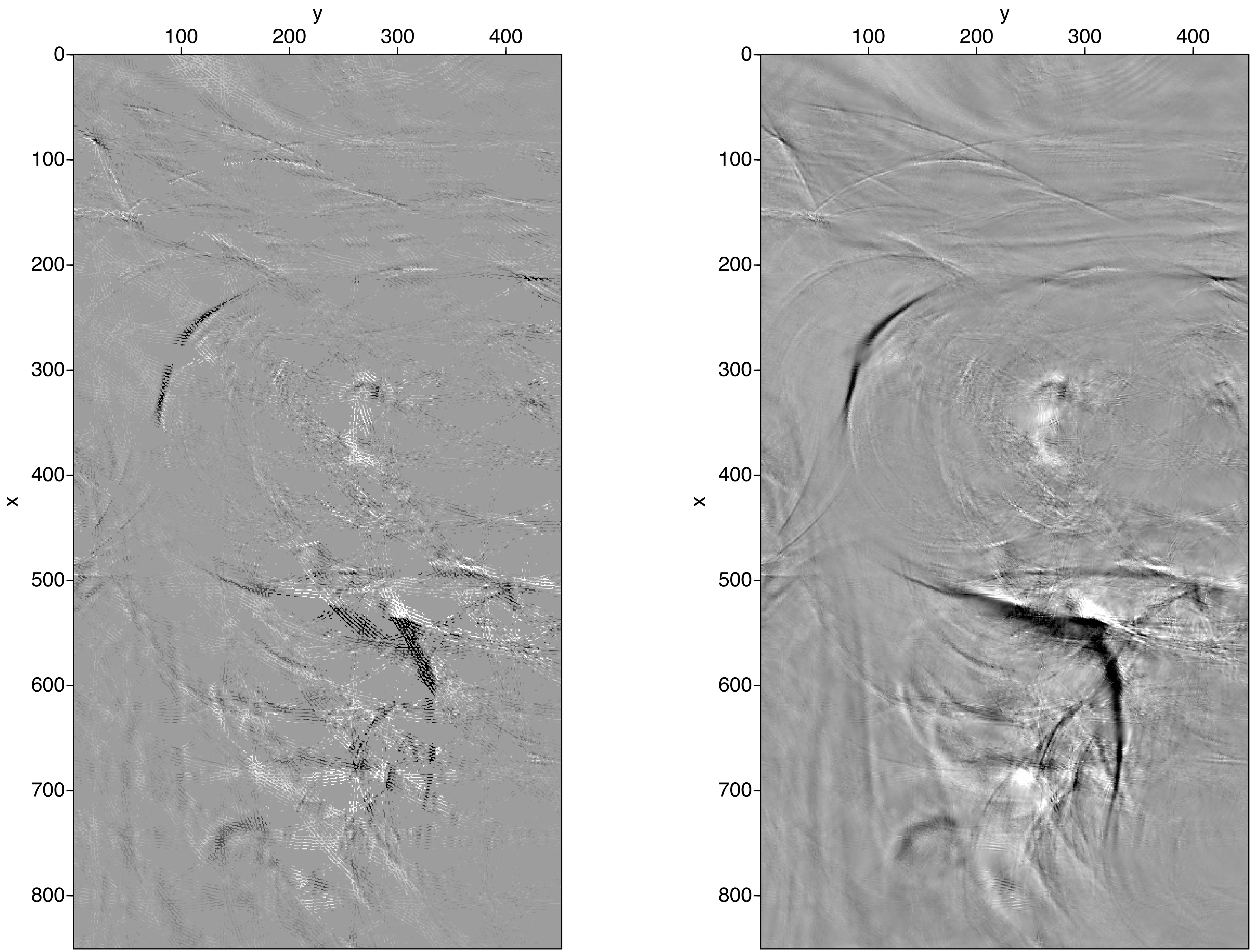


Courtesy Nick Moldoveanu



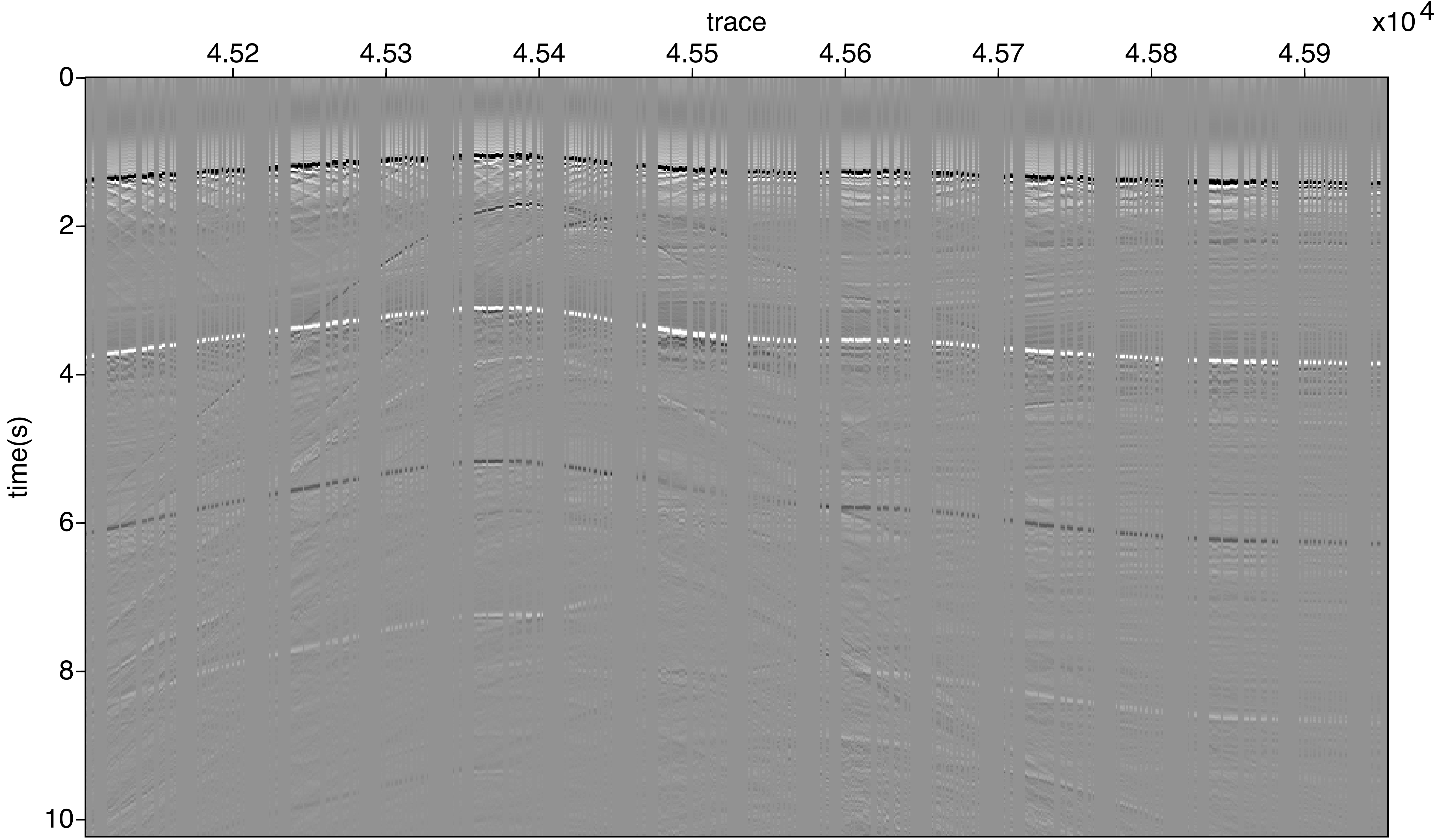
34 % of samples

Courtesy Nick Moldoveanu

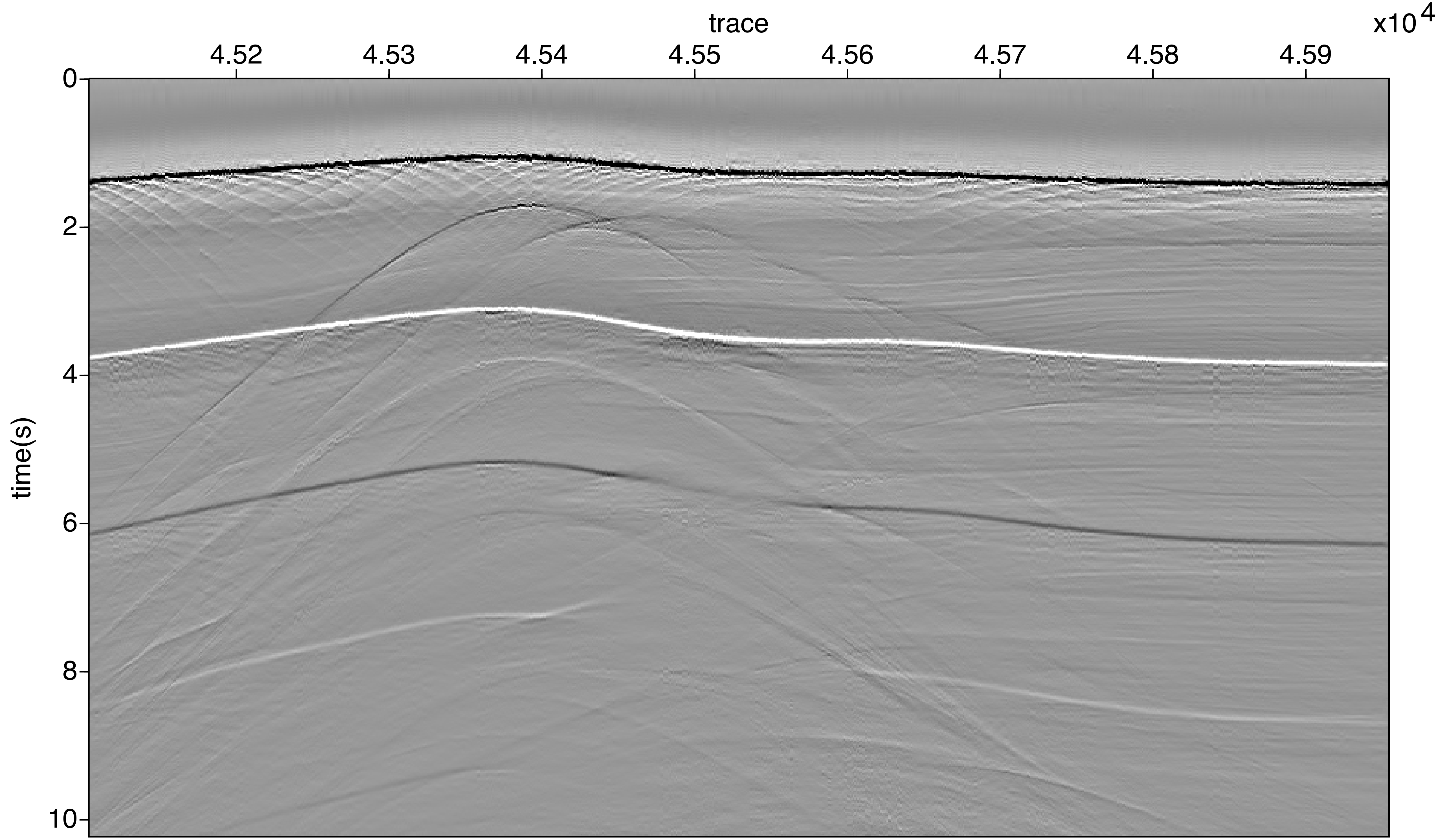


Input data

Interpolation with 2D Curvelet



Input data

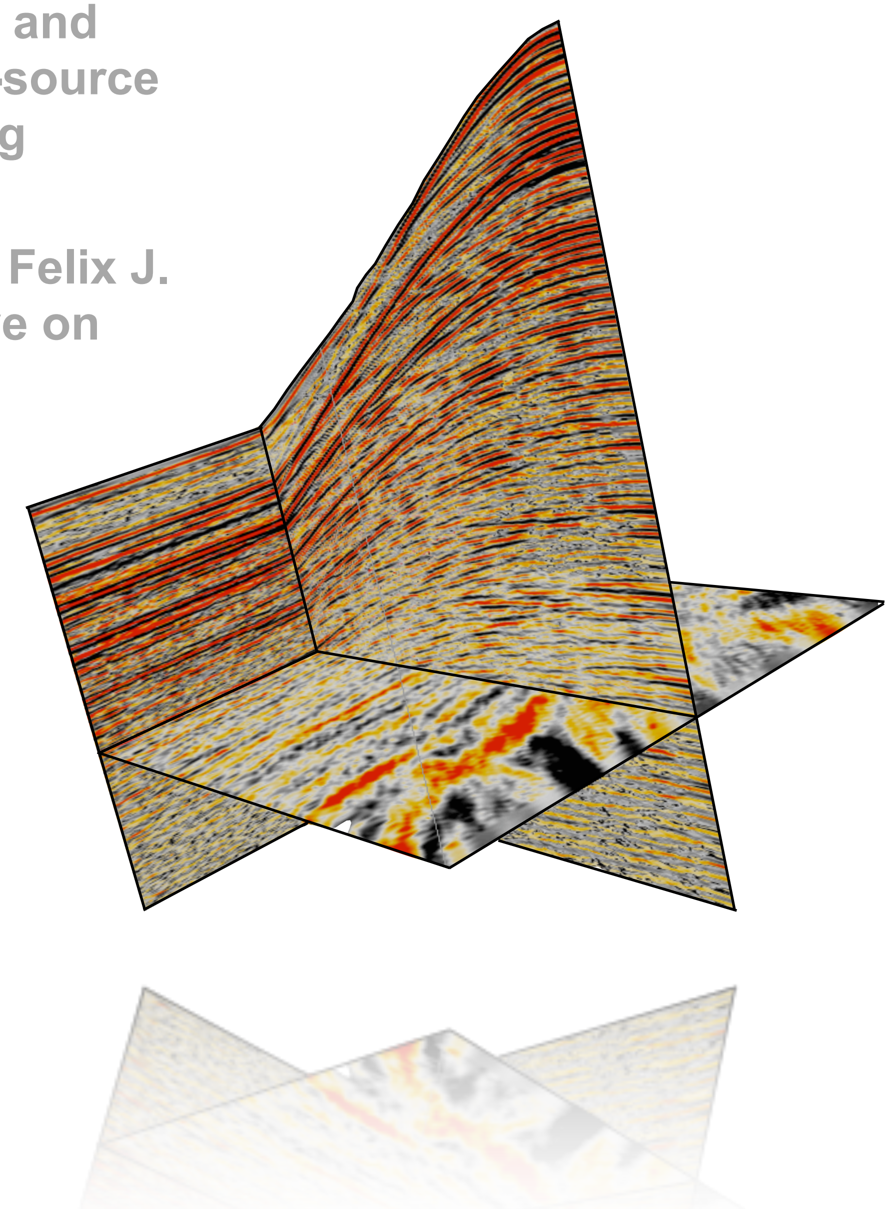


Interpolation with 2D Curvelet

Randomized marine acquisition

Hassan Mansour, Haneet Wason, Tim T.Y. Lin, and Felix J. Herrmann. TR-2011-04. Simultaneous-source marine acquisition with compressive sampling matrices.

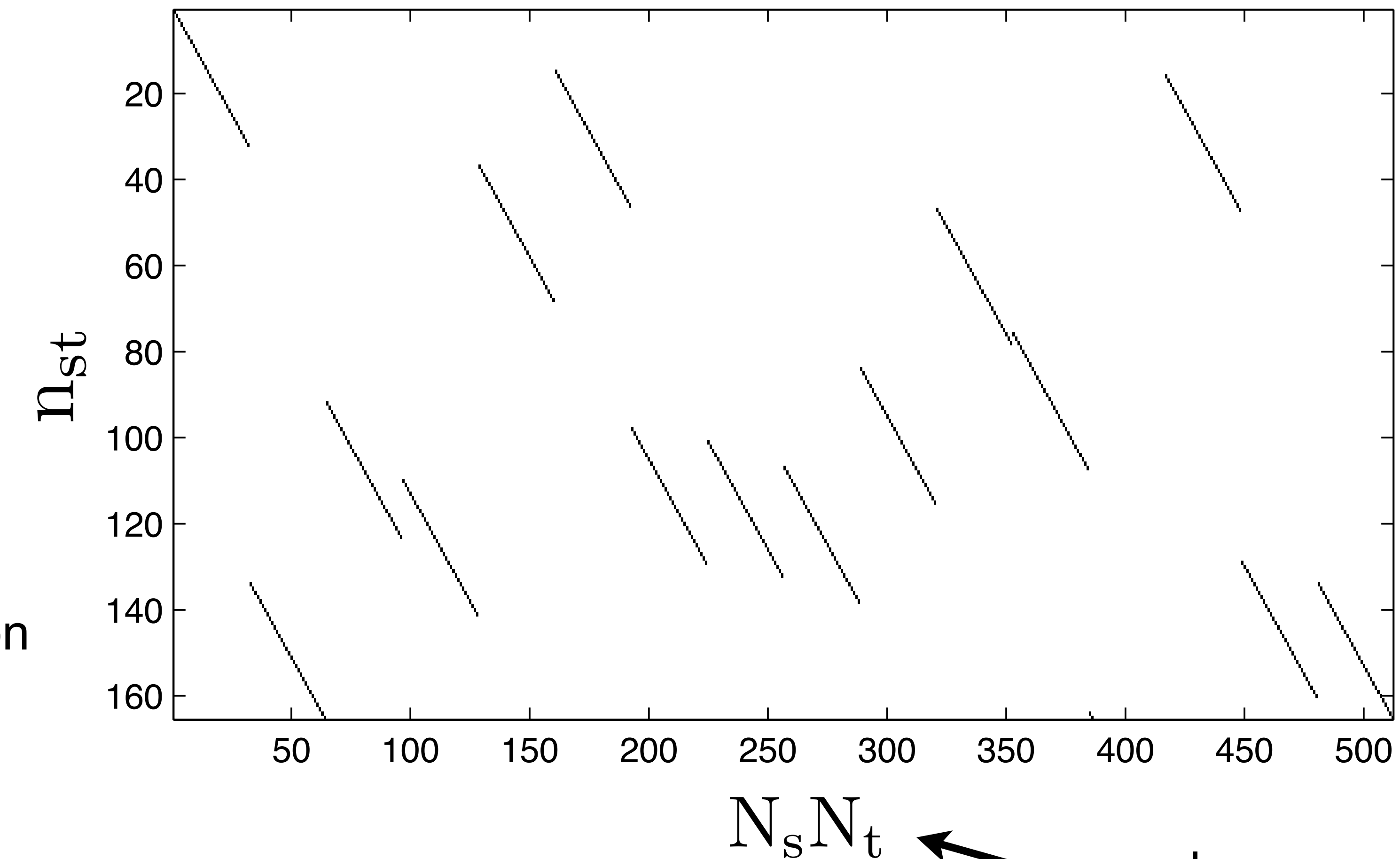
Hassan Mansour, Haneet Wason, Tim Lin and Felix J. Herrmann. A compressive sensing perspective on simultaneous marine acquisition. SBGF 2011.



Simultaneous acquisition matrix

For a seismic line with N_s sources, N_r receivers, and N_t time samples, the sampling matrix is

$$\mathbf{RM}$$

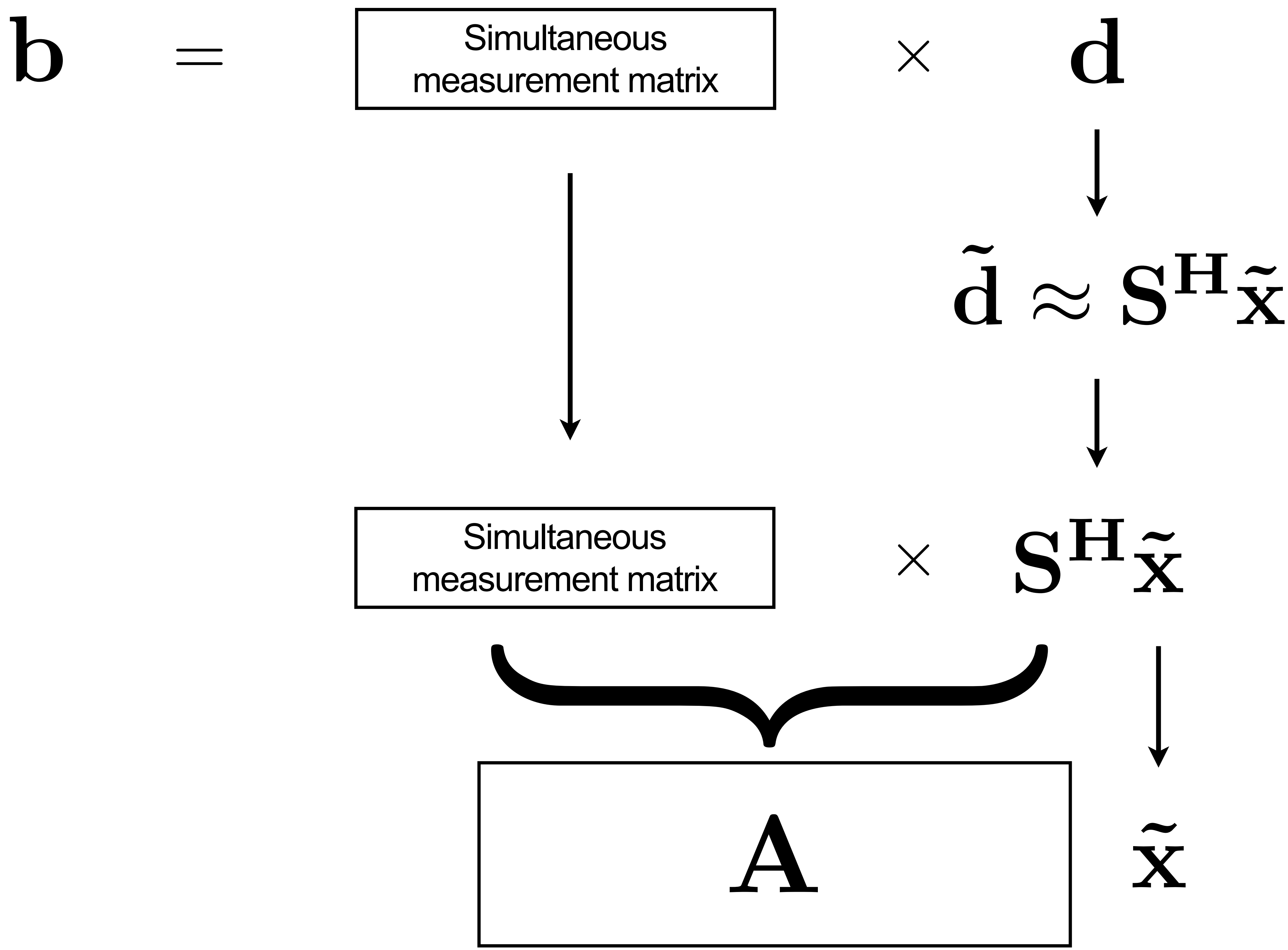


samples recorded at each receiver during simultaneous acquisition

samples recorded at each receiver during sequential acquisition

[Mansour et.al., '11]

Bigger picture



Bigger picture

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

$$\tilde{\mathbf{d}} \approx \mathbf{S^H}\tilde{\mathbf{x}}$$

$$\boxed{\mathbf{RM}} \times \mathbf{S^H}\tilde{\mathbf{x}}$$

$$\underbrace{\boxed{\mathbf{RM}} \times \mathbf{S^H}}_{\boxed{\mathbf{A}}} \; \tilde{\mathbf{x}}$$

"Compressive sampling matrix" ← $\mathbf{A}$

Sparse recovery

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

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \|\mathbf{x}\|_1 \quad \text{subject to} \quad \mathbf{A}\mathbf{x} = \mathbf{b}$$

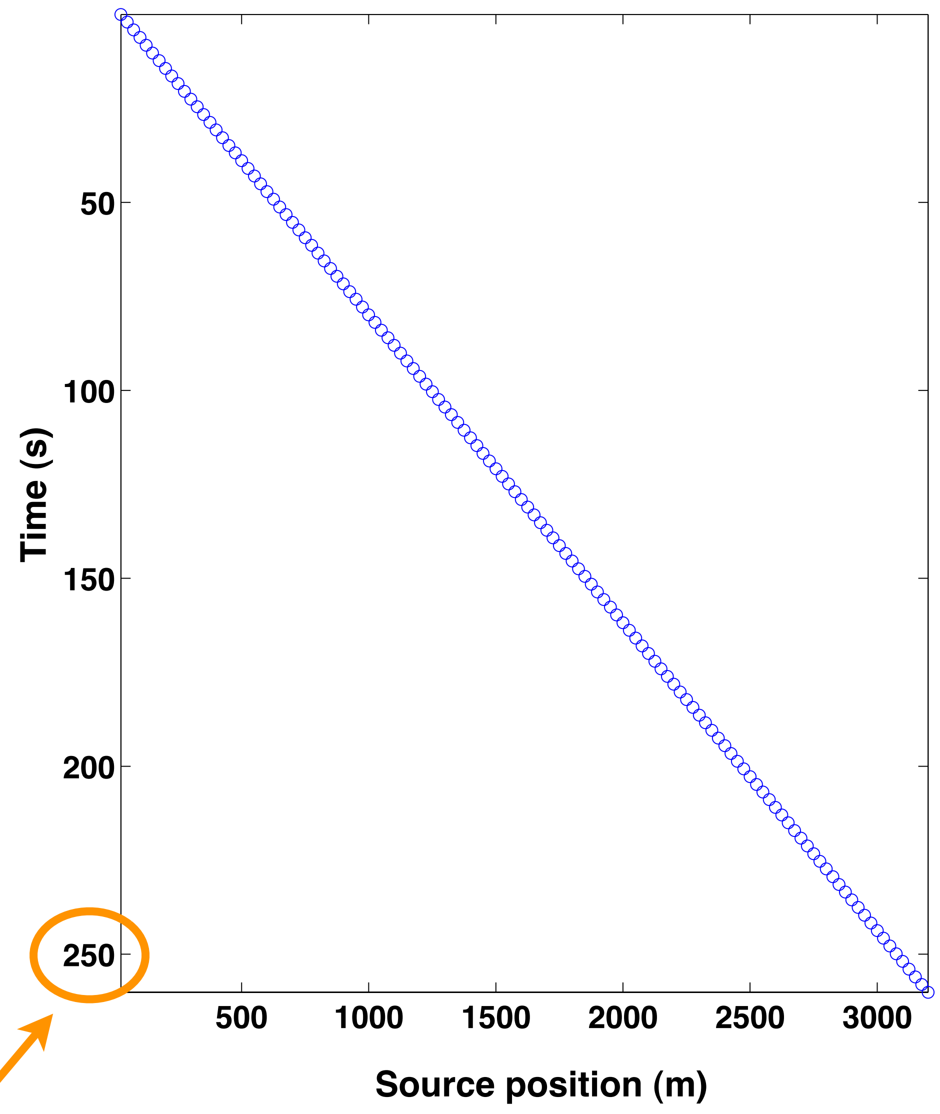
"sparsity"

data-consistent amplitude recovery

Sparsity-promoting solver: $\mathbf{SPG}\ell_1$ [van den Berg and Friedlander, '08]

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

Sequential vs. simultaneous sources

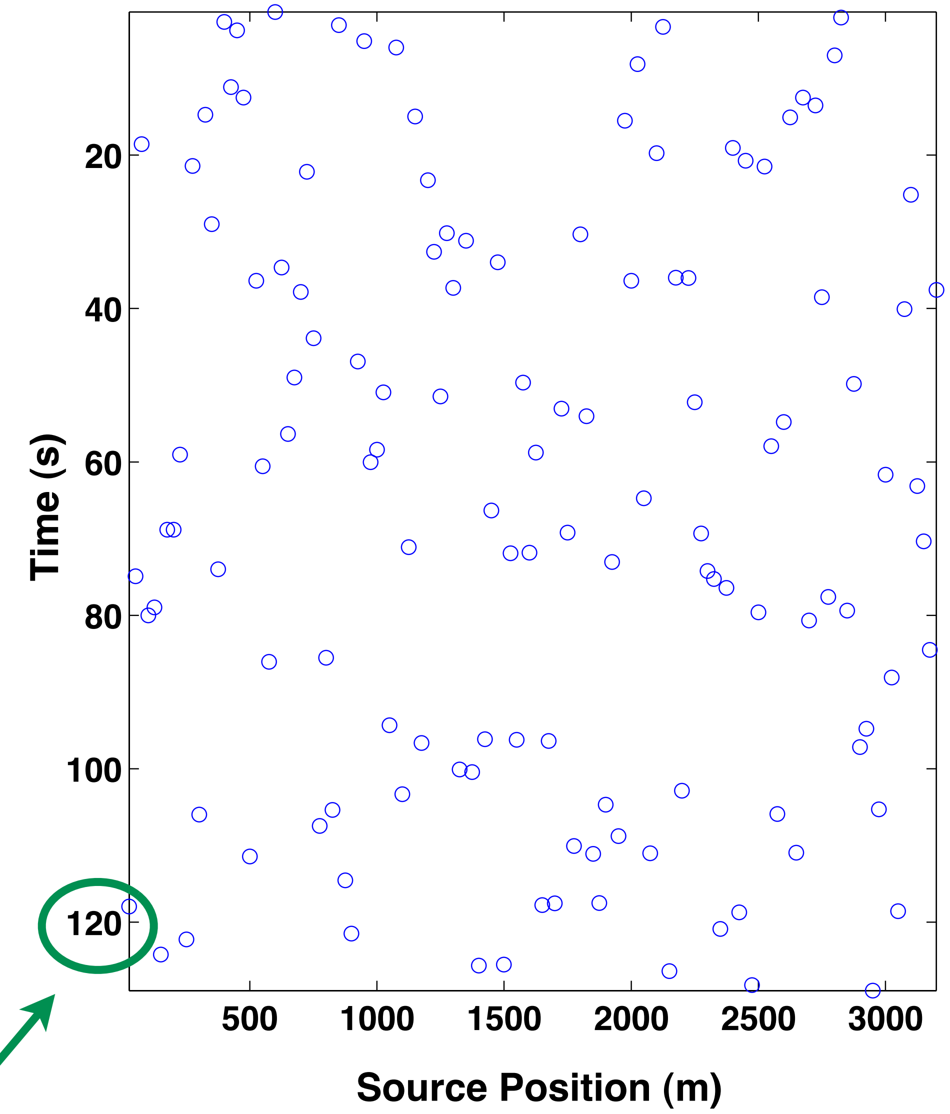


Conventional survey time:

$$t = \mathbf{N_s} \times \mathbf{N_t}$$

Sequential acquisition

Sampling scheme: Random dithering



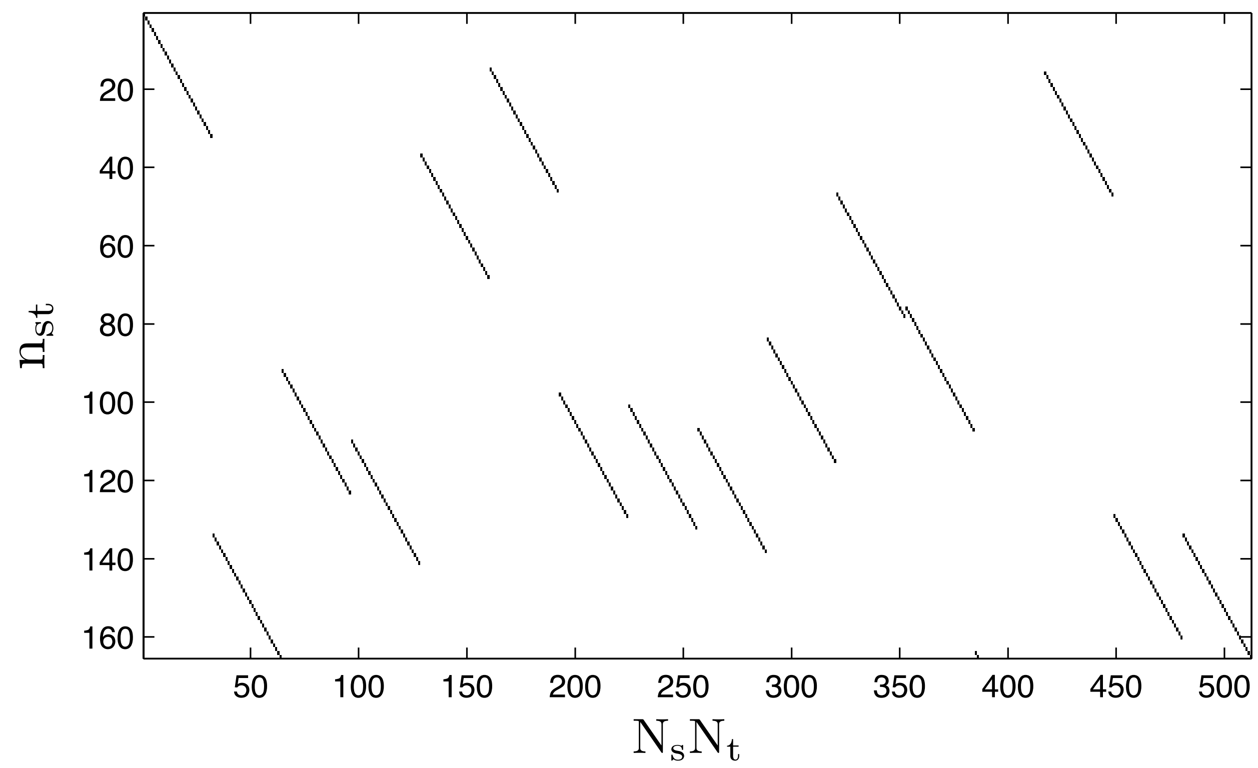
Theoretical survey time:

$$t = n_{st} \ll n_s \times \mathbf{N_t}$$

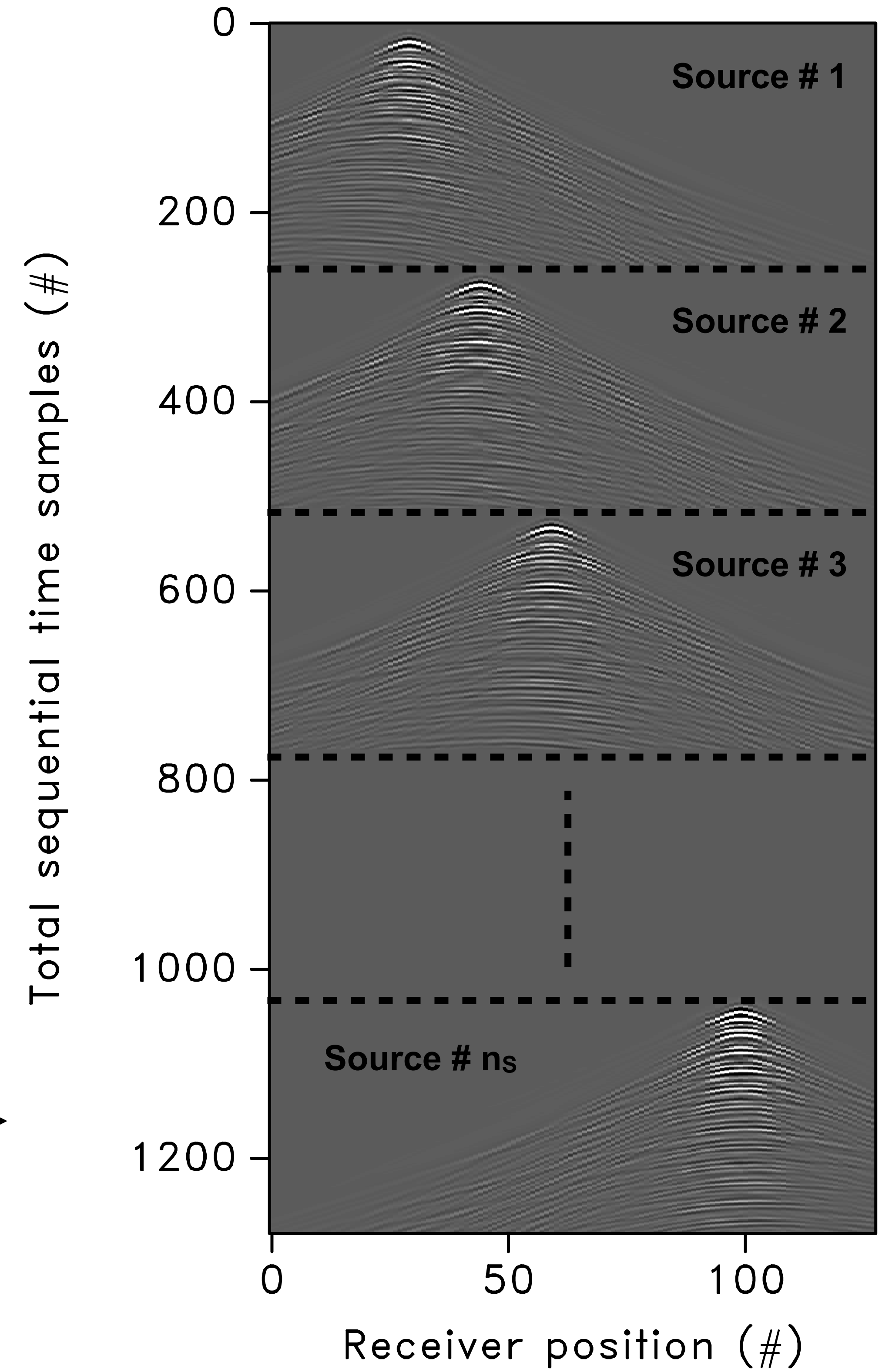
Simultaneous acquisition

Sampling scheme: Random dithering

$\mathbf{RM}$

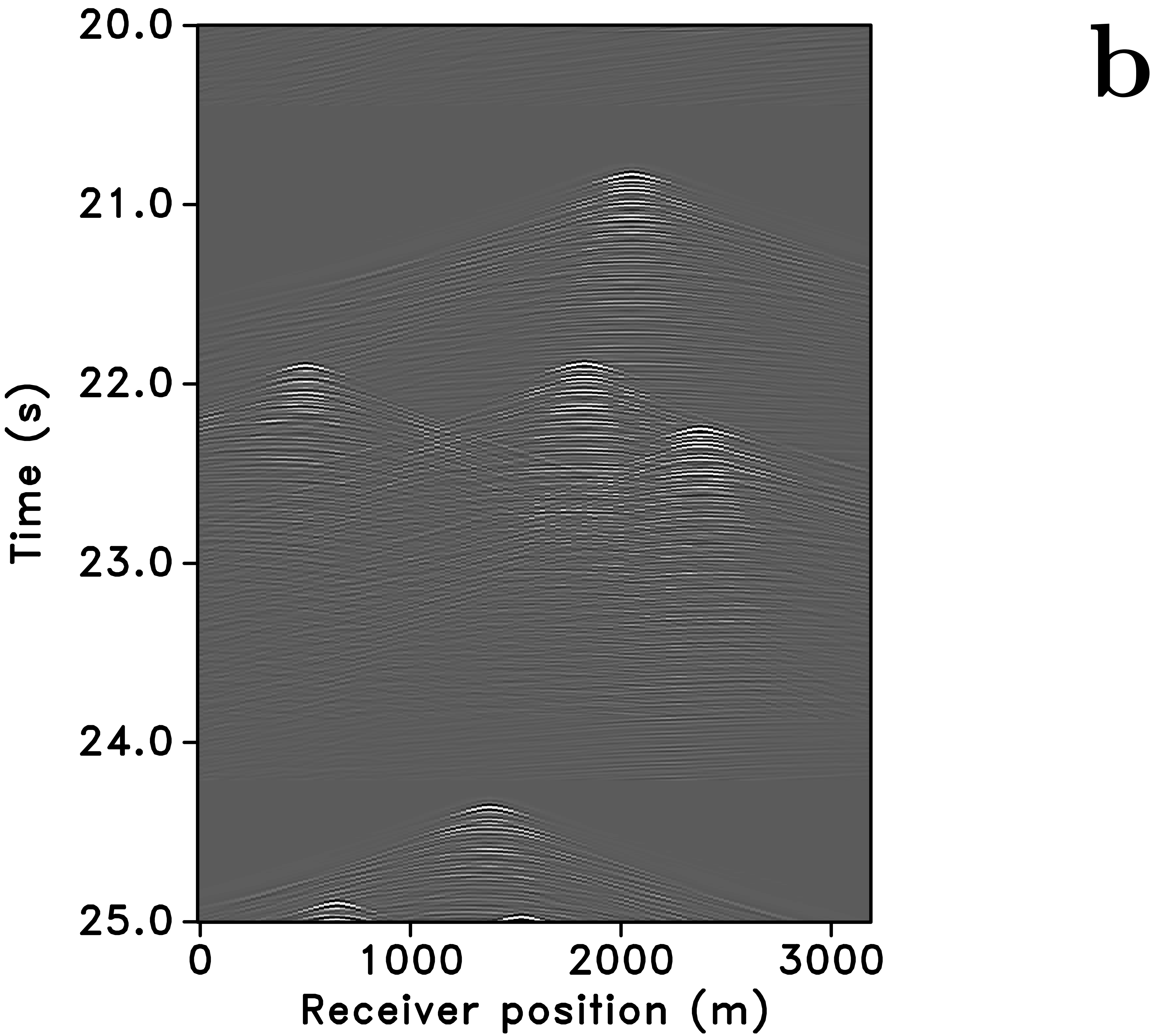


$\mathbf{d}$

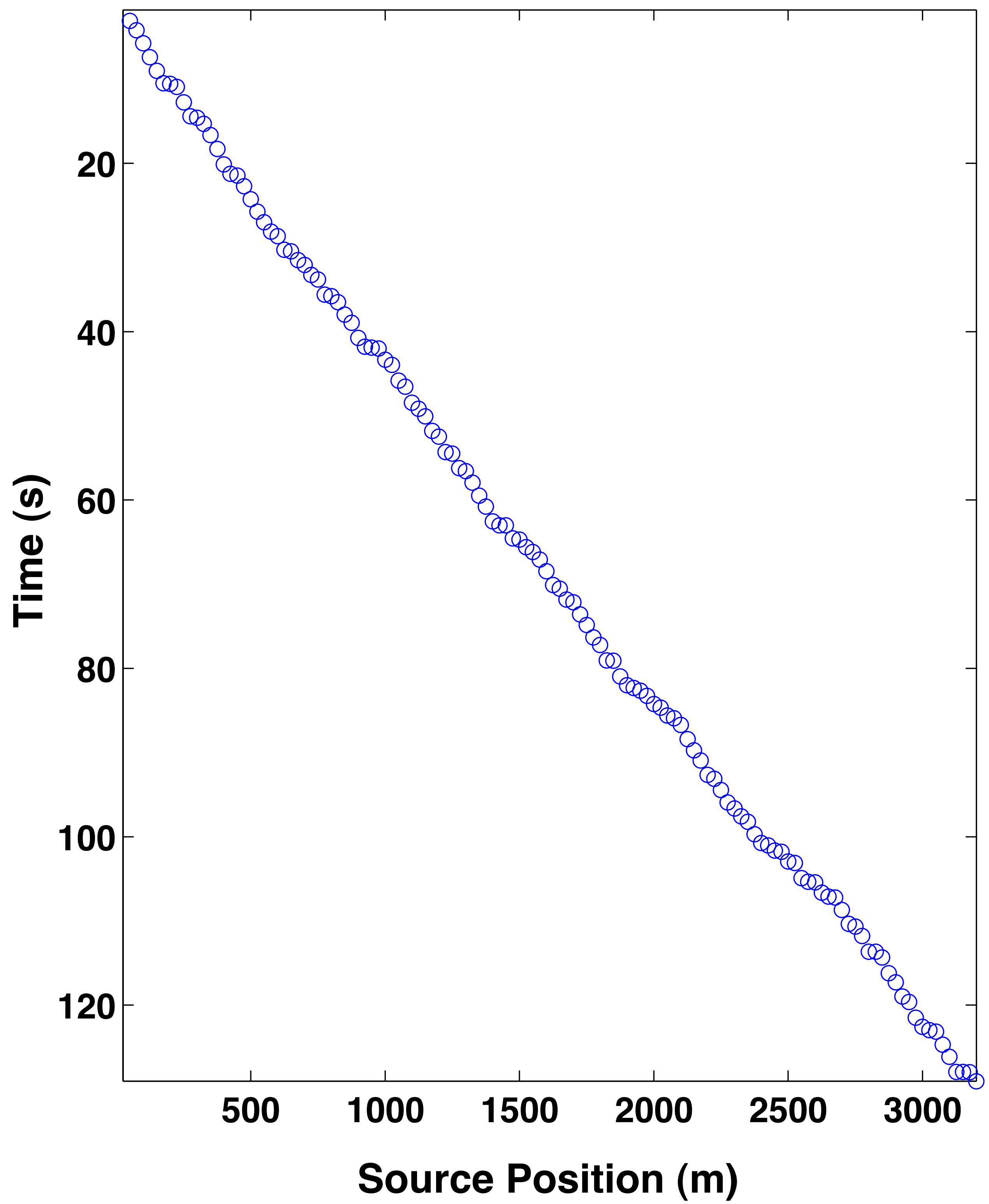


series of sequential shots →

Sampling scheme: Random dithering

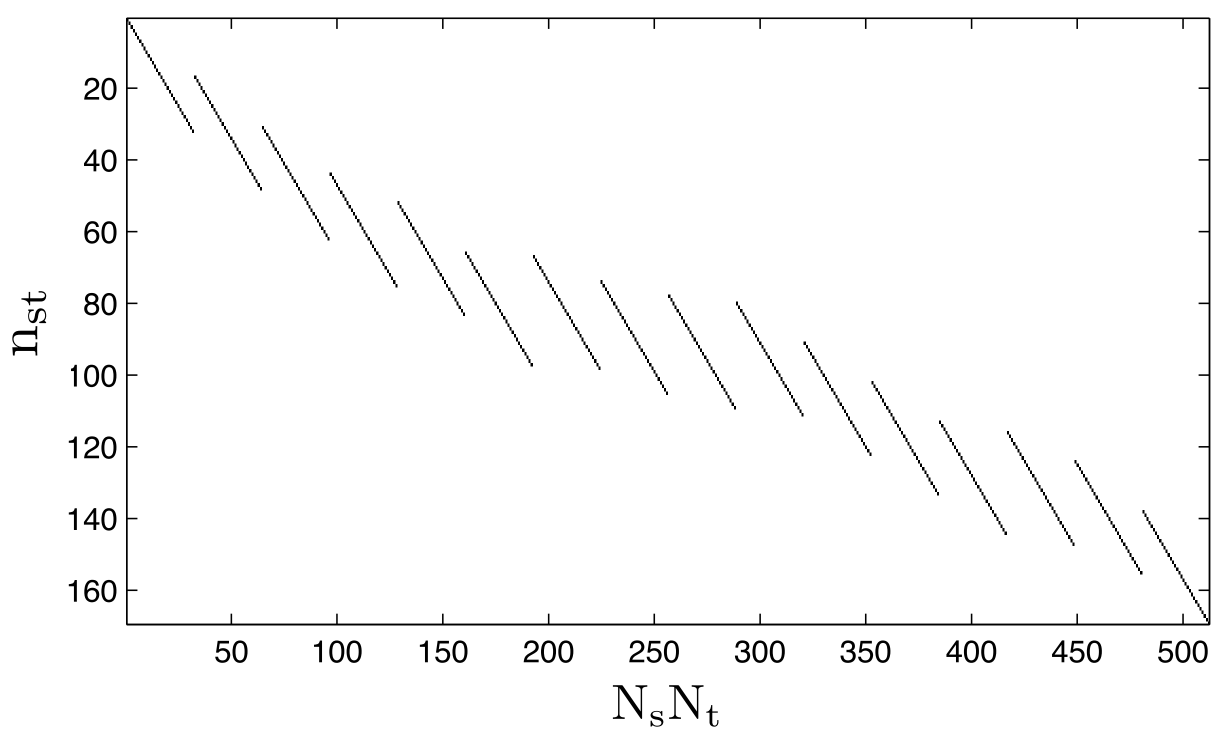


Sampling scheme: Random time-shifting



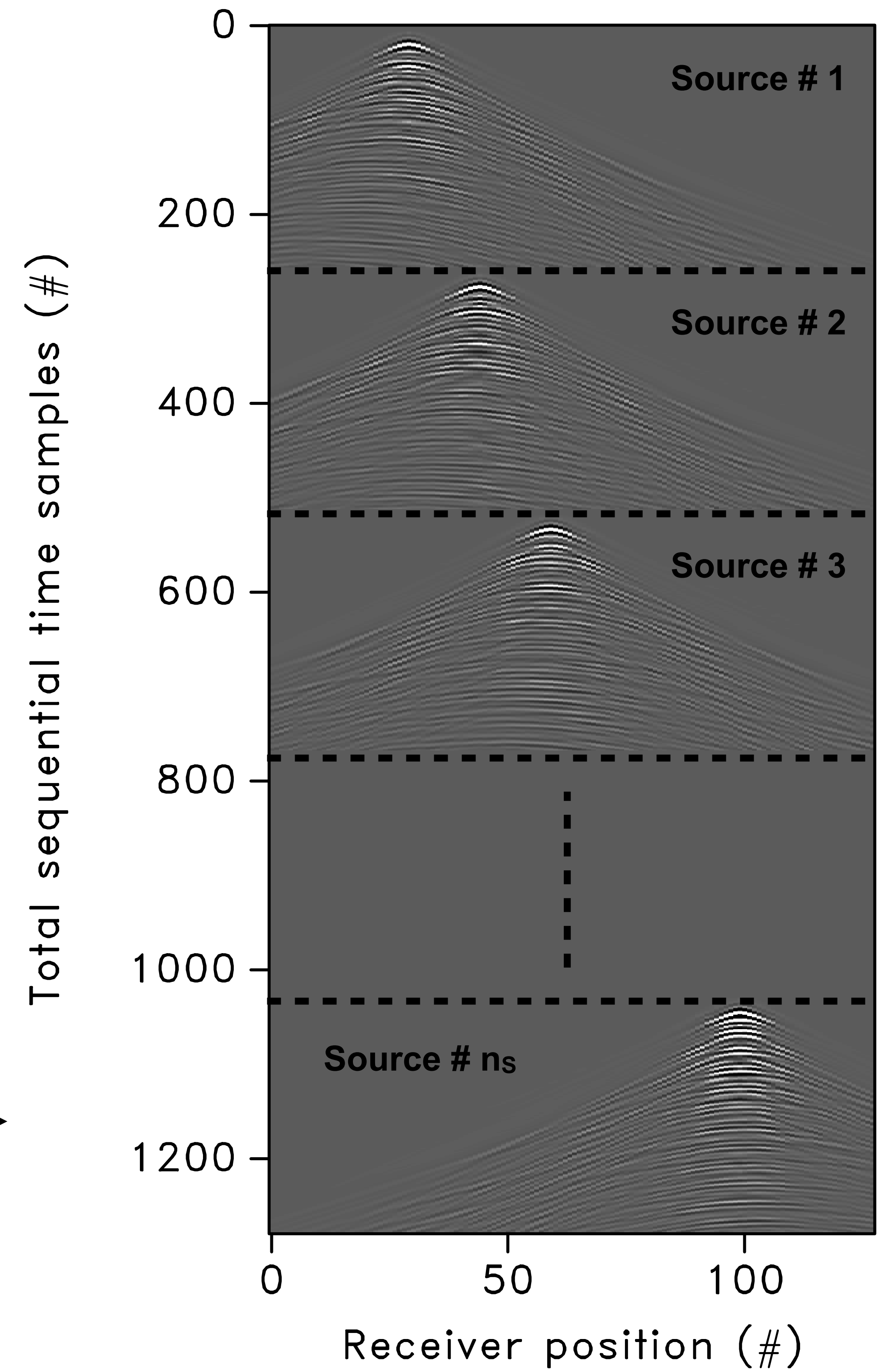
Sampling scheme: Random time-shifting

$\mathbf{RM}$

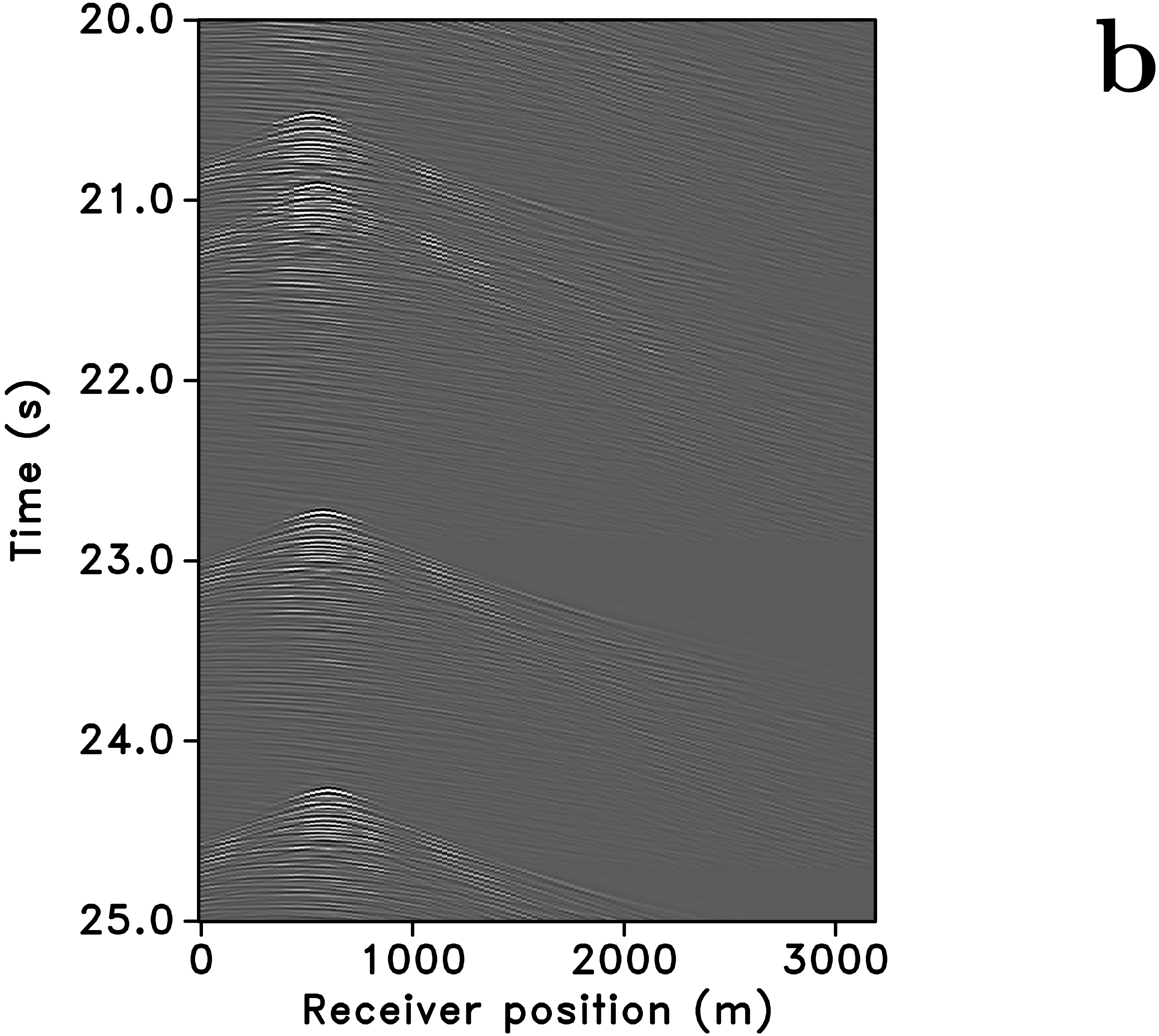


series of
sequential shots

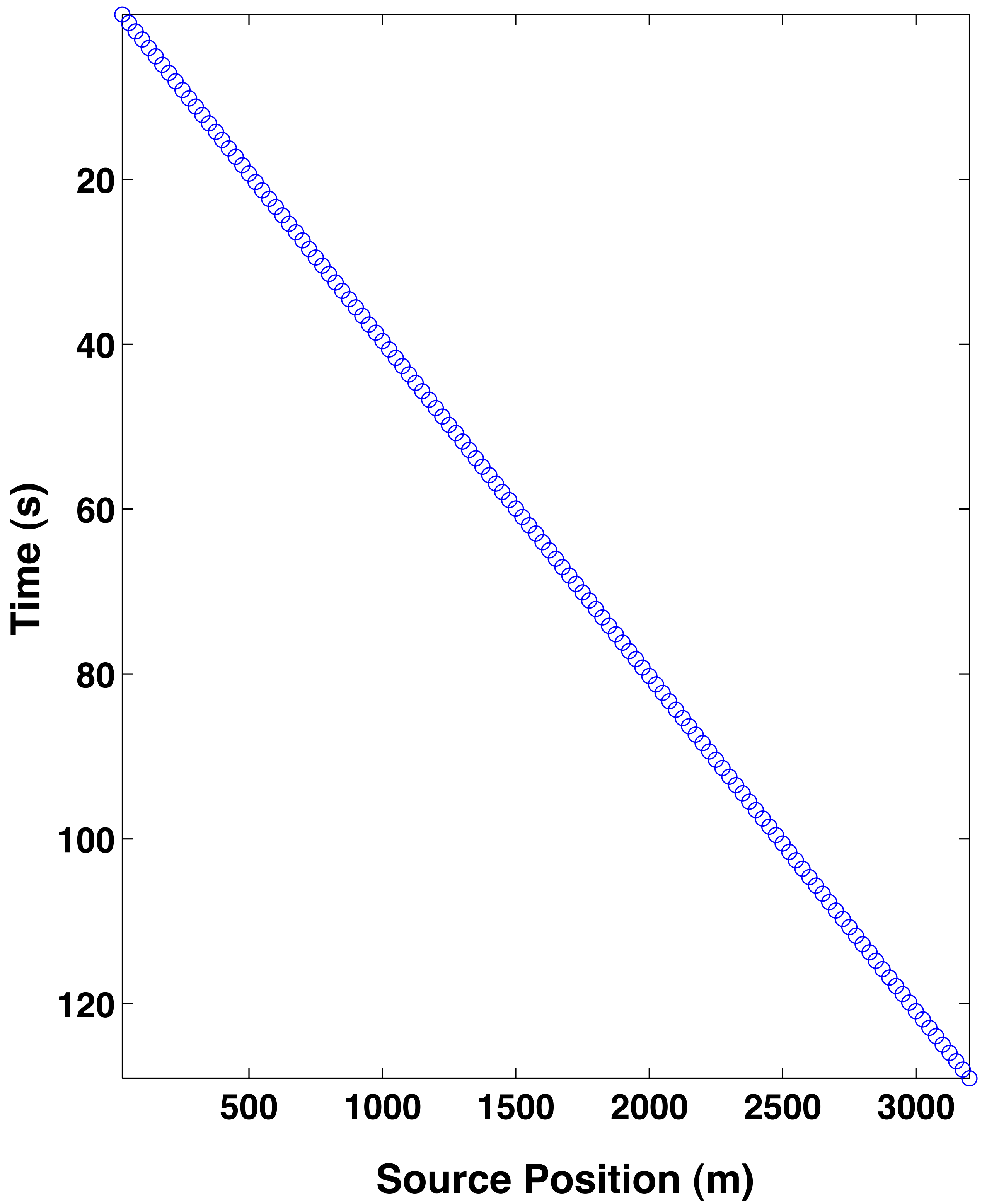
$\mathbf{d}$



Sampling scheme: Random time-shifting

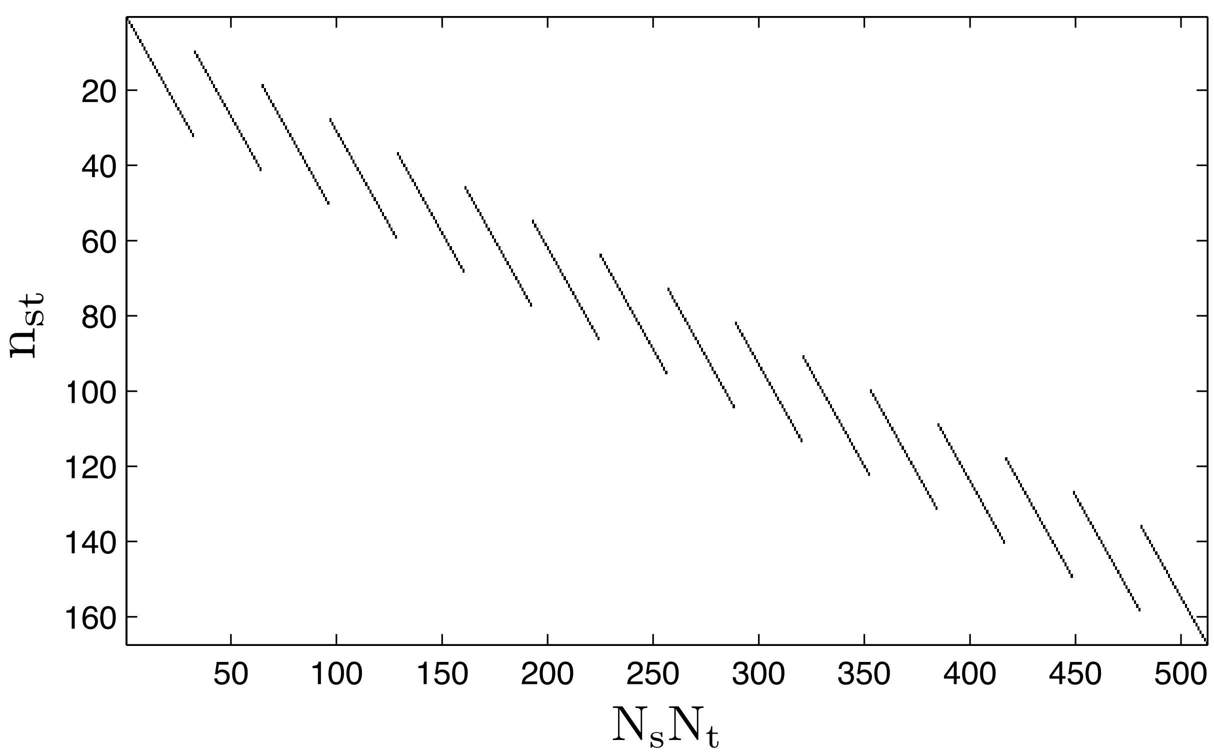


Sampling scheme: Constant time-shifting

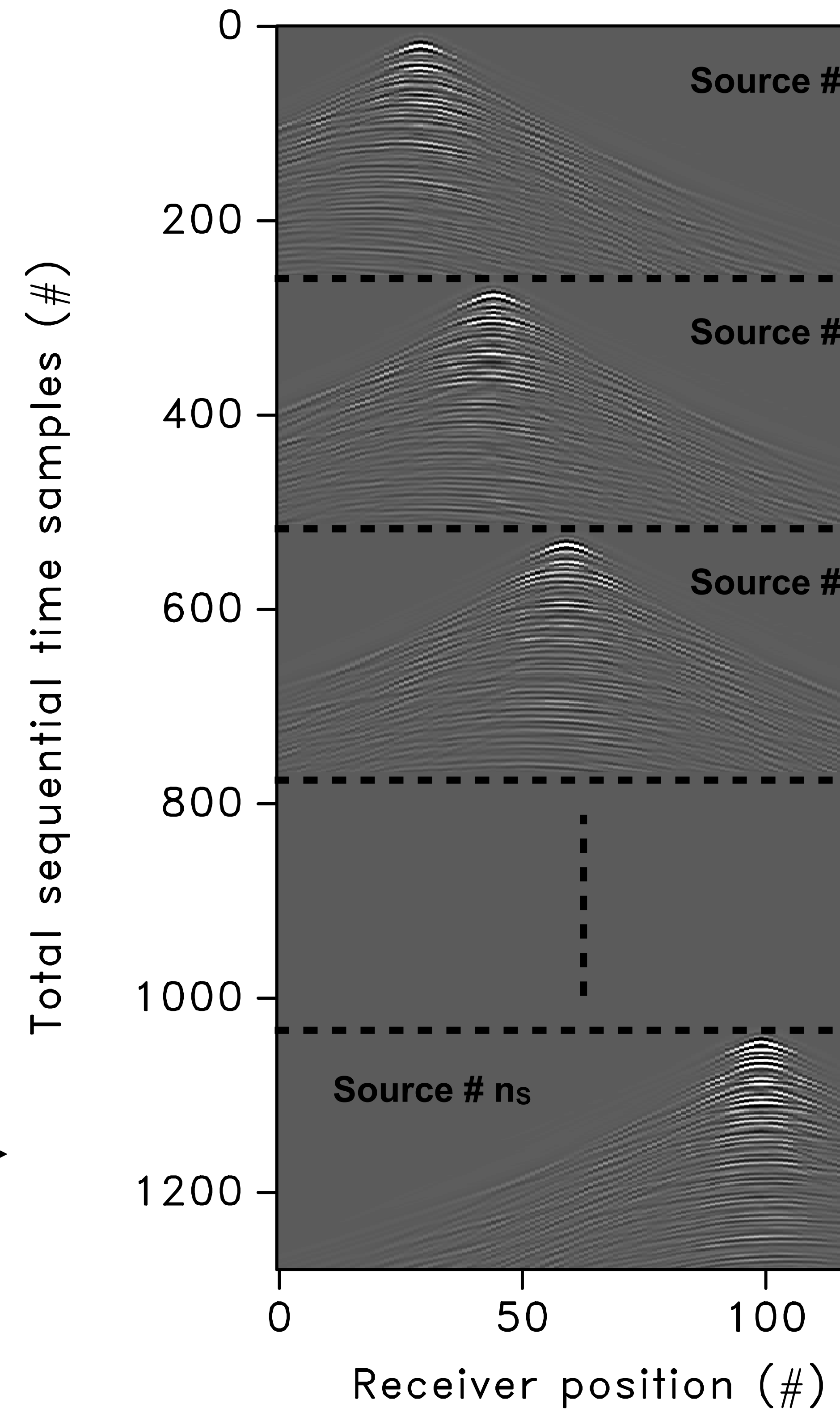


Sampling scheme: Constant time-shifting

$\mathbf{RM}$

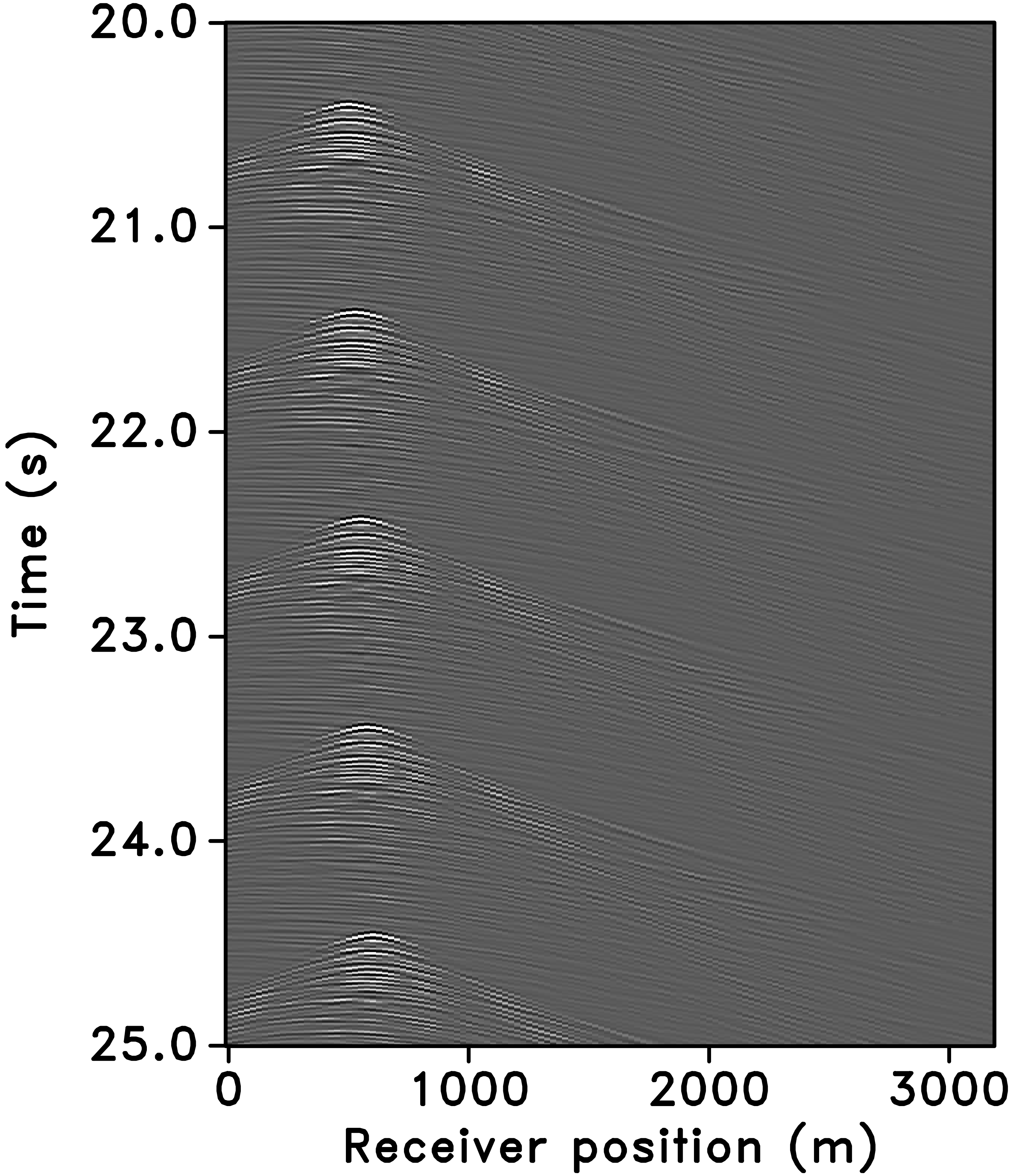


$\mathbf{d}$



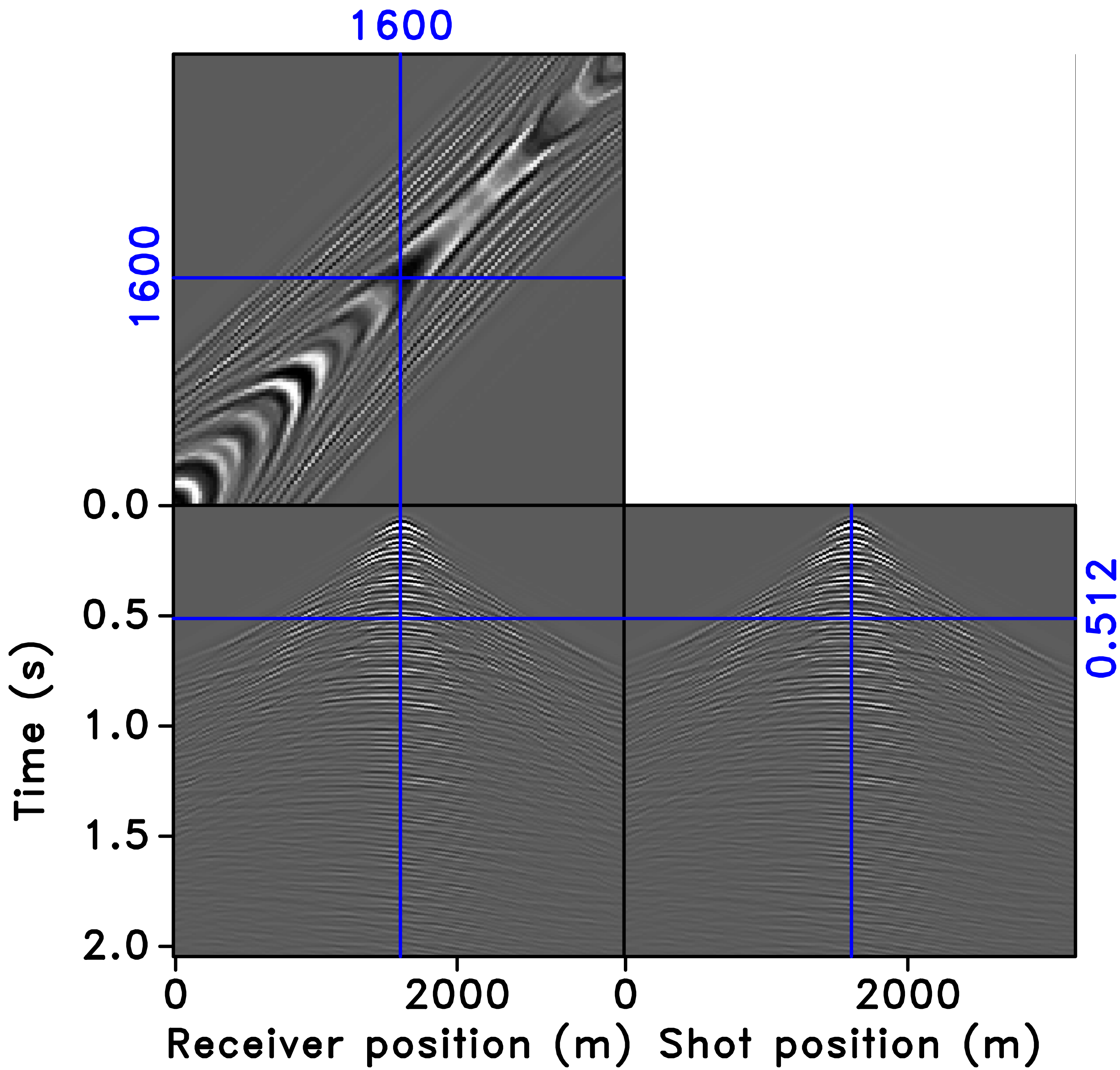
series of sequential shots →

Sampling scheme: Constant time-shifting

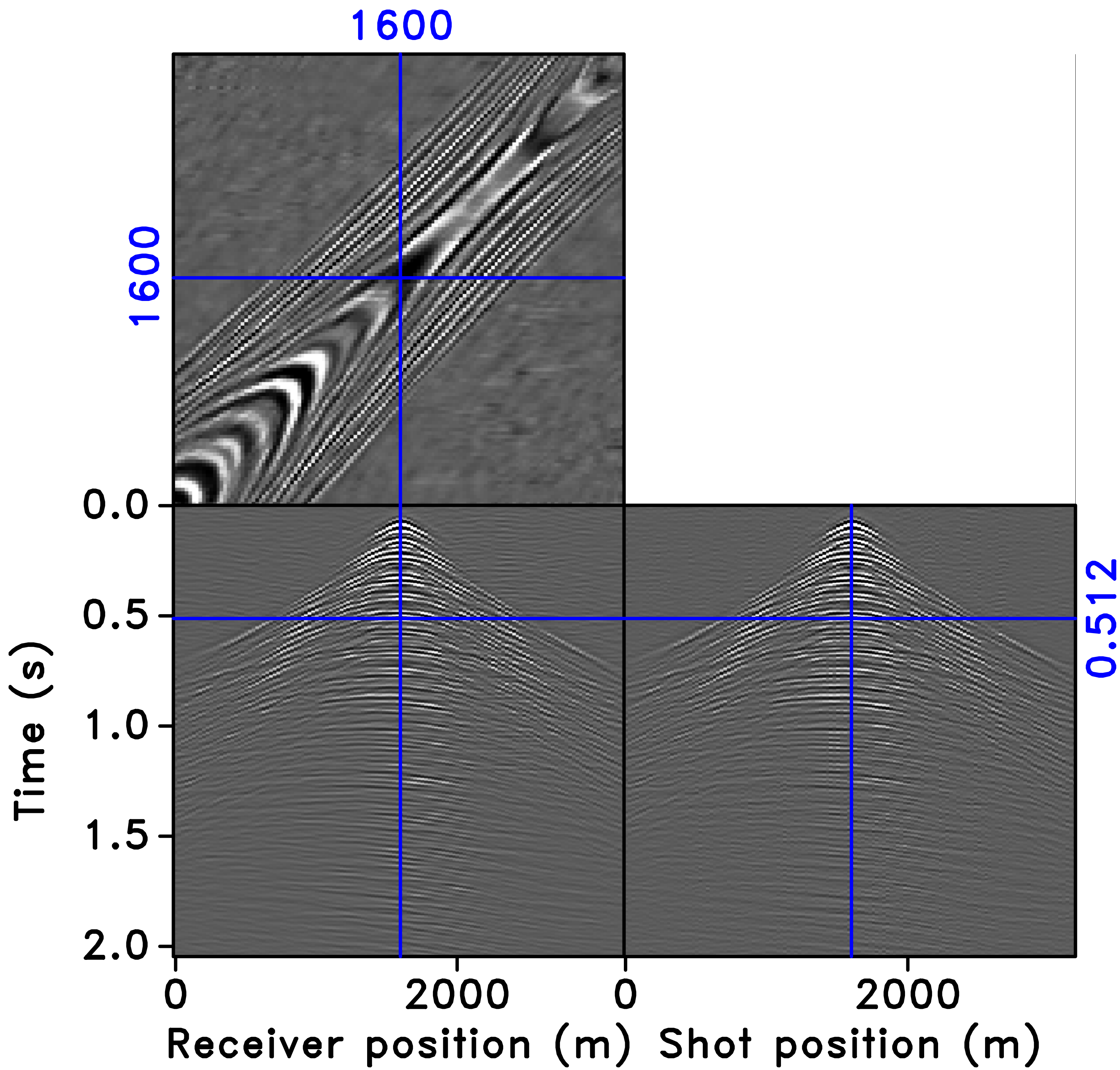


b

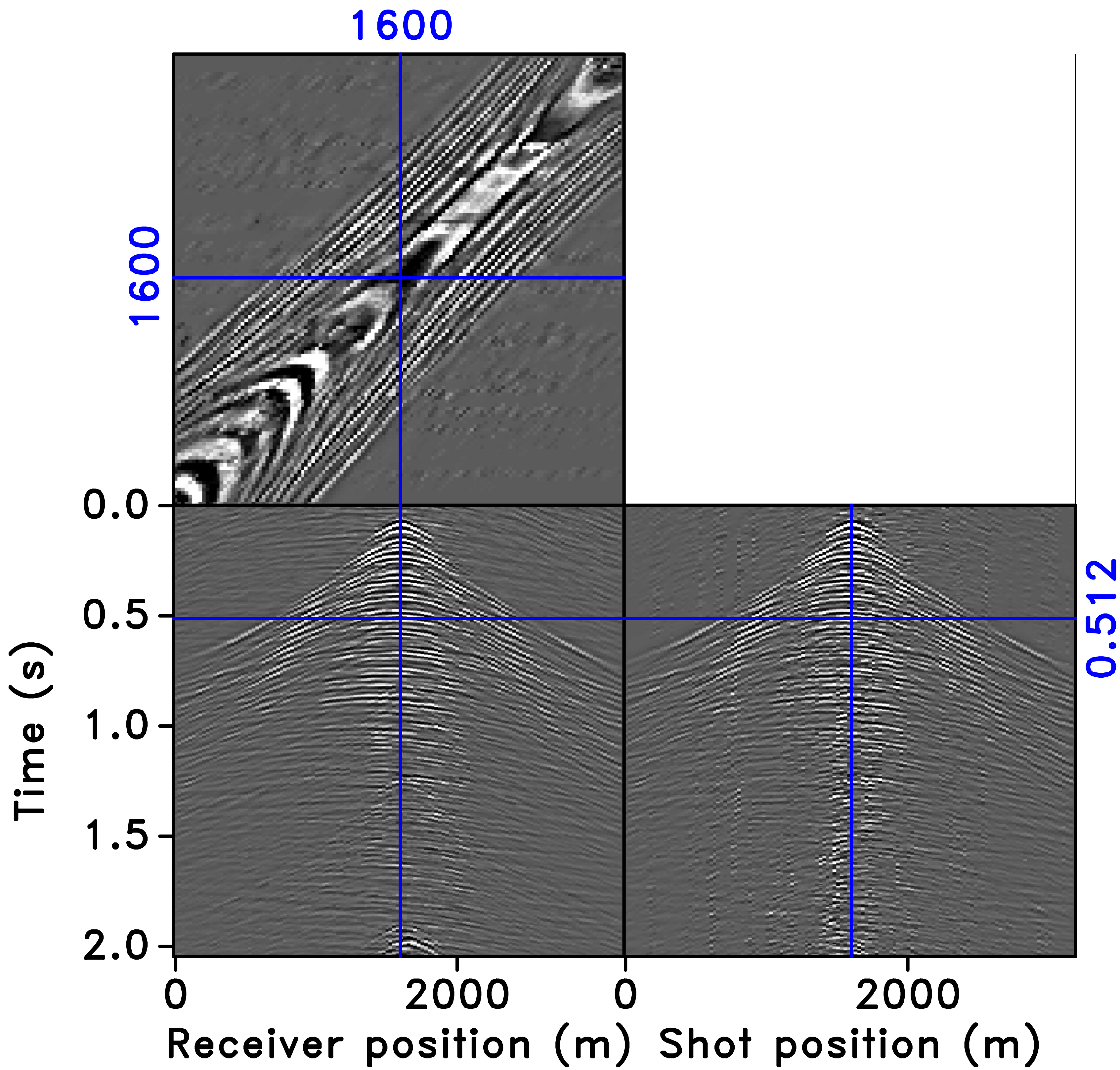
Original data (Sequential acquisition)



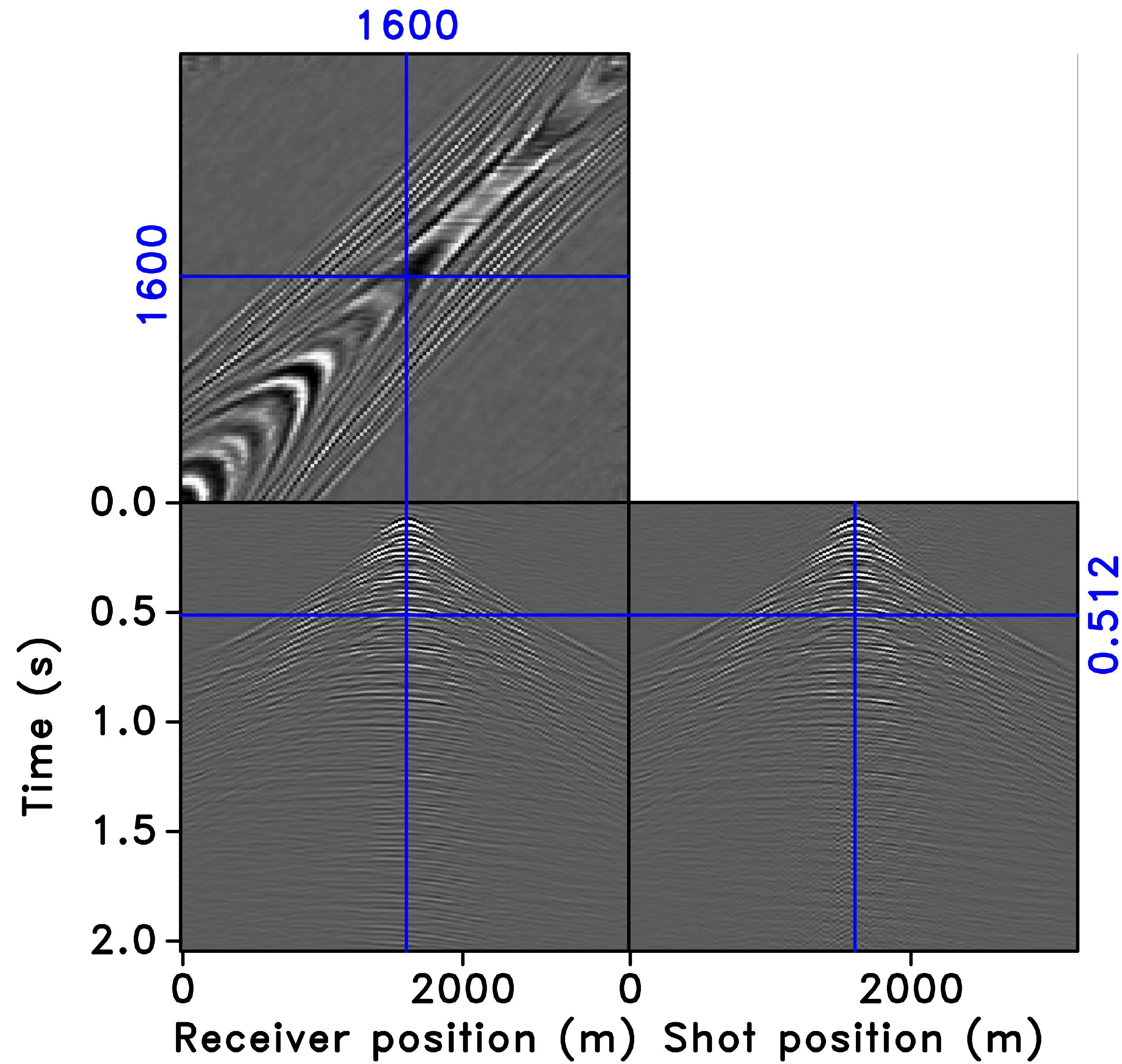
Sparsity-promoting recovery: Random dithering
SNR = 10.5 dB



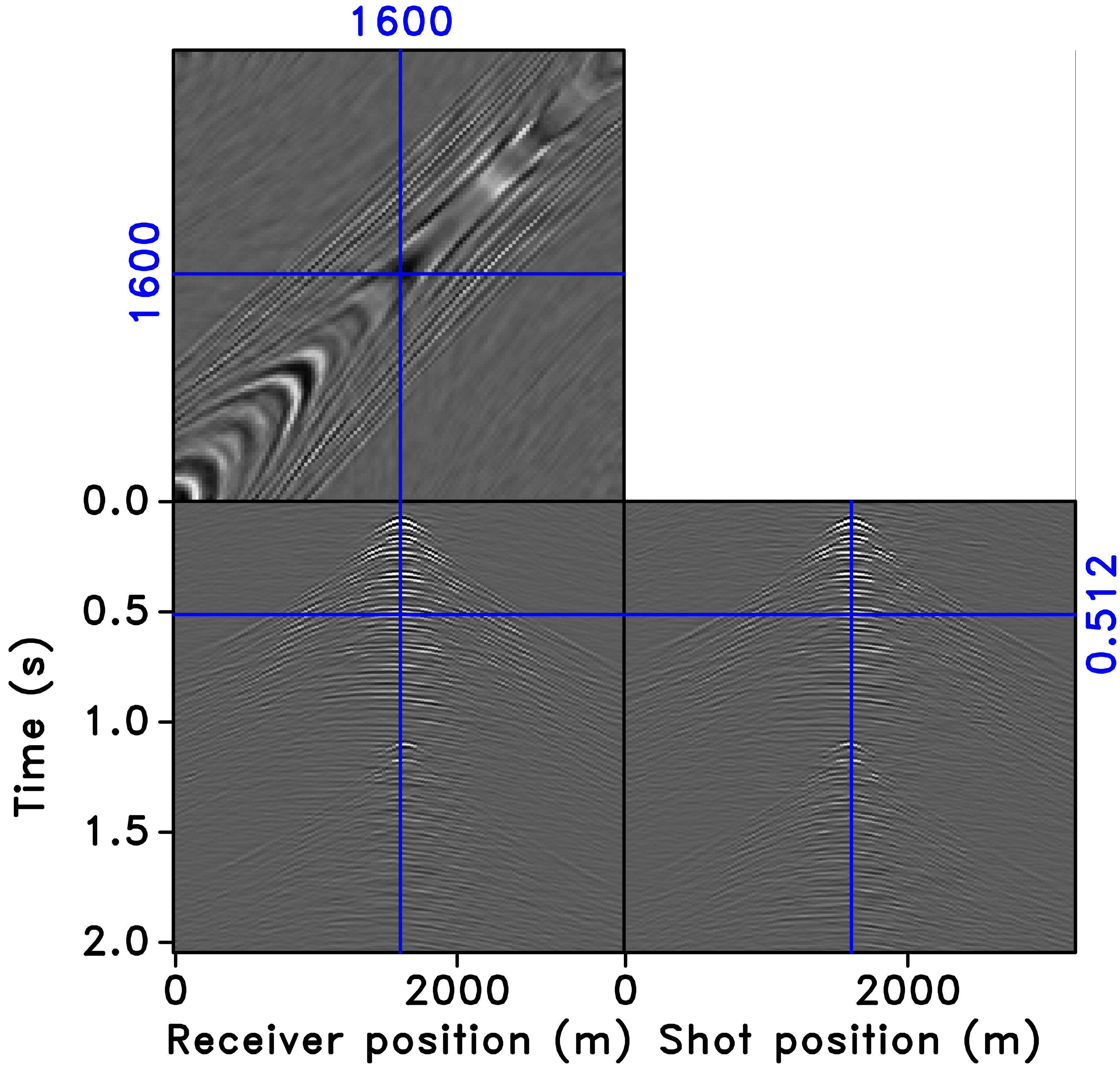
Conventional recovery: Random time-shifting
SNR = 5.04 dB



Sparsity-promoting recovery: Random time-shifting
SNR = 9.52 dB



Sparsity-promoting recovery: Constant time-shifting
SNR = 4.80 dB



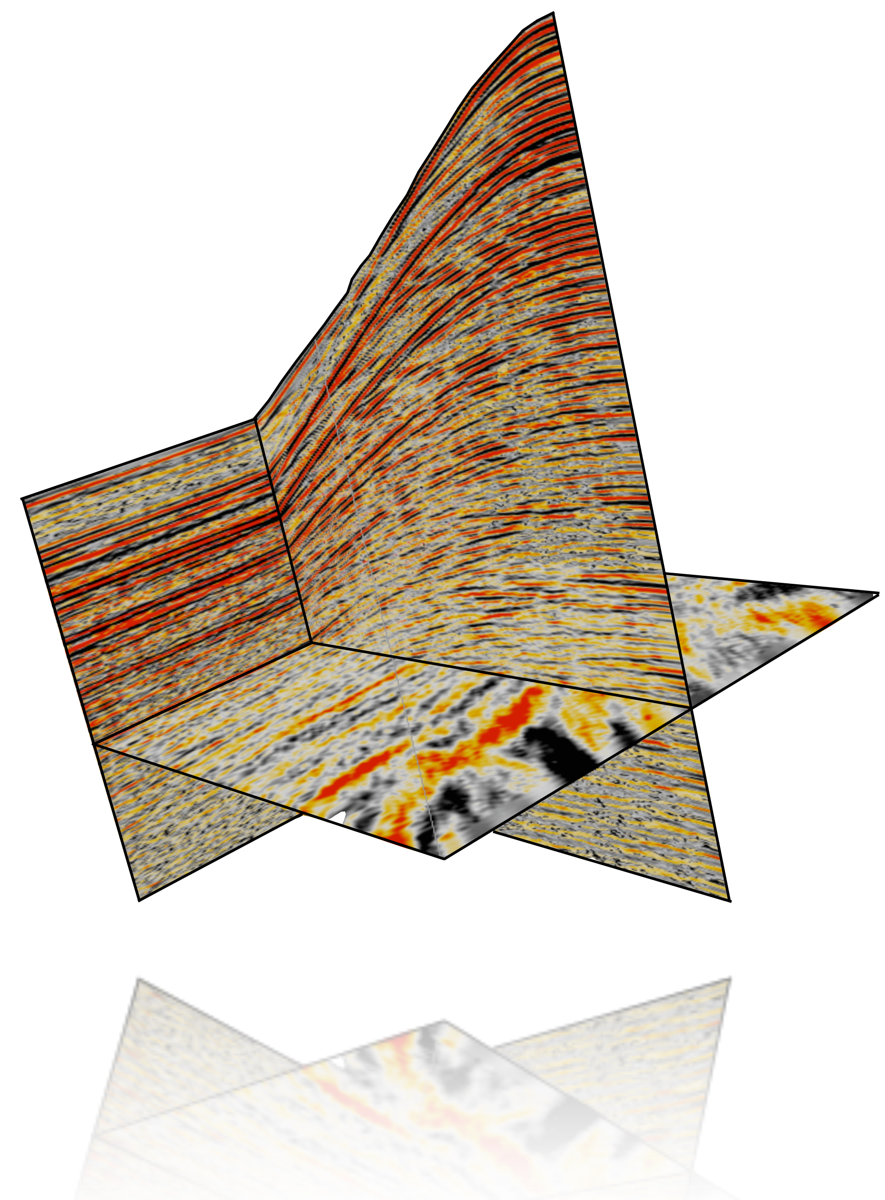
Key contributions

Practical acquisition design & recovery

- land and marine acquisitions
- curvelet-domain sparsity induction
- large-scale one-norm solvers
- highly suitable for OBC

Challenge: upscale to *full 3D*

Robust & dimensionality-reduced full-waveform *inversion*



Key goals

Reduce computational burden & memory imprint

- keep data in memory for each (GN) update

Improve imaging & *inversion* results by

- *exploiting* transform-domain sparsity
- incorporating *robustness* in the formulation

Key strategies

Exploit *structure* and break *coherences*

- *separable* structure
(*randomized* source *superposition* / *selection*, *stochastic* approximation, robust statistics)

- *multiscale* structure
(transform-domain *sparsity* & convex optimization)

- *convex-composite* structure
(compressive sensing)

Separable structure

FWI:

- is *linear* in the *sources*

$$\min_{\mathbf{m}} \phi(\mathbf{m}) = \sum_{i=1}^{K} \phi_i(\mathbf{m})$$

- *costs* are dominated by # of PDE solves = # of sources

Stochastic optimization

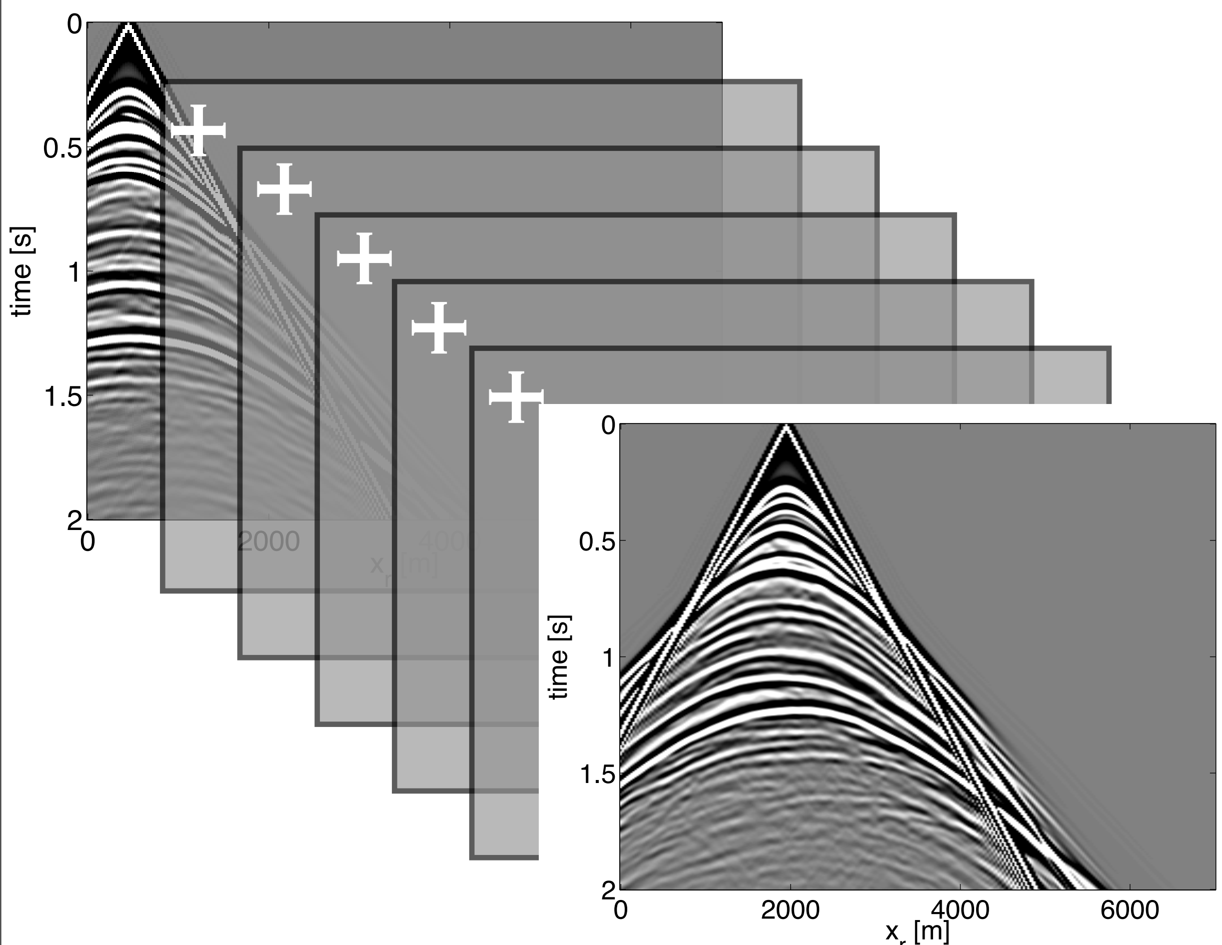
[Haber, Chung, and FJH, '10]
[van Leeuwen, Aravkin, FJH, '10]
[Haber, Chung, and FJH, '10]
[Bertsekas, '96, Nemirovsky, '08]

Exploit *separable* structure = *linearity* w.r.t. sources by

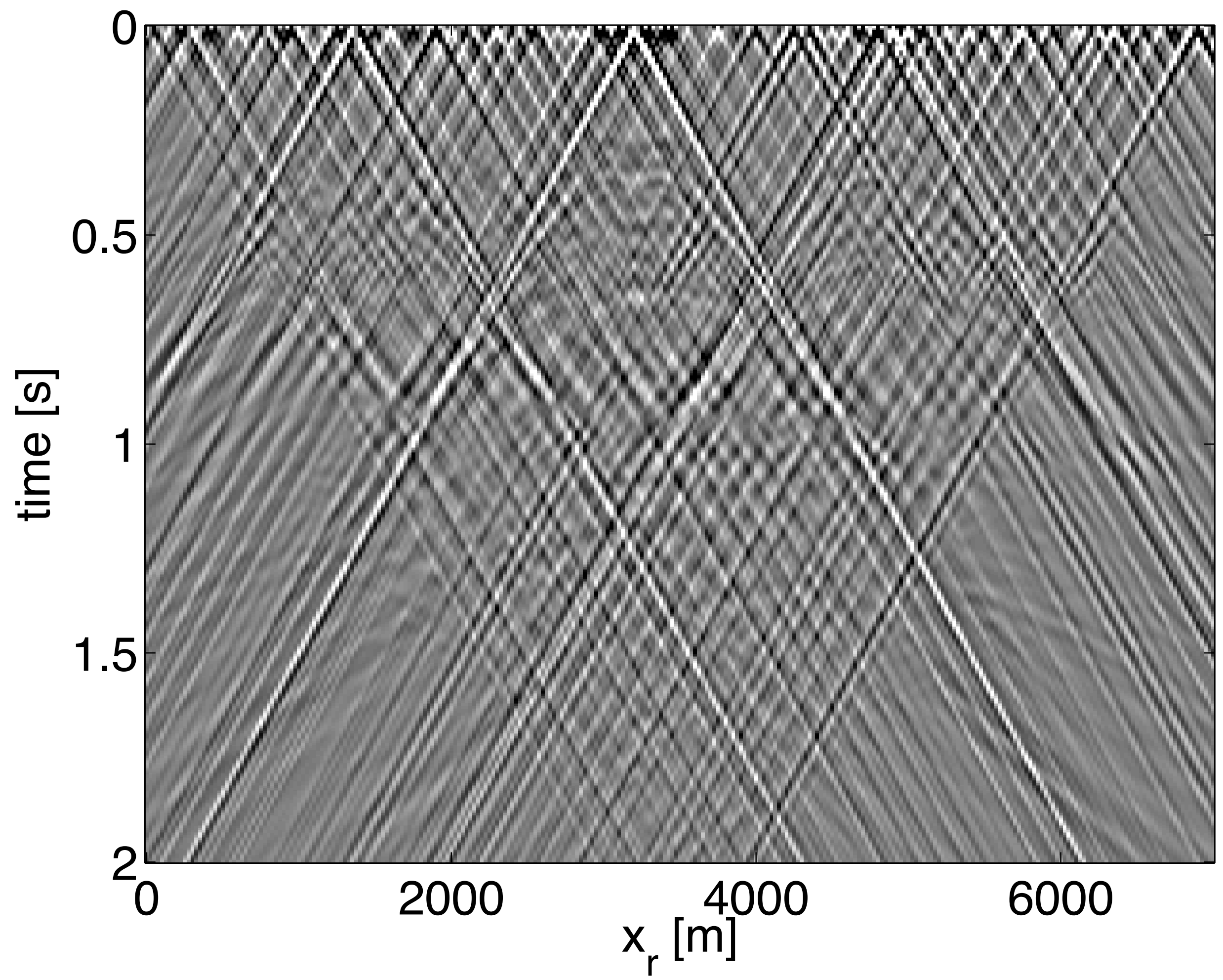
replacing deterministic FWI with *sums* cycling over *each source* & *corresponding shot record* (columns of D & Q):

$$\min_{\mathbf{m}} \phi(\mathbf{m}) = \frac{1}{N} \sum_{i=1}^{n_s} \frac{1}{2} \|\mathbf{d}_i - \mathcal{F}[\mathbf{m}; \mathbf{q}_i]\|_2^2$$

Randomized source encoding



=



randomized superposition

super shot

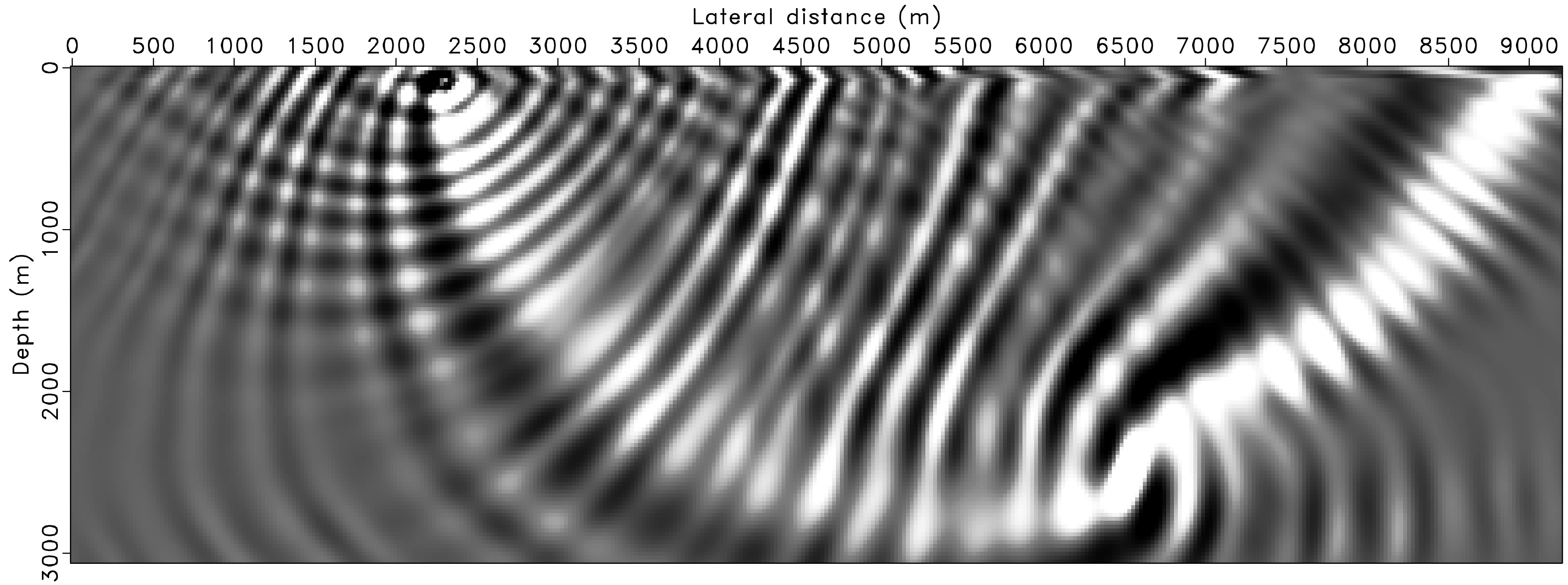
[Morton, '98, Romero, '00]

SLIM

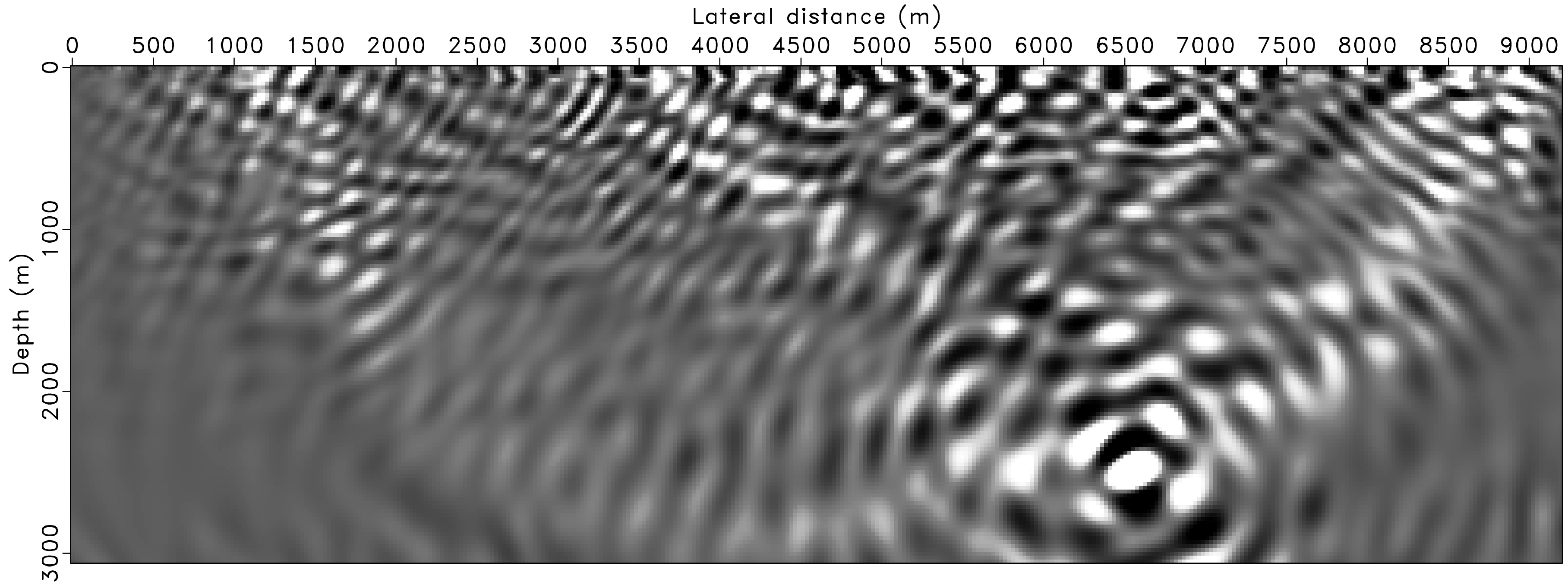
Gradient

[one shot]

Sequential-source image



Simultaneous-source image



Two strategies

Reduce *interferences* by *averaging* amongst

- (sim.) source experiments
  (stochastic-average approximation)
- model *iterates*
  (stochastic approximation)

or by

- transform-domain sparsity promotion
  (curvelet-domain one-norm minimization on updates)

Stochastic *average* approximation (SAA)

[Haber, Chung, & FJH, '10]

by *a stochastic-optimization* problem:

$$\begin{aligned}\min_{\mathbf{m}} \mathbf{E}_{\mathbf{w}}\{\phi(\mathbf{m},\mathbf{w}) &= \frac{1}{2}\|\mathbf{D}\mathbf{w} - \mathcal{F}[\mathbf{m};\mathbf{Q}\mathbf{w}]\|_2^2\} \\ &= \min_{\mathbf{m}} \phi(\mathbf{m}) \\ &\approx \min_{\mathbf{m}} \frac{1}{K}\sum_{j=1}^{K}\frac{1}{2}\|\underline{\mathbf{d}}_j - \mathcal{F}[\mathbf{m};\underline{\mathbf{q}}_j]\|_2^2\end{aligned}$$

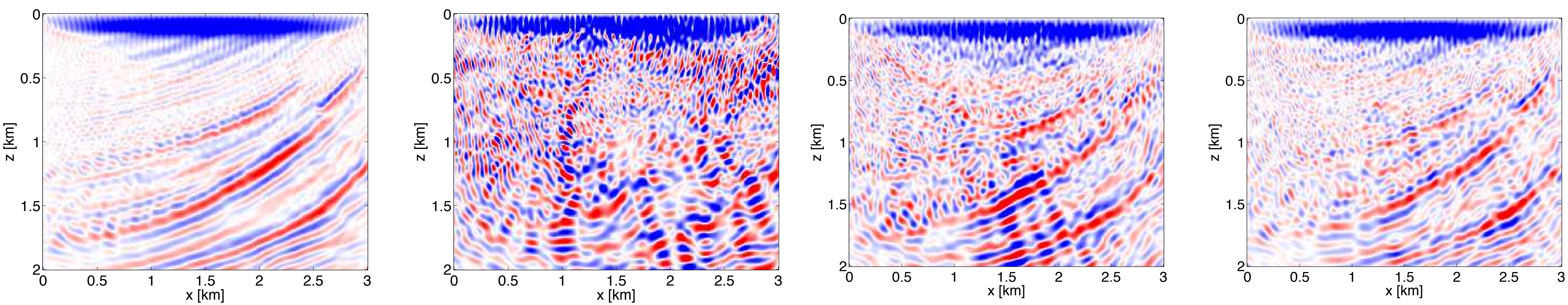
with $\mathbf{E}_{\mathbf{w}}\{\mathbf{w}\mathbf{w}^H\} = \mathbf{I}$

and $\underline{\mathbf{d}}_j = \mathbf{D}\mathbf{w}_j,\ \underline{\mathbf{q}}_j = \mathbf{Q}\mathbf{w}_j$

Gradients

Search direction for *increasing* batch size K:

$$\mathbf{g}_K = \frac{1}{K}\sum_{j=1}^{K} \nabla\mathcal{F}^*[\mathbf{m};\underline{\mathbf{q}}_j]\delta\underline{\mathbf{d}}_j$$



full K=1 K=5 K=10

Stochastic approximation (SA)

Algorithm 1: Stochastic gradient descent

Result: Output estimate for the model $\mathbf{m}$

$\mathbf{m} \longleftarrow \mathbf{m}_0$; $k \longleftarrow 0$; `// initial model`

while not converged **do**

$\{\underline{\mathbf{d}}^k, \underline{\mathbf{q}}^k\} \longleftarrow \{\mathbf{D}\mathbf{w}^k, \mathbf{Q}\mathbf{w}^k\}$ with $\mathbf{w}^k \in N(0,1)$; `// draw sim. exp.`

$\mathbf{g}^k \longleftarrow \nabla\mathcal{F}^*[\mathbf{m}^{k-1}, \underline{\mathbf{q}}^k](\underline{\mathbf{d}}^k - \mathcal{F}[\mathbf{m}^{k-1}, \underline{\mathbf{q}}^k])$; `// gradient`

$\underline{\mathbf{m}}^{k+1} \longleftarrow \mathbf{m}^k - \gamma^k \mathbf{g}^k$; `// update`

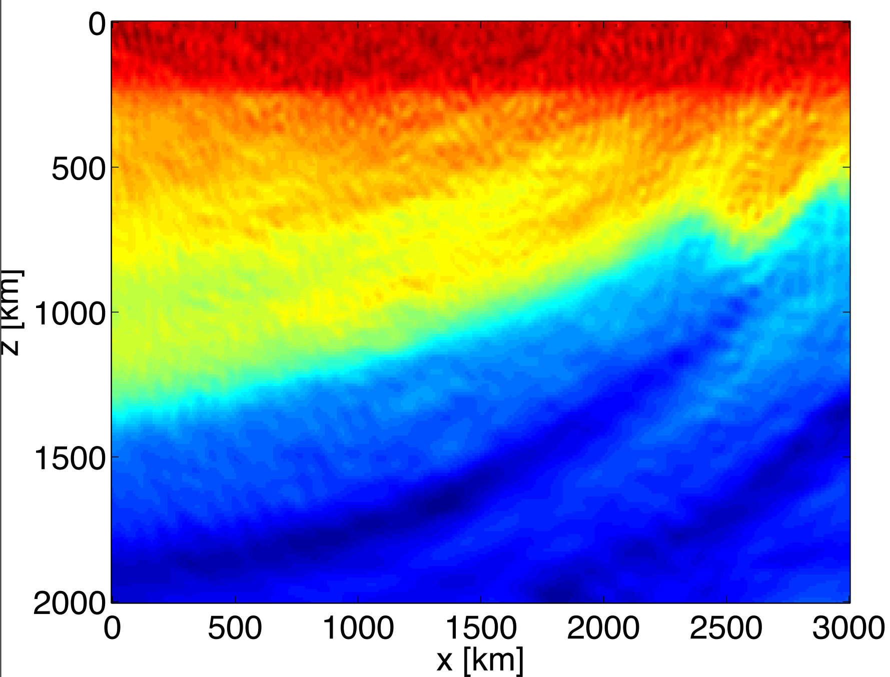
$\mathbf{m}^{k+1} = \frac{1}{k+1}\left(\sum_{i=1}^k \mathbf{m}^i + \underline{\mathbf{m}}^{k+1}\right)$; `// average`

$k \longleftarrow k+1$;

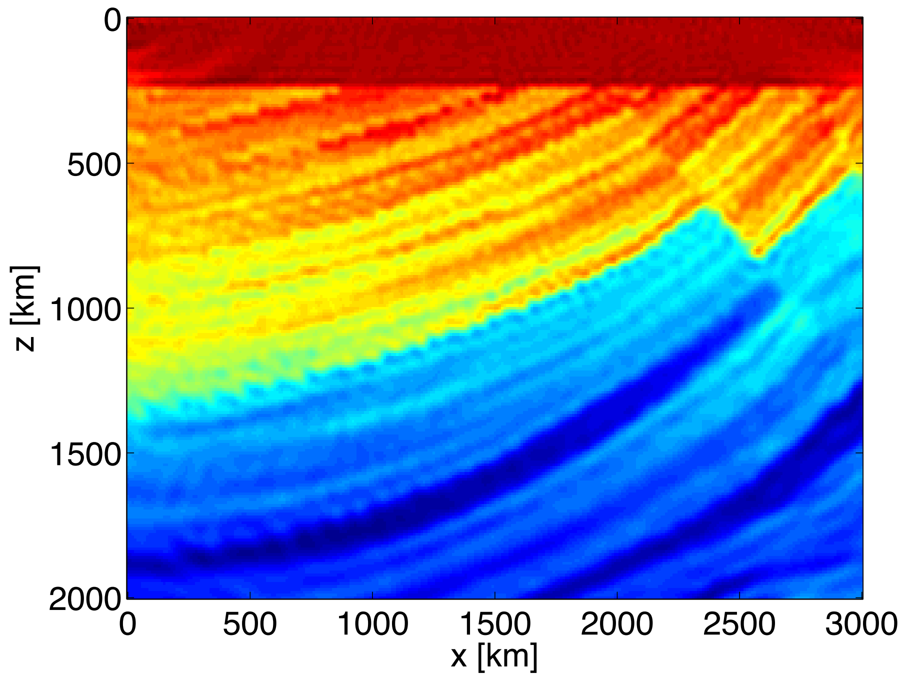
end

[Bertsekas, '96; Haber, Chung, and FJH, '10]

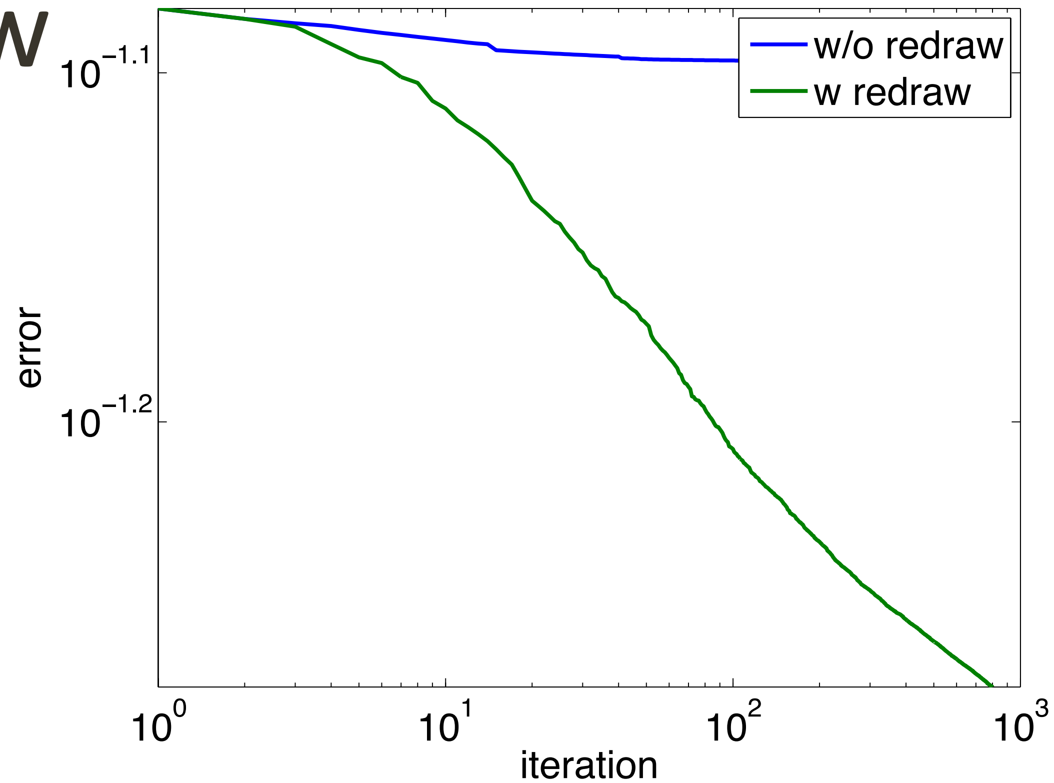
K=1 w and w/o redraw [noise-free case]



w/o redraw



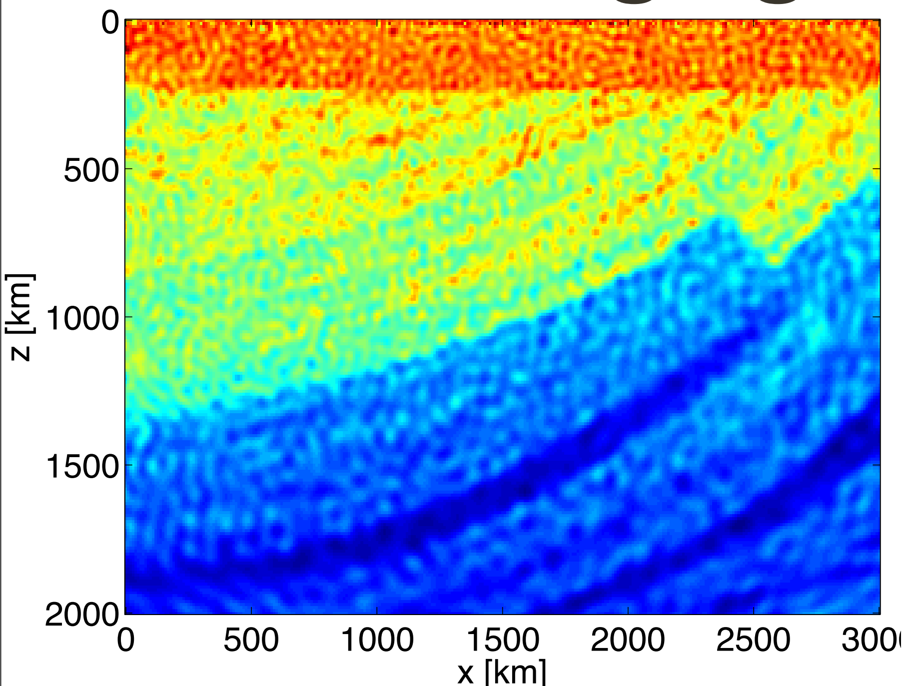
w redraw



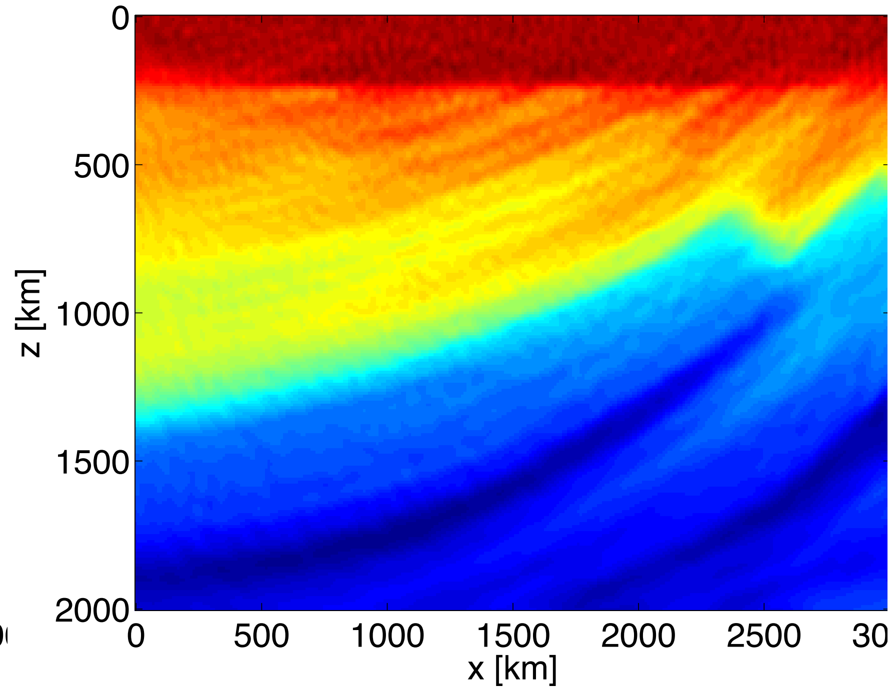
model error K=1

K=1w redraw [noisy case]

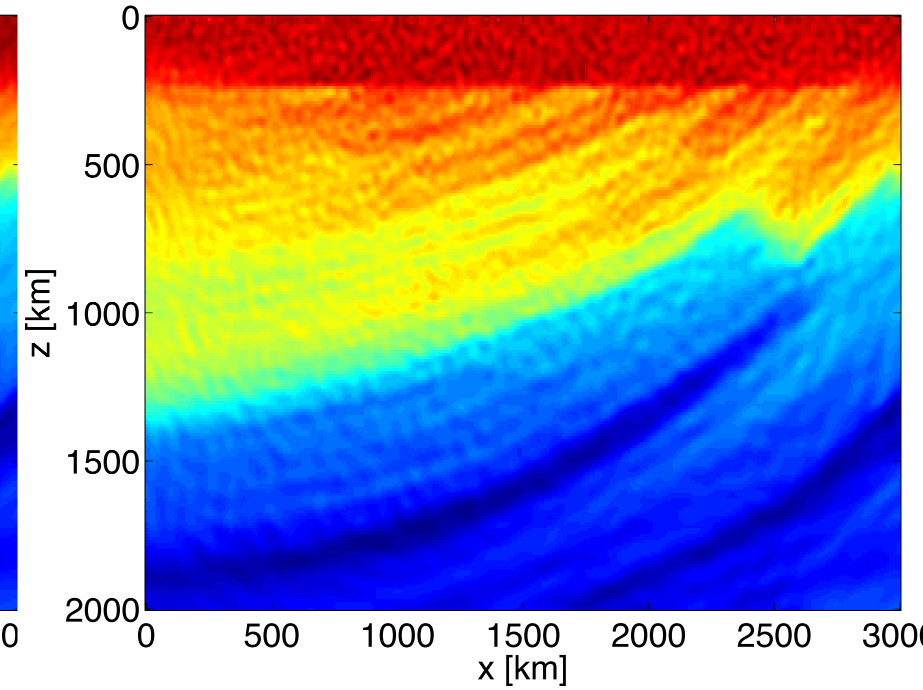
w/o averaging

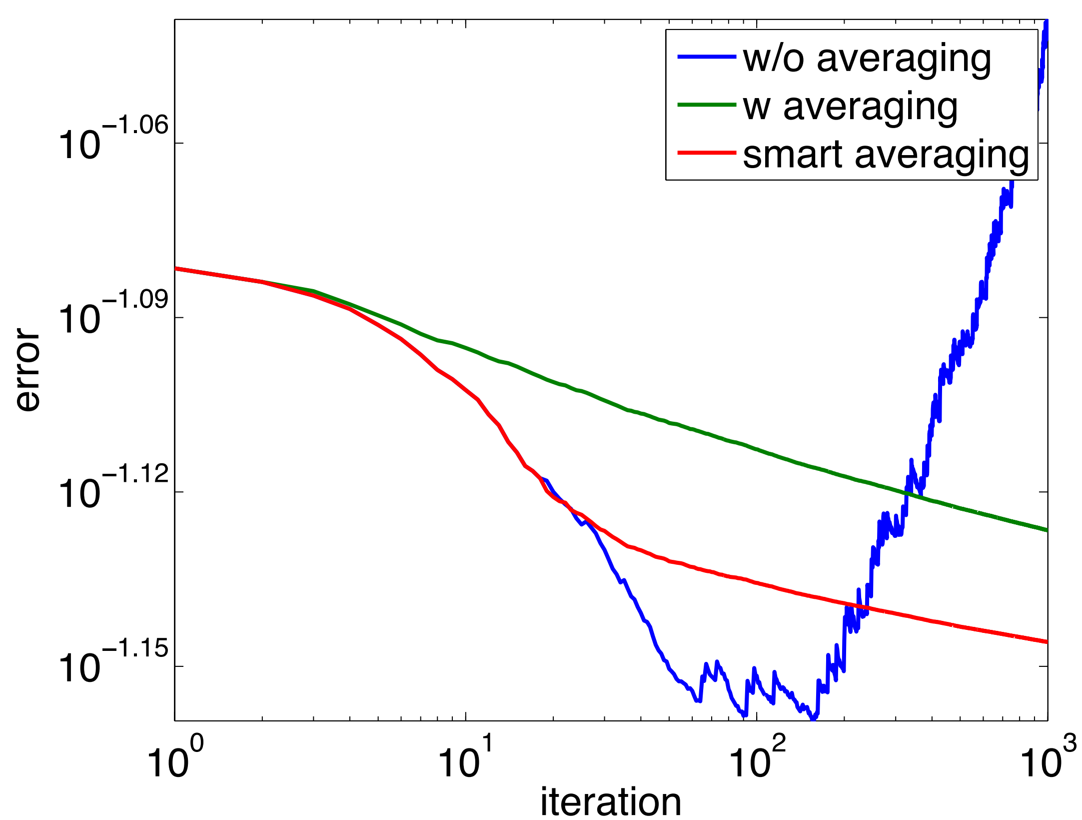


w averaging



smart averaging





SLIM

Observations

Stochastic-average approximation (**SAA**):

- Error due to *crosstalk* decays slowly with batch size K

Stochastic approximation (**SA**):

- Renewals improve convergence *significantly*
- Requires *averaging* to remove *crosstalk* & noise *instability*, which is *detrimental* to *convergence*

Both methods rely on *averaging* to mitigate *crosstalk*. Are there *better* alternatives?

Contributions

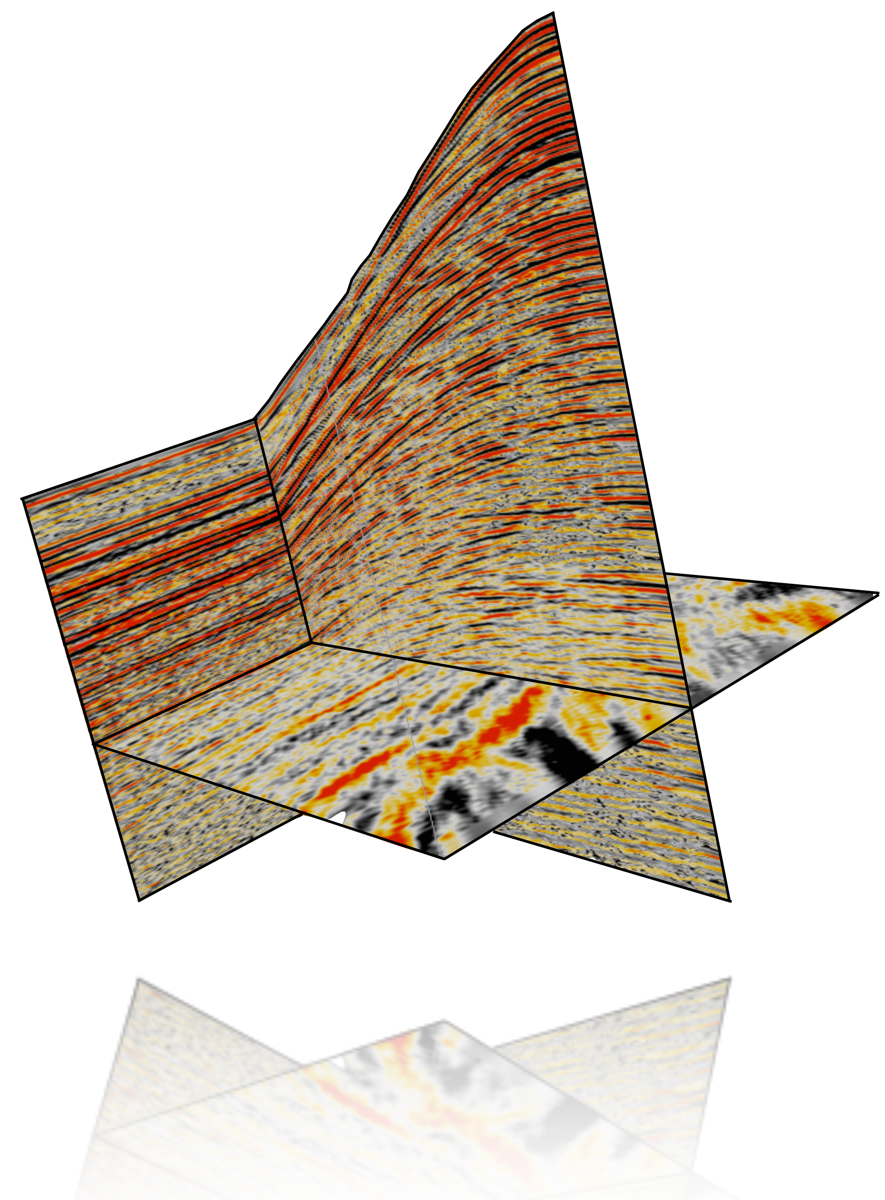
Control of the *errors* by increasing the *batch* size

- by moving from *stochastic* to *deterministic* optimization
- works with *randomized* sequential or simultaneous source experiments

Add *robustness*

- by using student t misfit functional
- works with *inaccurate* forward models

Hybrid stochastic-deterministic optimization

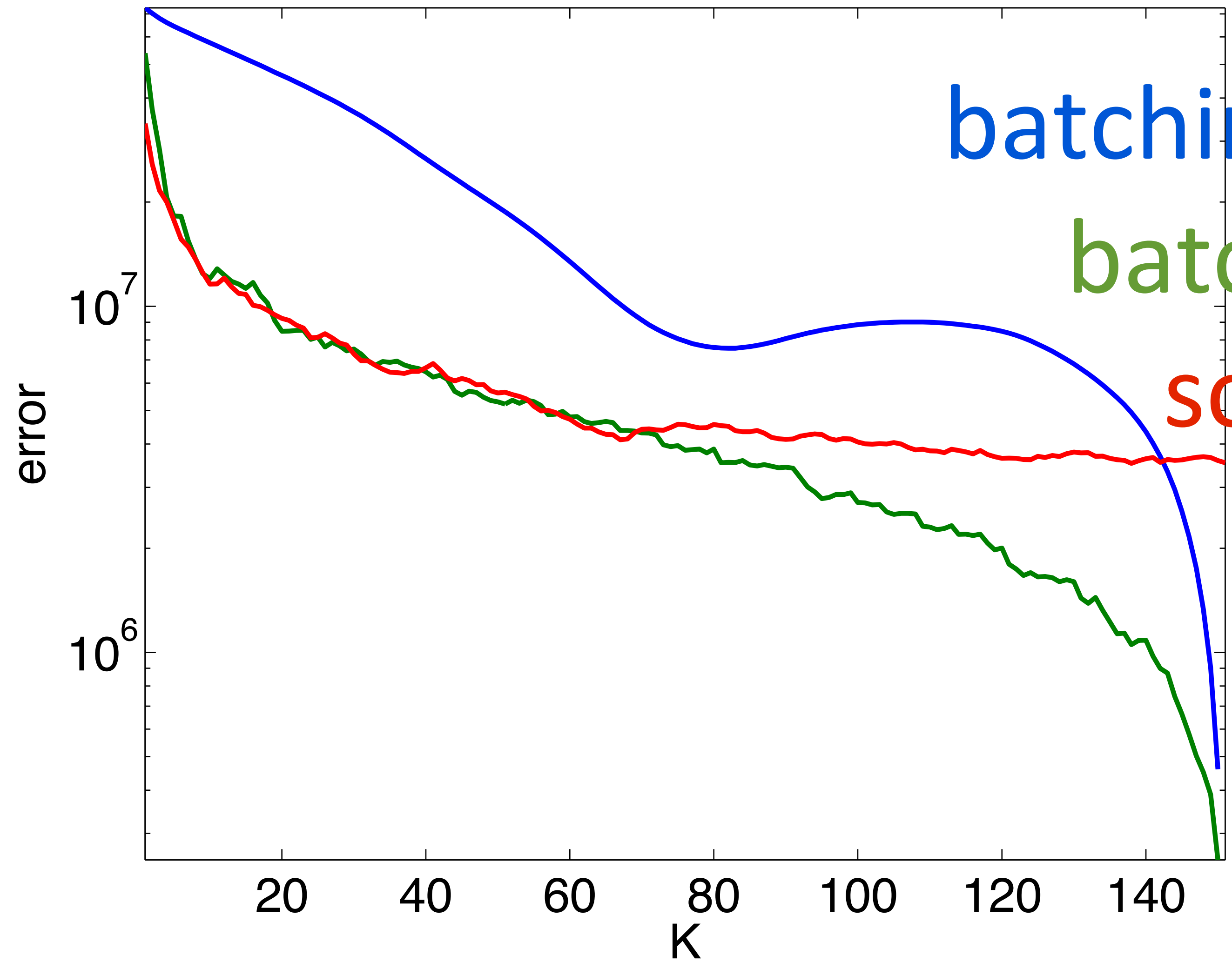


Fast FWI w/o encoding

- Work with small subset of randomly chosen shots at each iteration
- slowly increase number of shots
- ...

Fast FWI w/o encoding

- Error in the gradient determines convergence

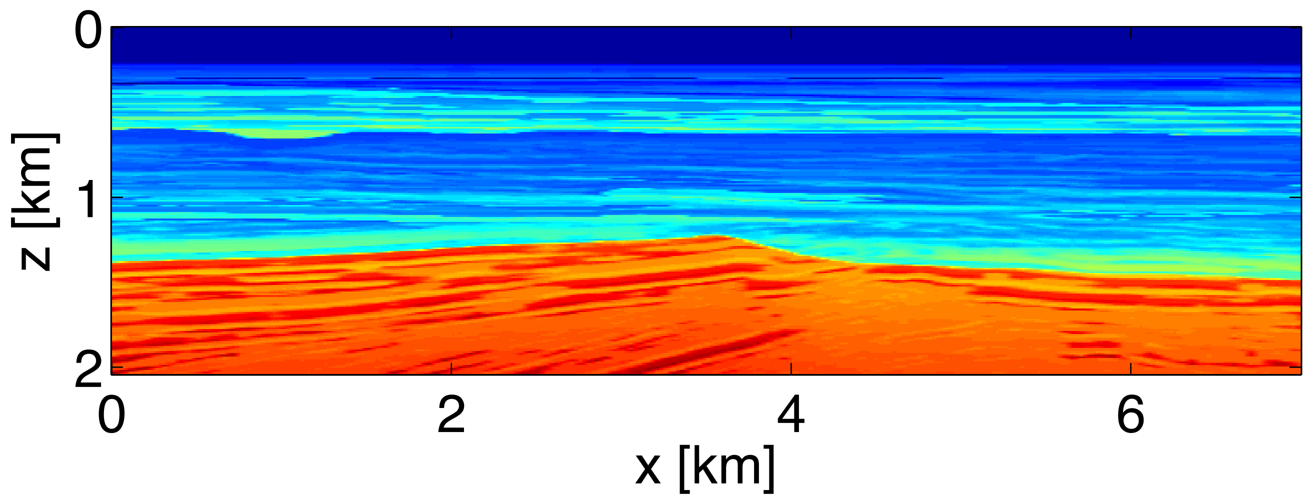


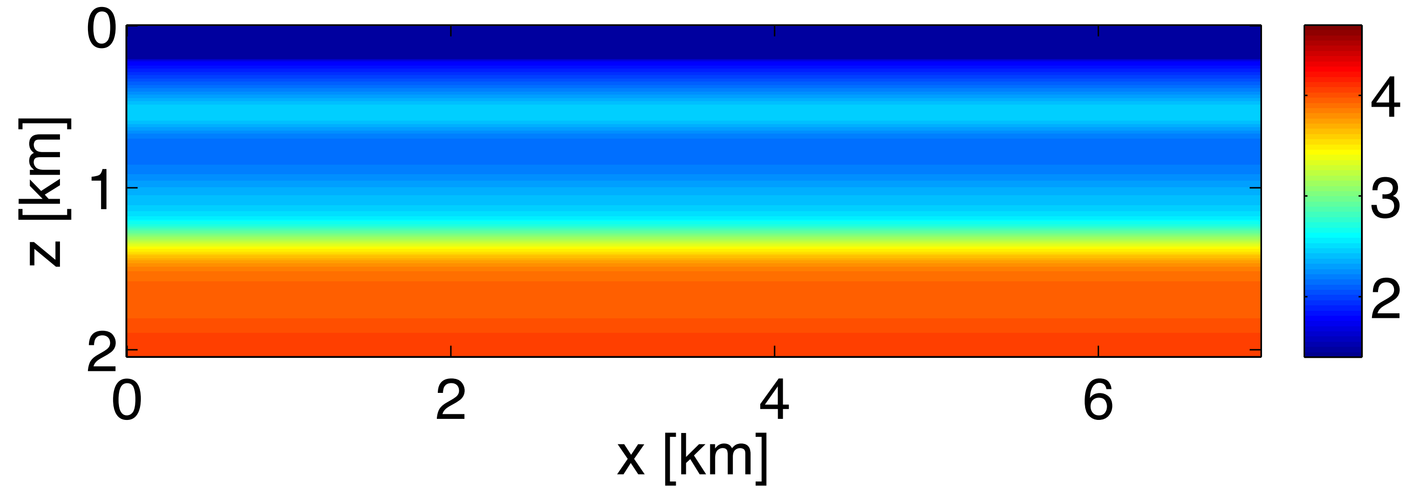
batching (worst case)

batching (random)

source encoding

Full waveform inversion



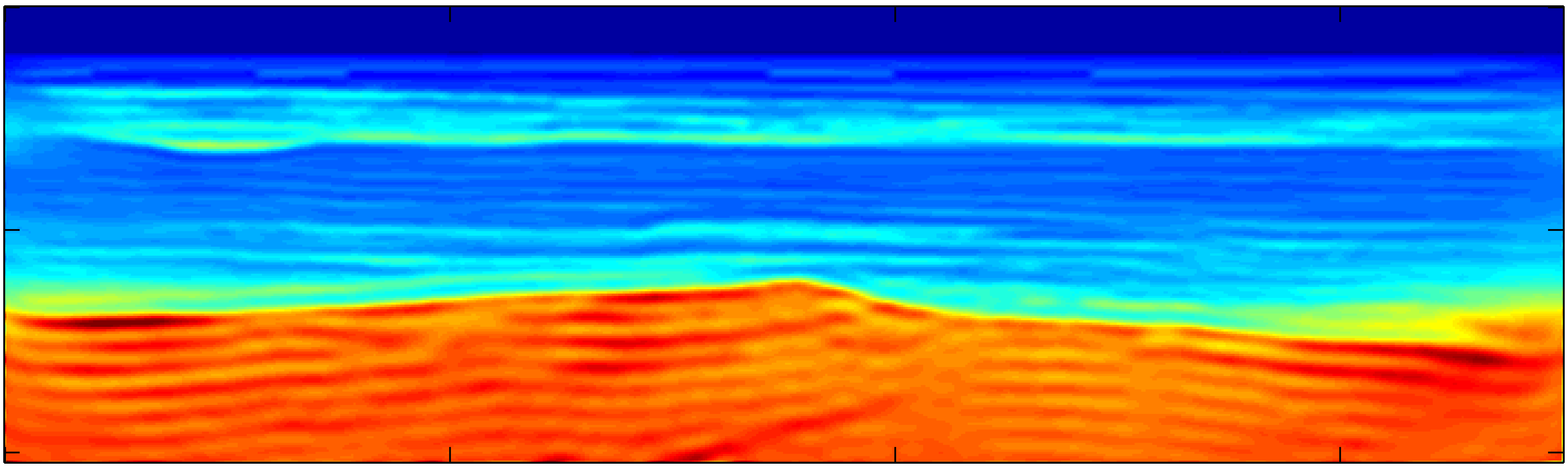


data for
141 sources, 281
receivers, 15 Hz Ricker

multi-scale frequency
domain inversion:
[2.5-20] Hz in 16 bands

[Bunks `95; Pratt `98]

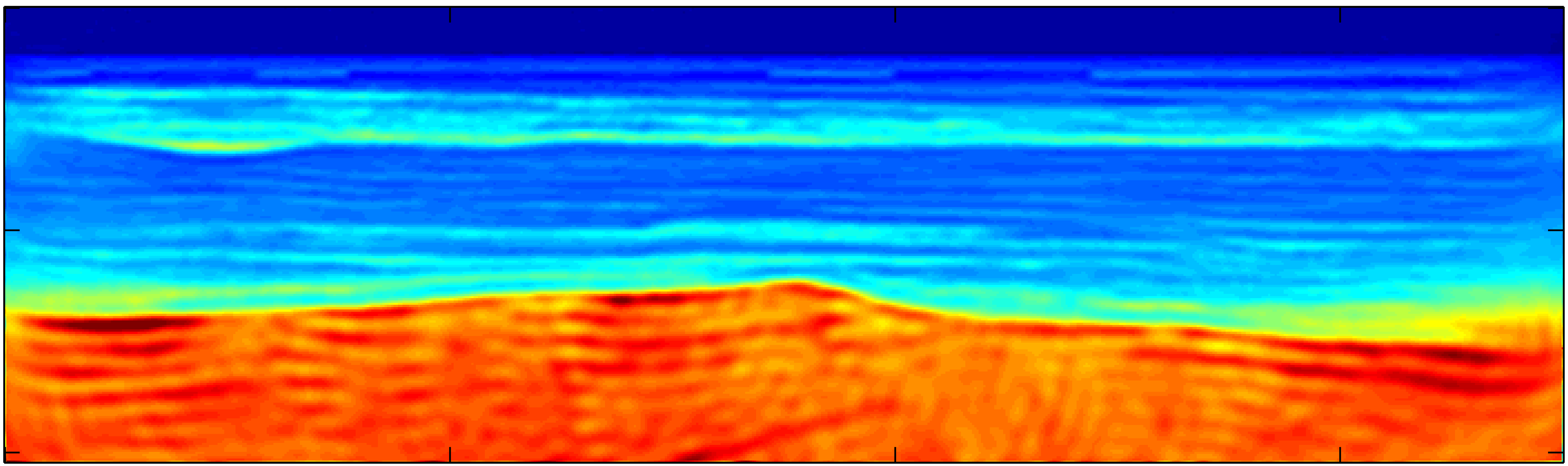
FWI



traditional L-BFGS

~10 full evaluations per frequency band

FWI



hybrid method

~2 full evaluations per frequency band

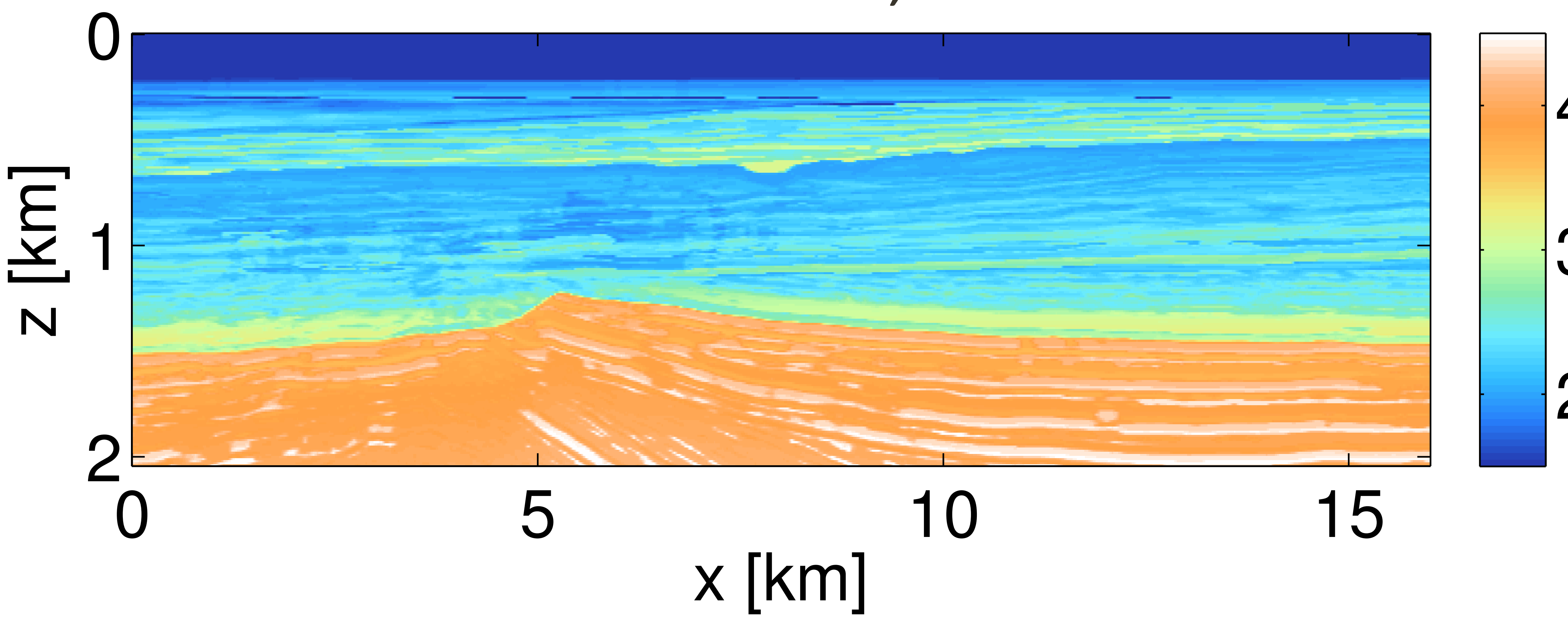
SLIM

FWI 2

time domain data

min offset 100m, max offset 3 km

320 sources at 50m, 15 Hz Ricker



FWI 2

Estimate source wavelet:

$$\Phi[\mathbf{m}, \mathbf{a}] = ||a_i F[\mathbf{m}]\mathbf{q}_i - \mathbf{d}_i||_2^2$$

LS solution for $\mathbf{a}$:

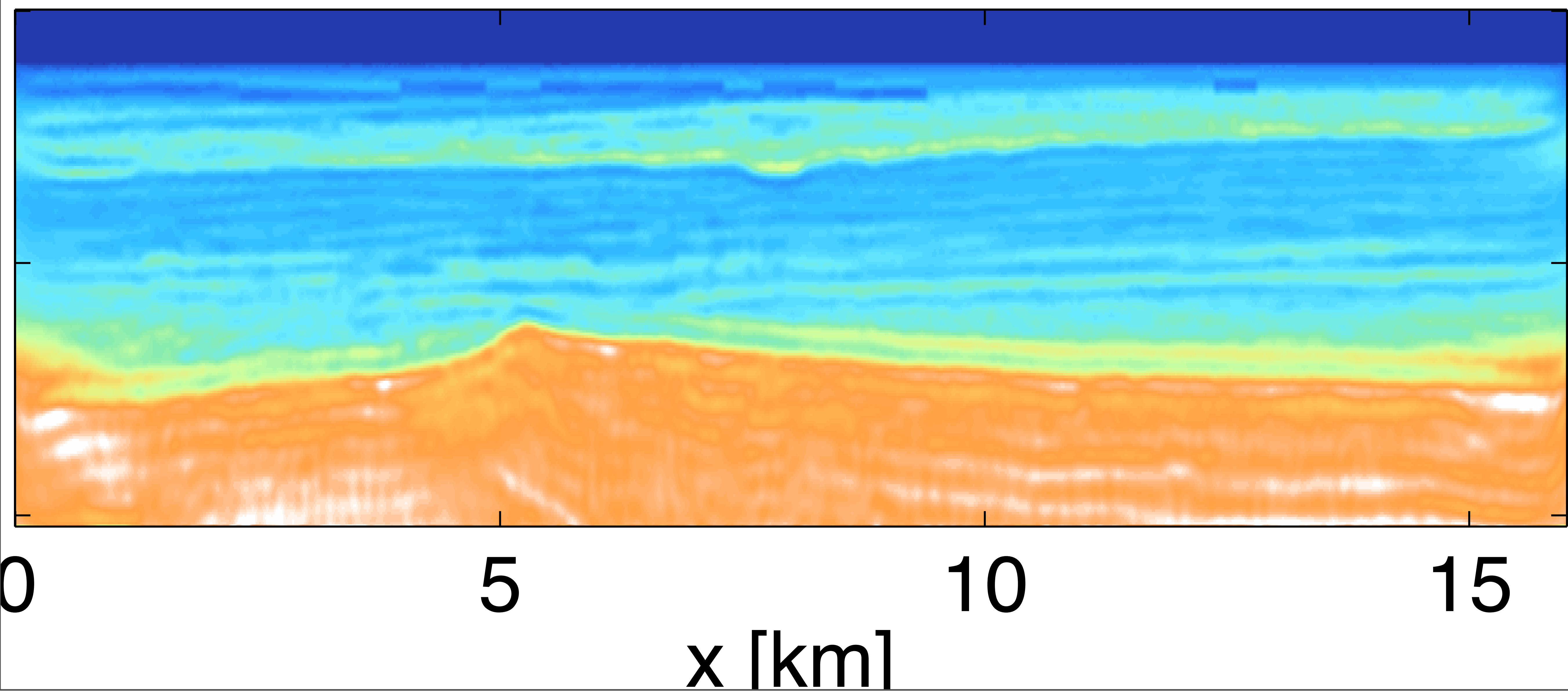
$$\hat{a}_i = \frac{(F[\mathbf{m}]\mathbf{q}_i)^H \mathbf{d}_i}{||\mathbf{d}_i||_2^2}$$

then:

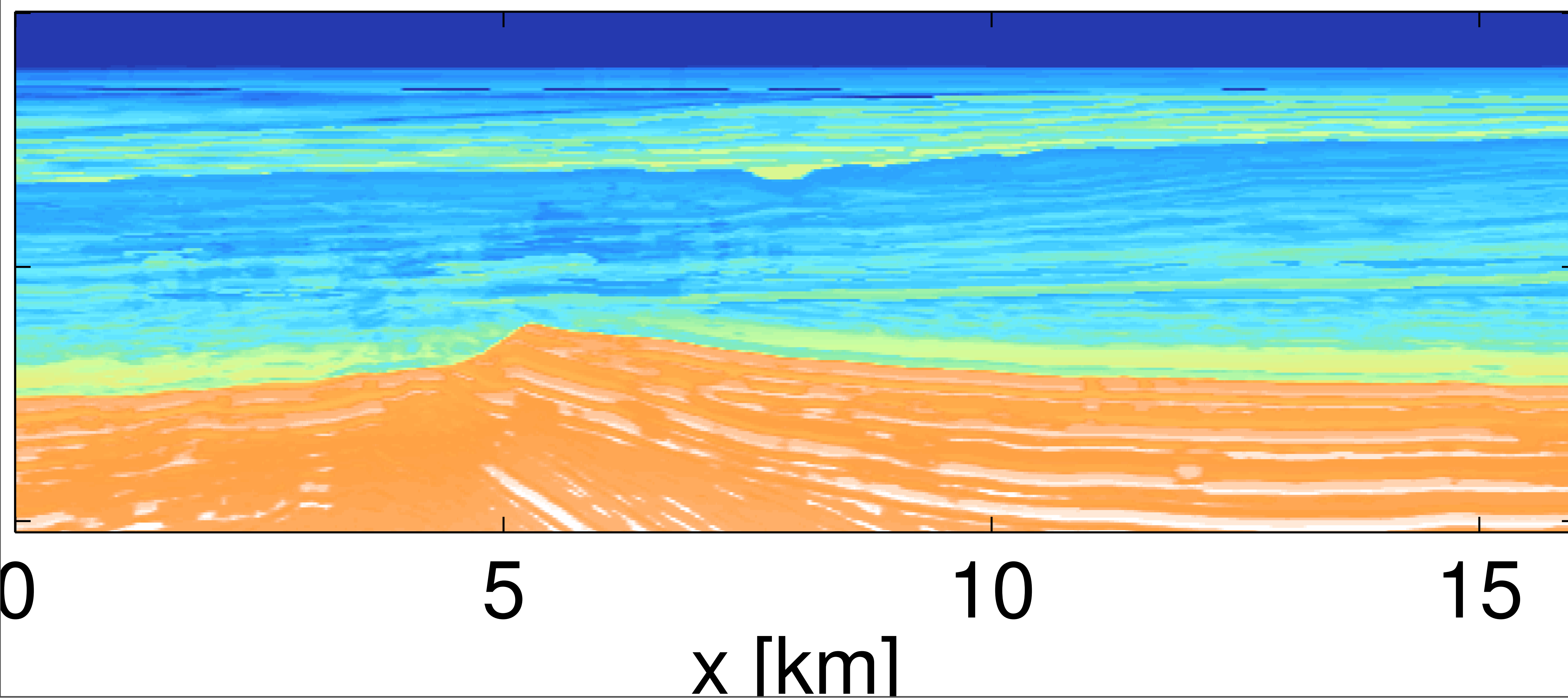
$$\nabla\Phi[\mathbf{m}, \hat{\mathbf{a}}] = \left(\frac{\partial a_i F[\mathbf{m}]\mathbf{q}_i}{\partial \mathbf{m}}\right)^H (a_i F[\mathbf{m}]\mathbf{q}_i - \mathbf{d}_i)$$

FWI 2

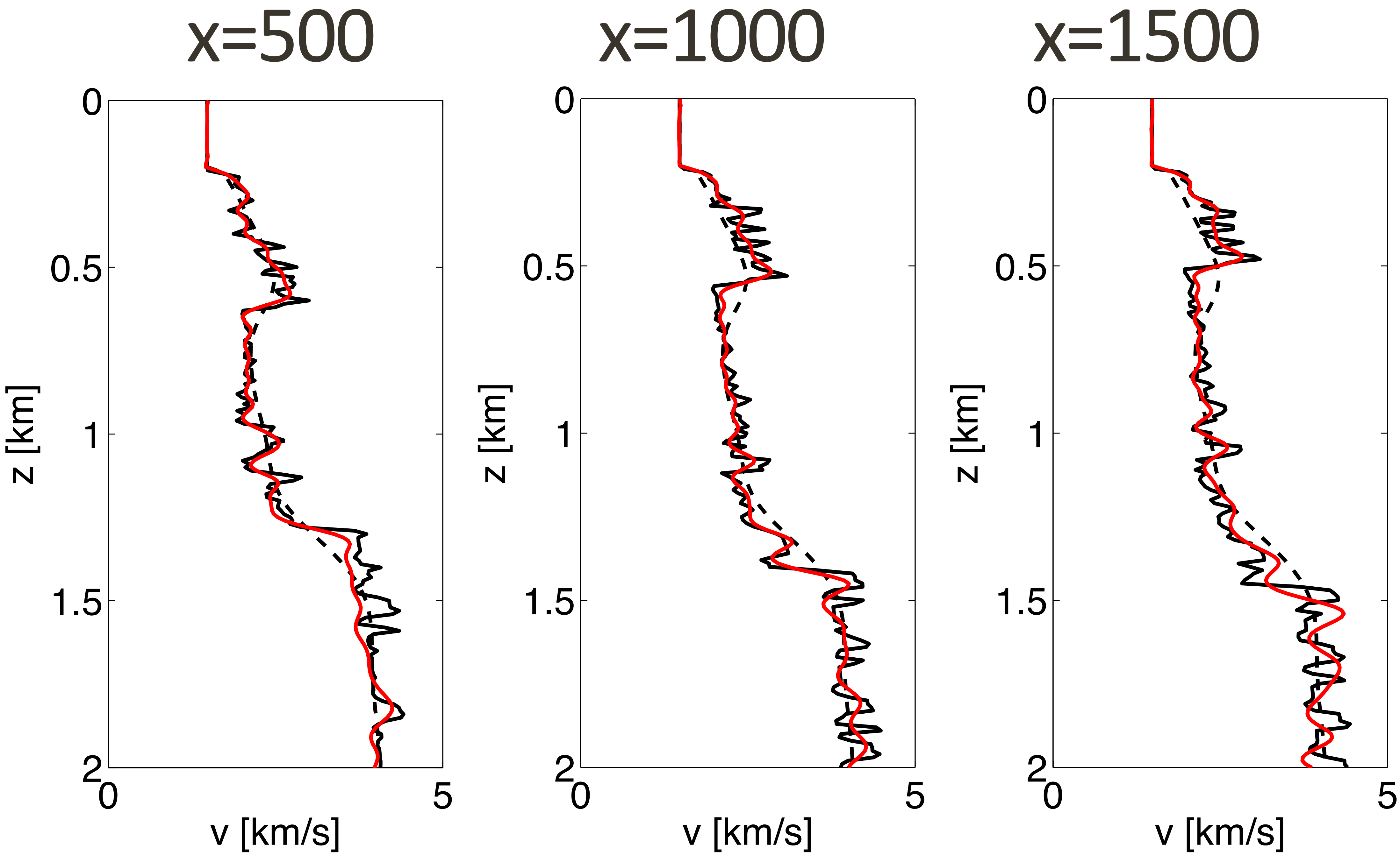
2 passes through the data for each freq. band



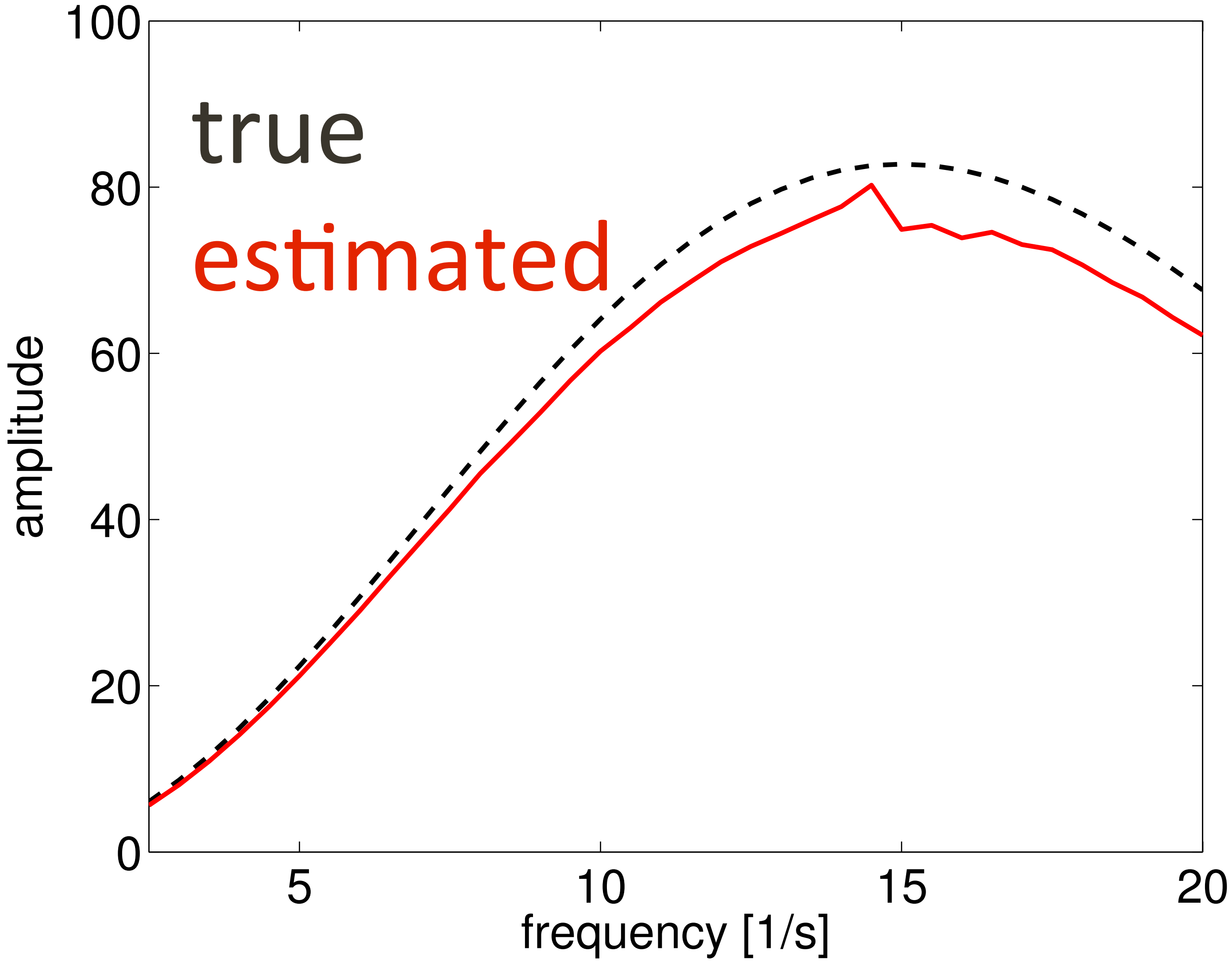
FWI 2



FWI 2



FWI 2


100
80
60
40
20
0
5
10
15
20
frequency [1/s]
amplitude
true
estimated

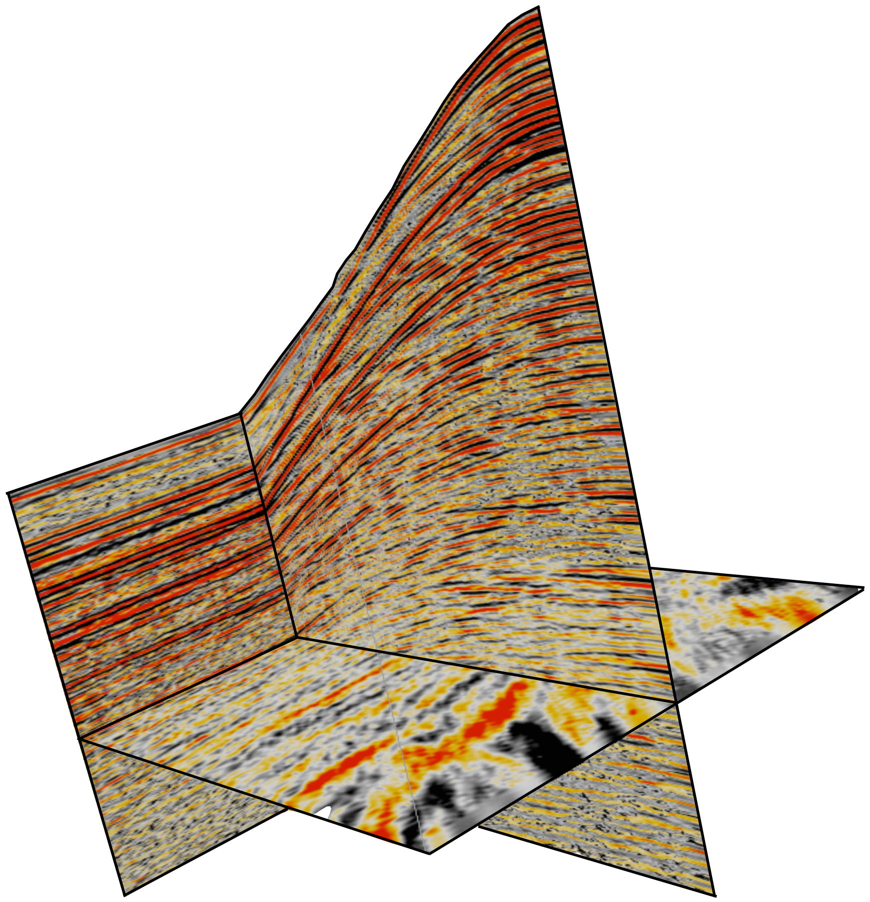
Fast FWI

- work with small subsets of data at each iteration
- makes more sophisticated approaches feasible

Robust FWI

- LS approach very sensitive to noise or unexplained artifacts in the data
- Use `robust’ penalty

Fast robust FWI



From the Statistics to the Optimization

- Begin with assumptions on the model error

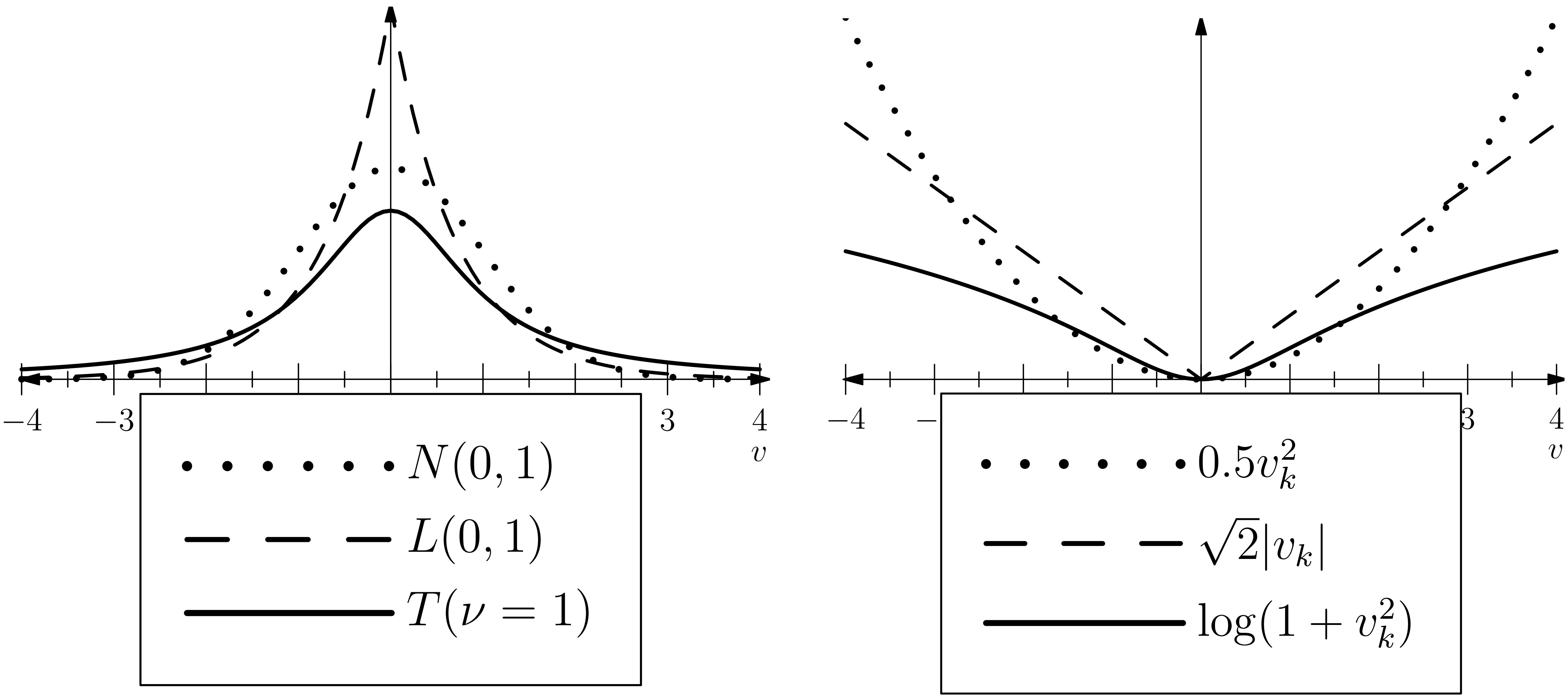
$$\begin{aligned} \mathbf{D} &= \mathcal{F}[\mathbf{x}; \mathbf{Q}] + \boldsymbol{\epsilon} \\ \boldsymbol{\epsilon} &\sim \text{Heavy Tailed Distribution with density } \mathbf{p} \end{aligned}$$

- Compute the Maximum a Posteriori (MAP) estimate for **p:**

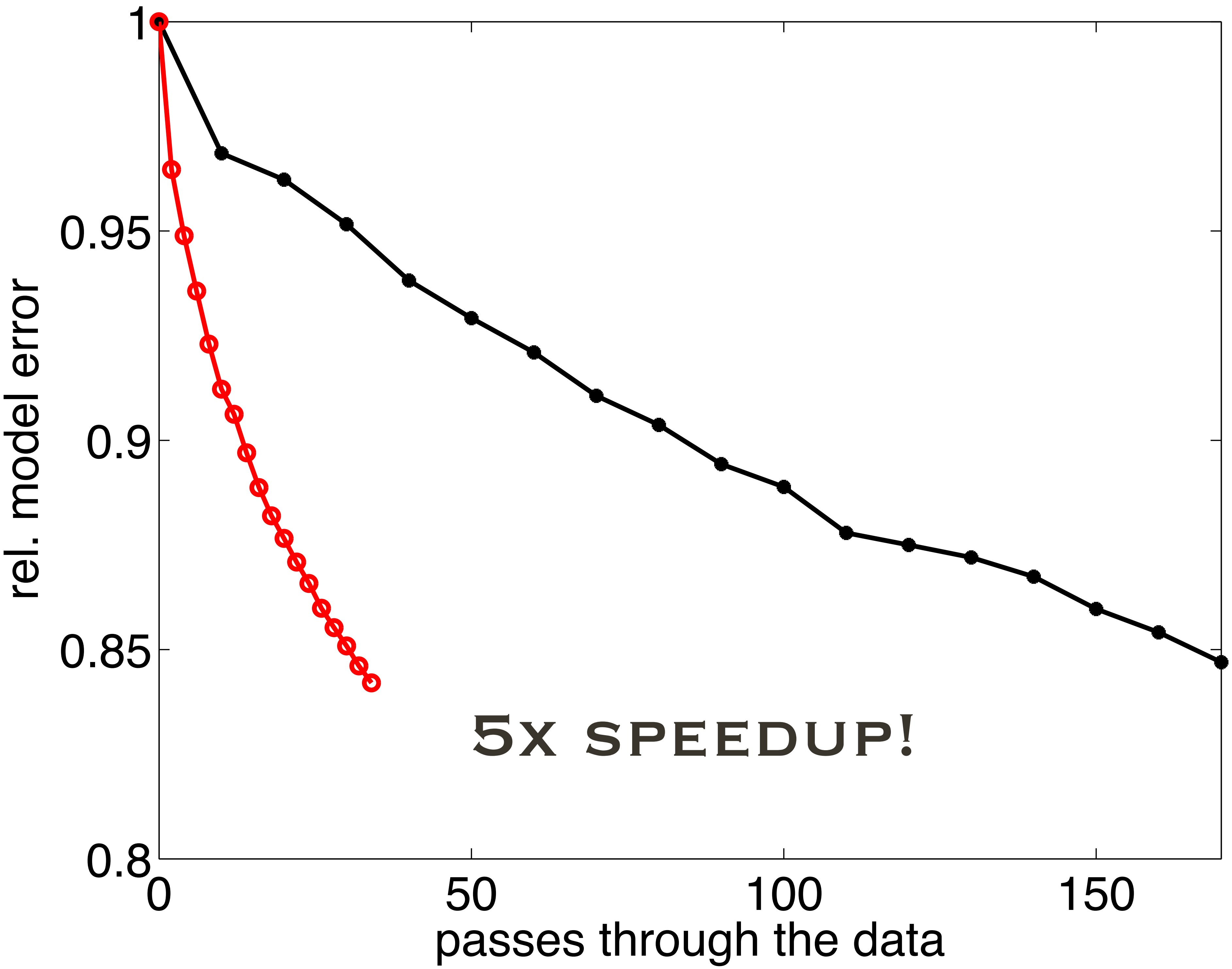
$$\min_{\mathbf{x}} \phi(\mathbf{x}) := -\log\left[\mathbf{p}\Big(\mathbf{D} - \mathcal{F}[\mathbf{x}; \mathbf{Q}]\Big)\right]$$

- Note if you start with Gaussian errors, you get LS formulation.

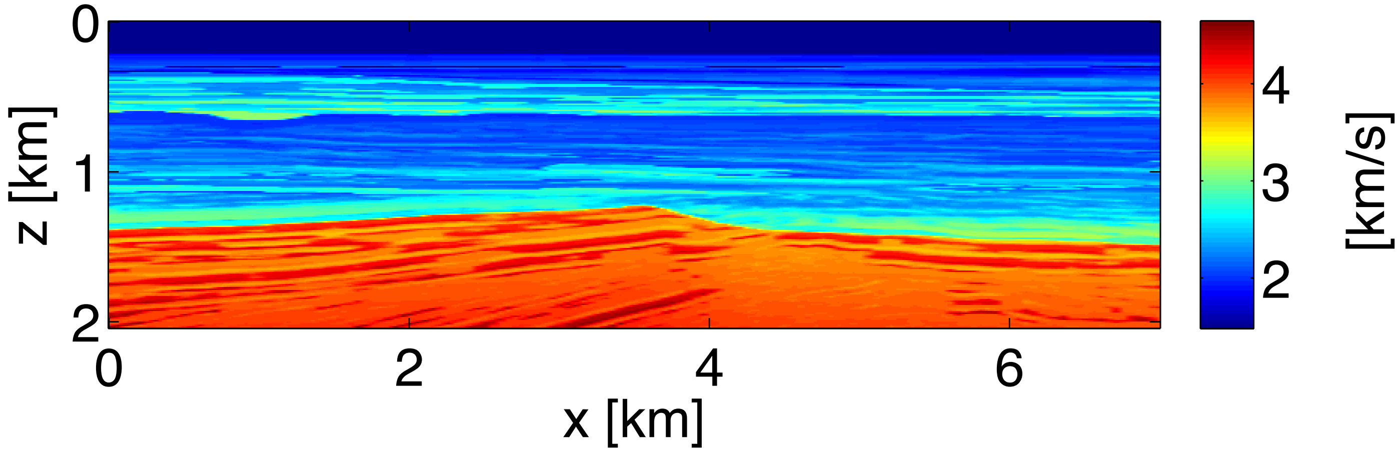
Densities and Penalties



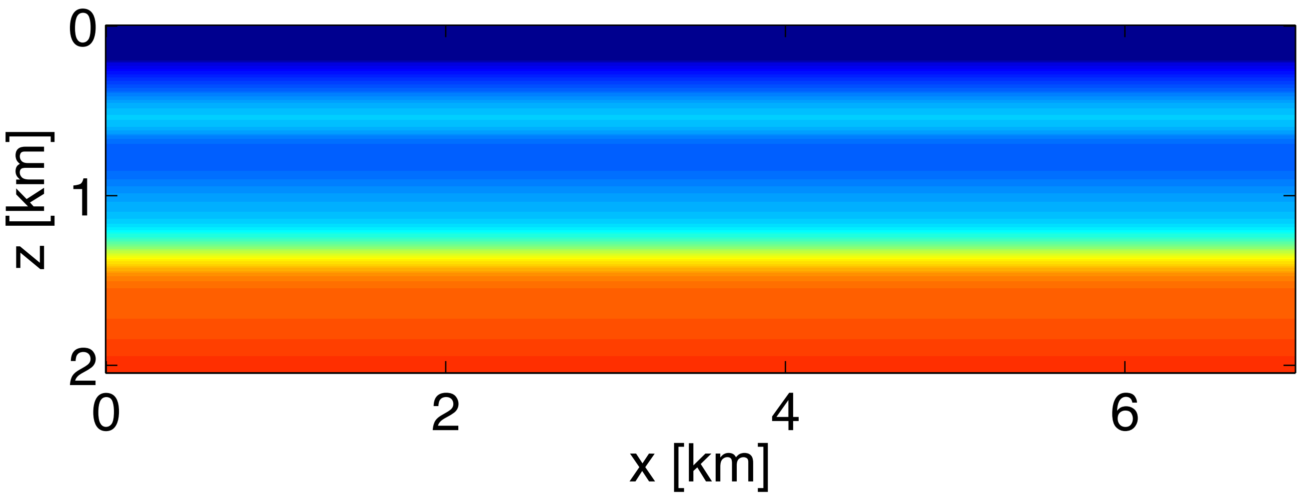
Speedup: Semistochastic vs. Direct



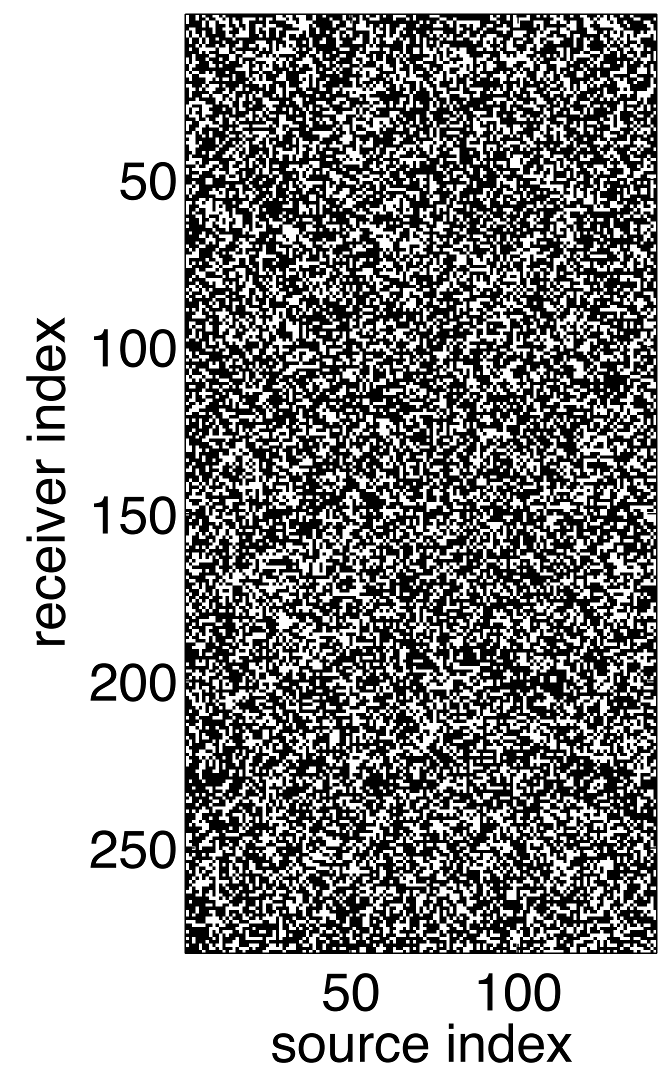
Synthetic Example II: Missing Data



BG Compass Model (Truth)

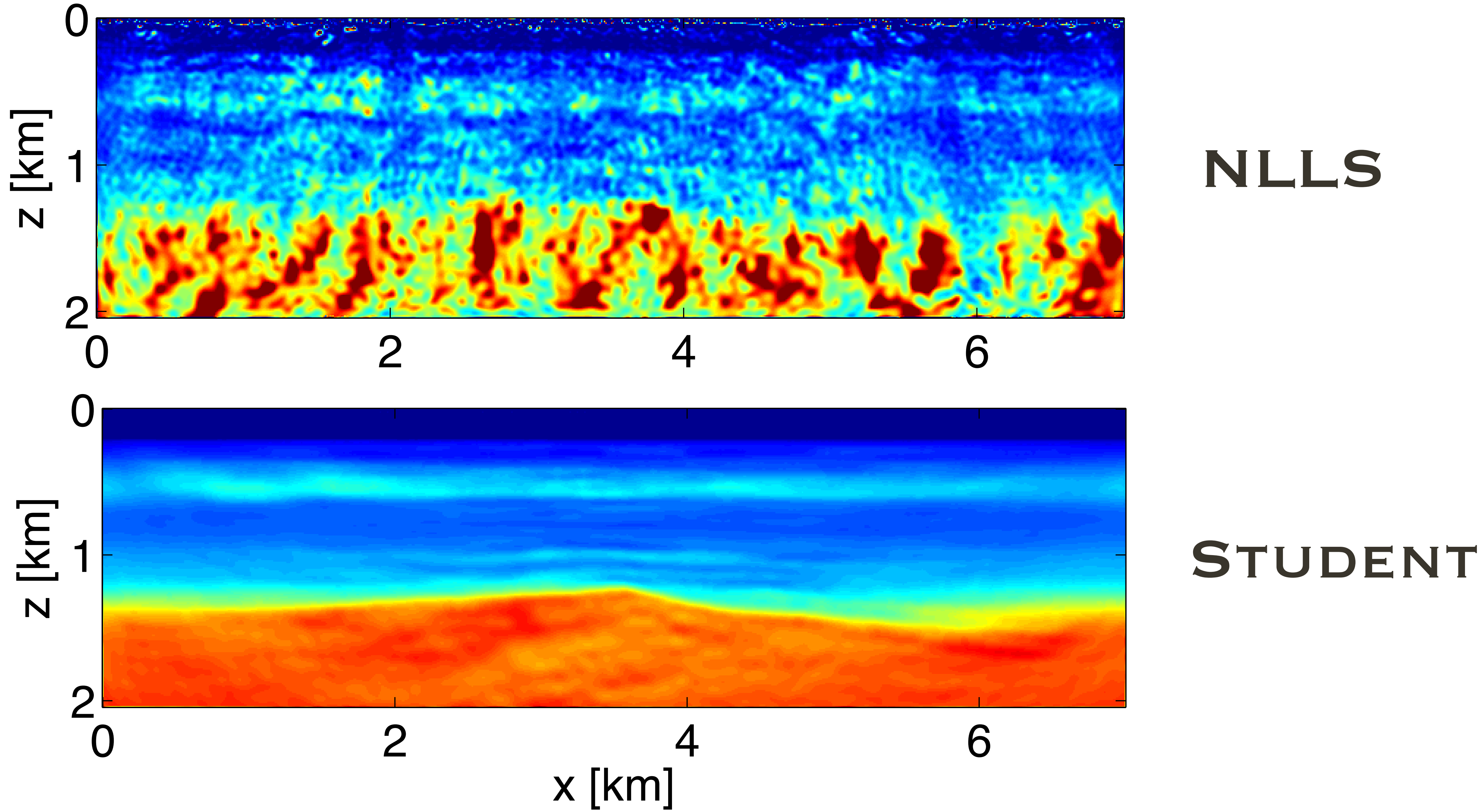


Initial model

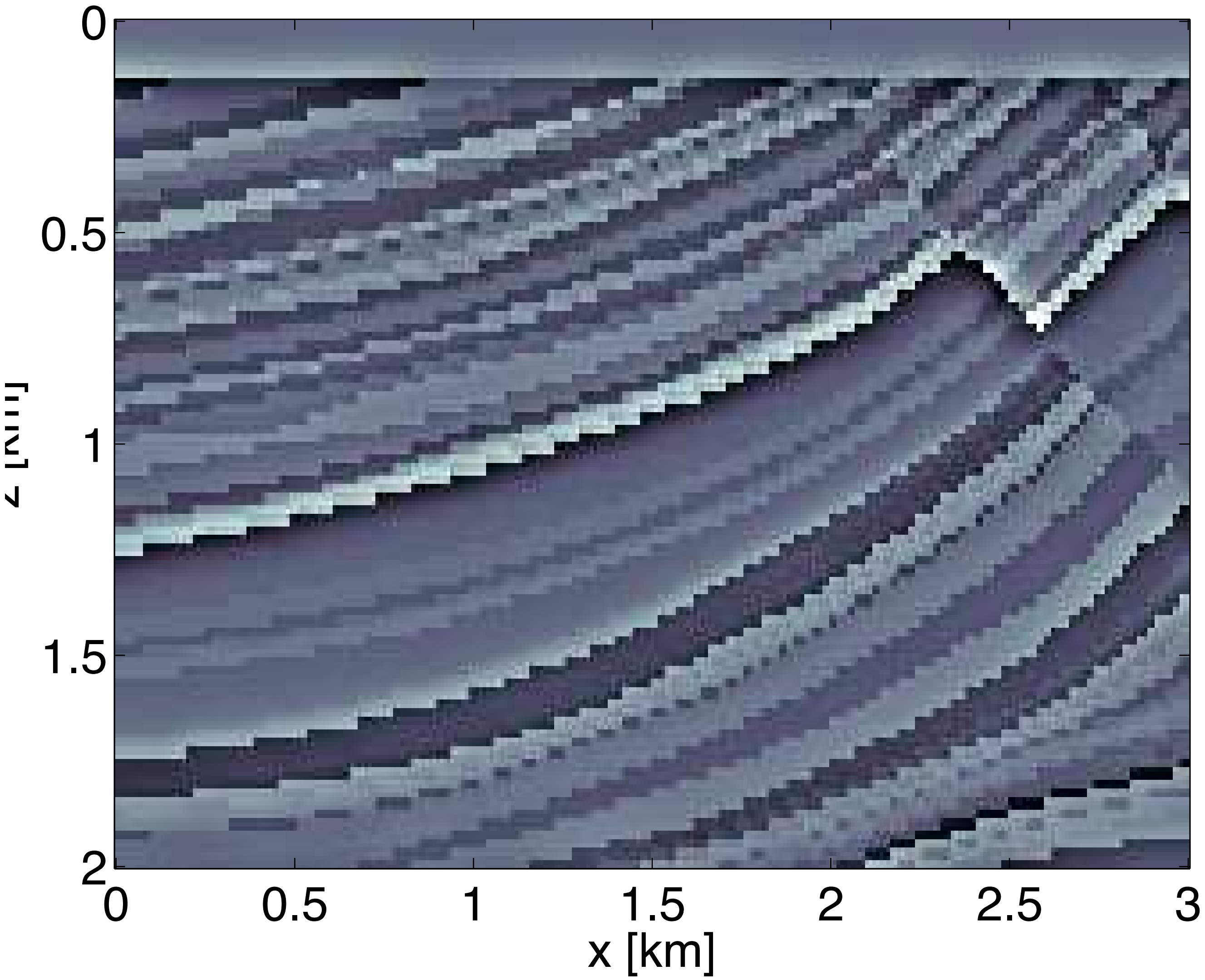


60 % missing data!

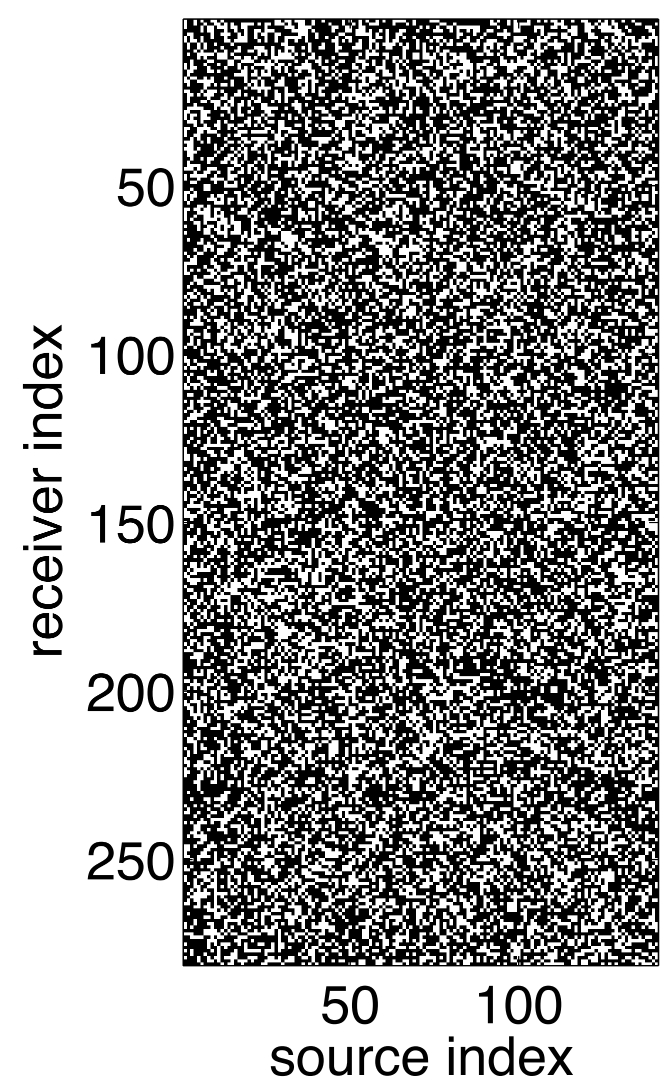
Inversion Results



Robust FWI

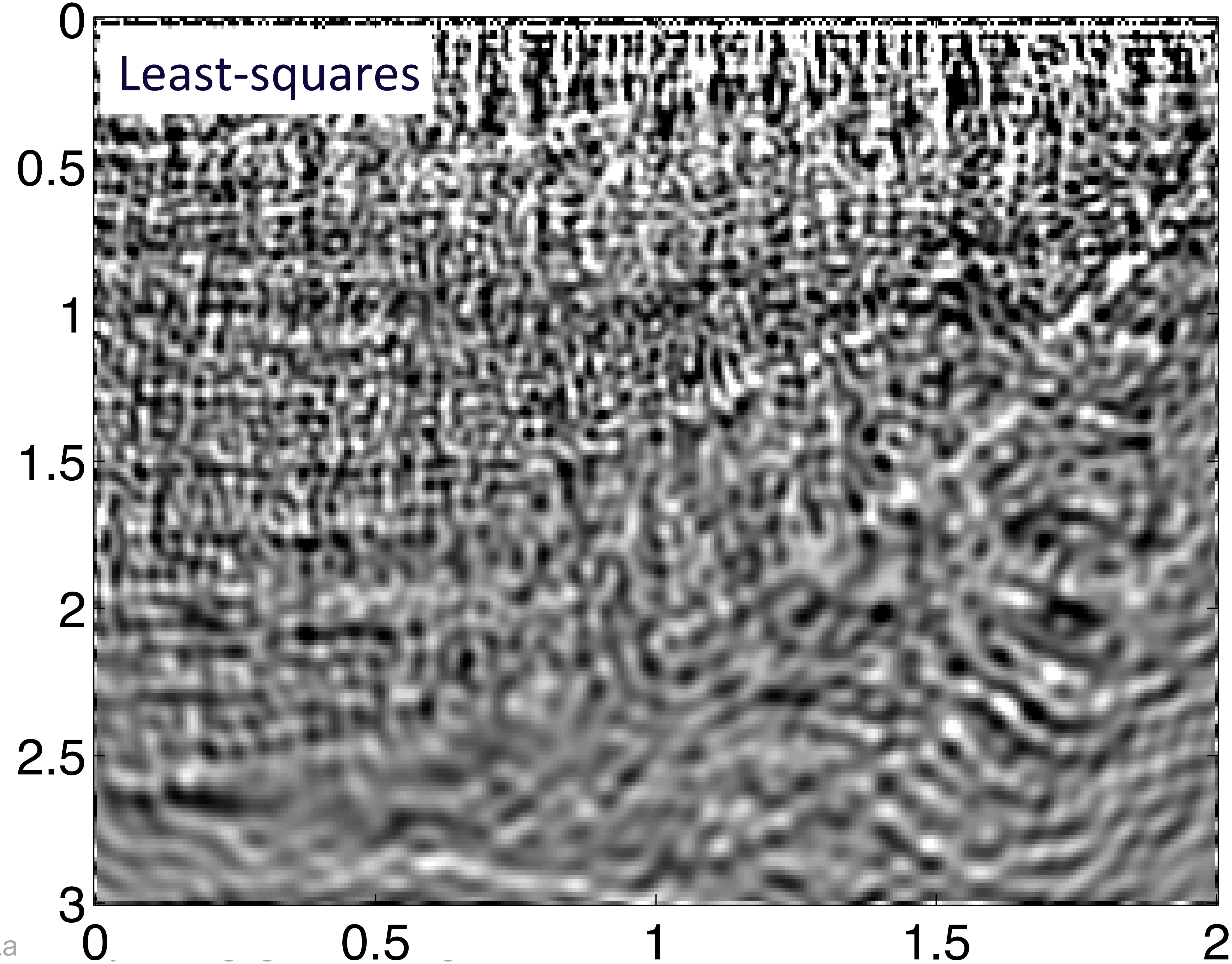


True Reflectivity

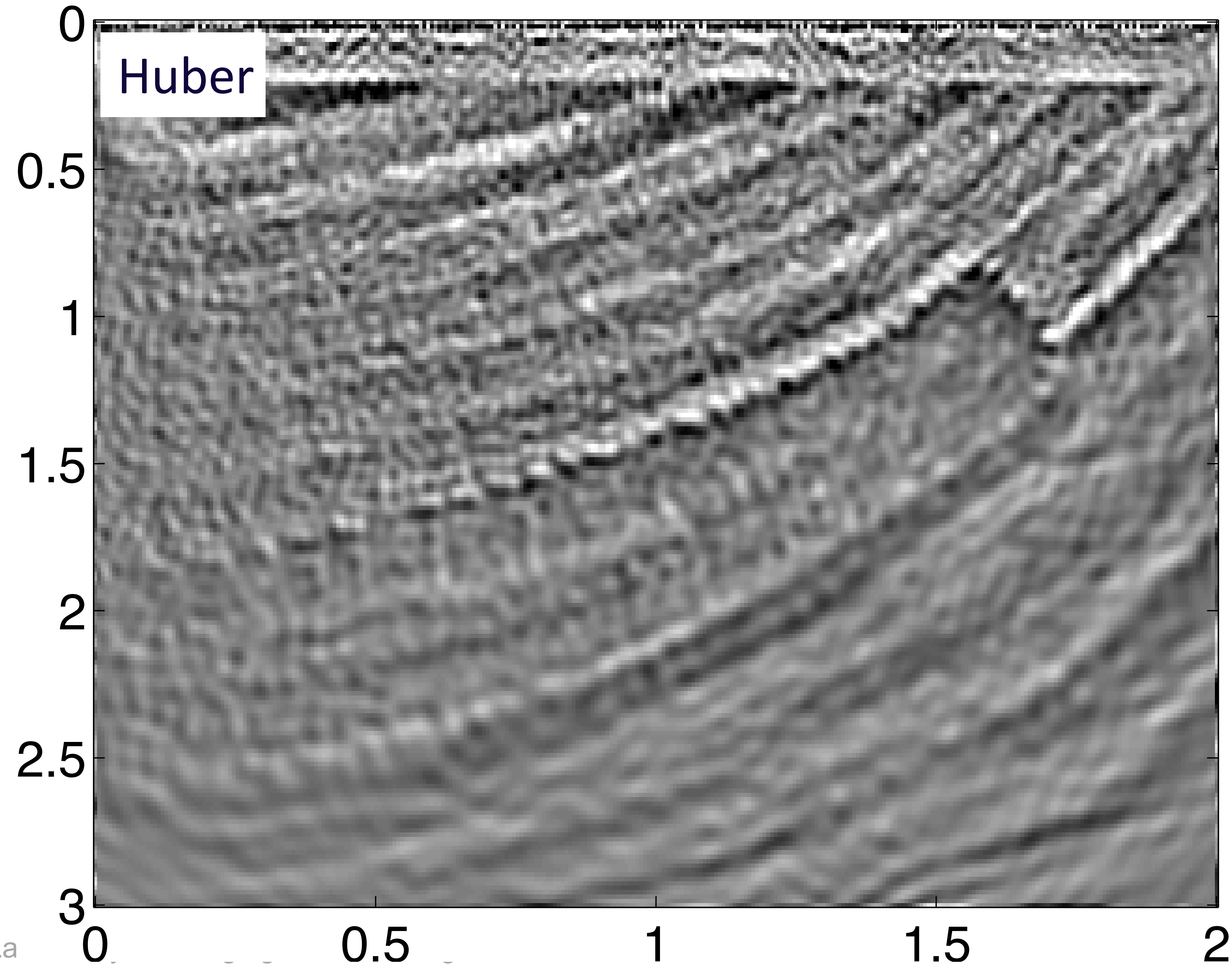


50% missing data

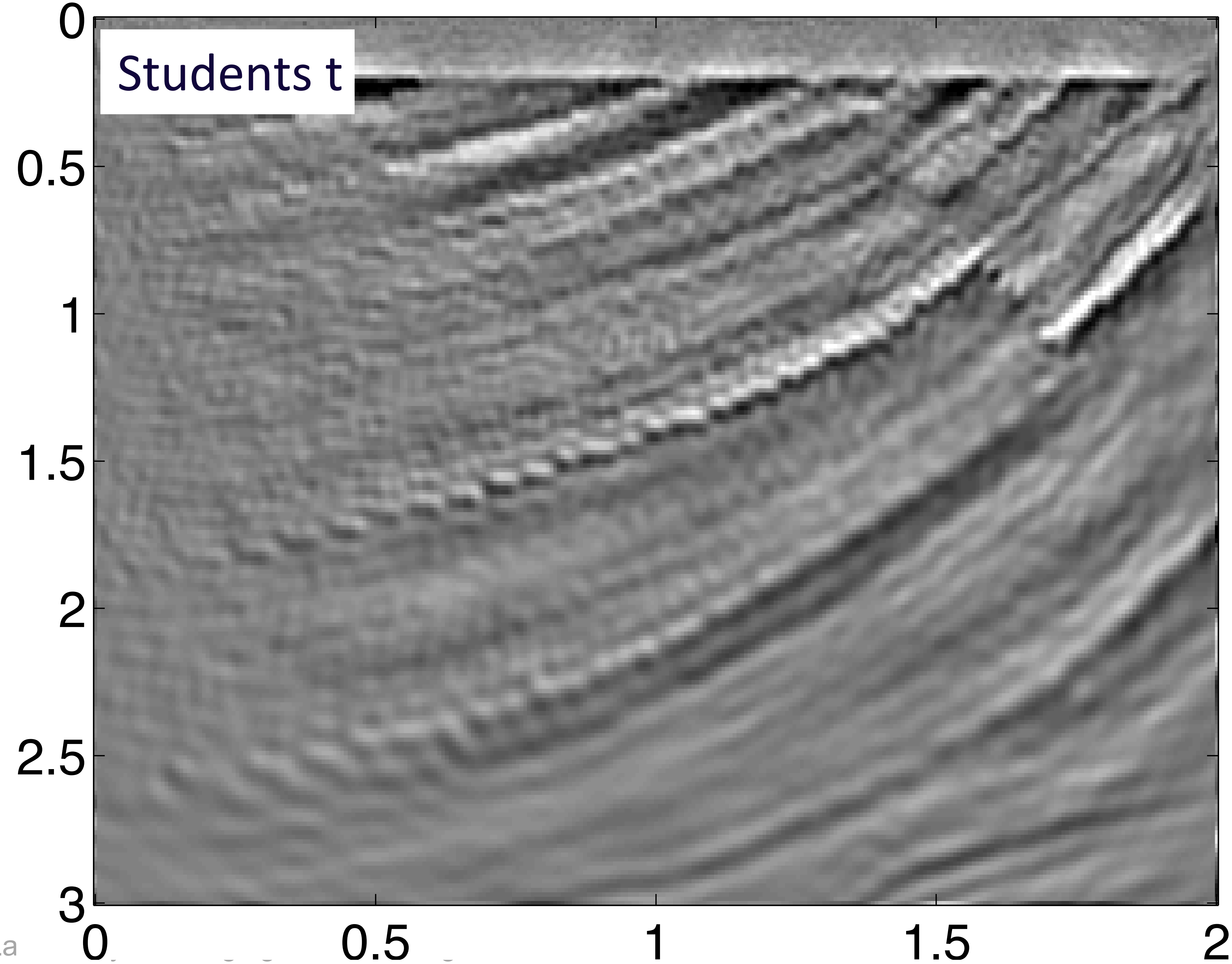
Robust FWI



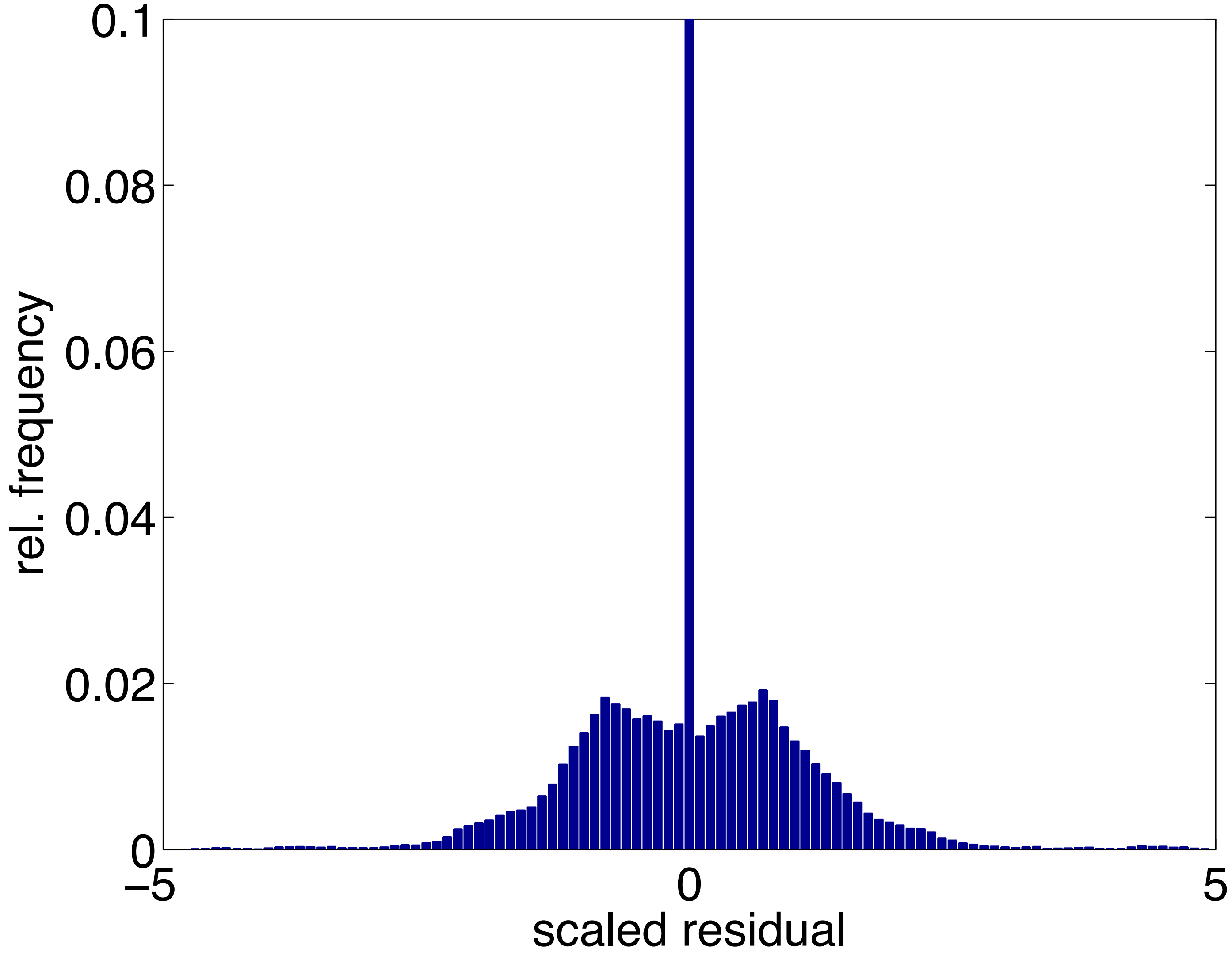
Robust FWI



Robust FWI

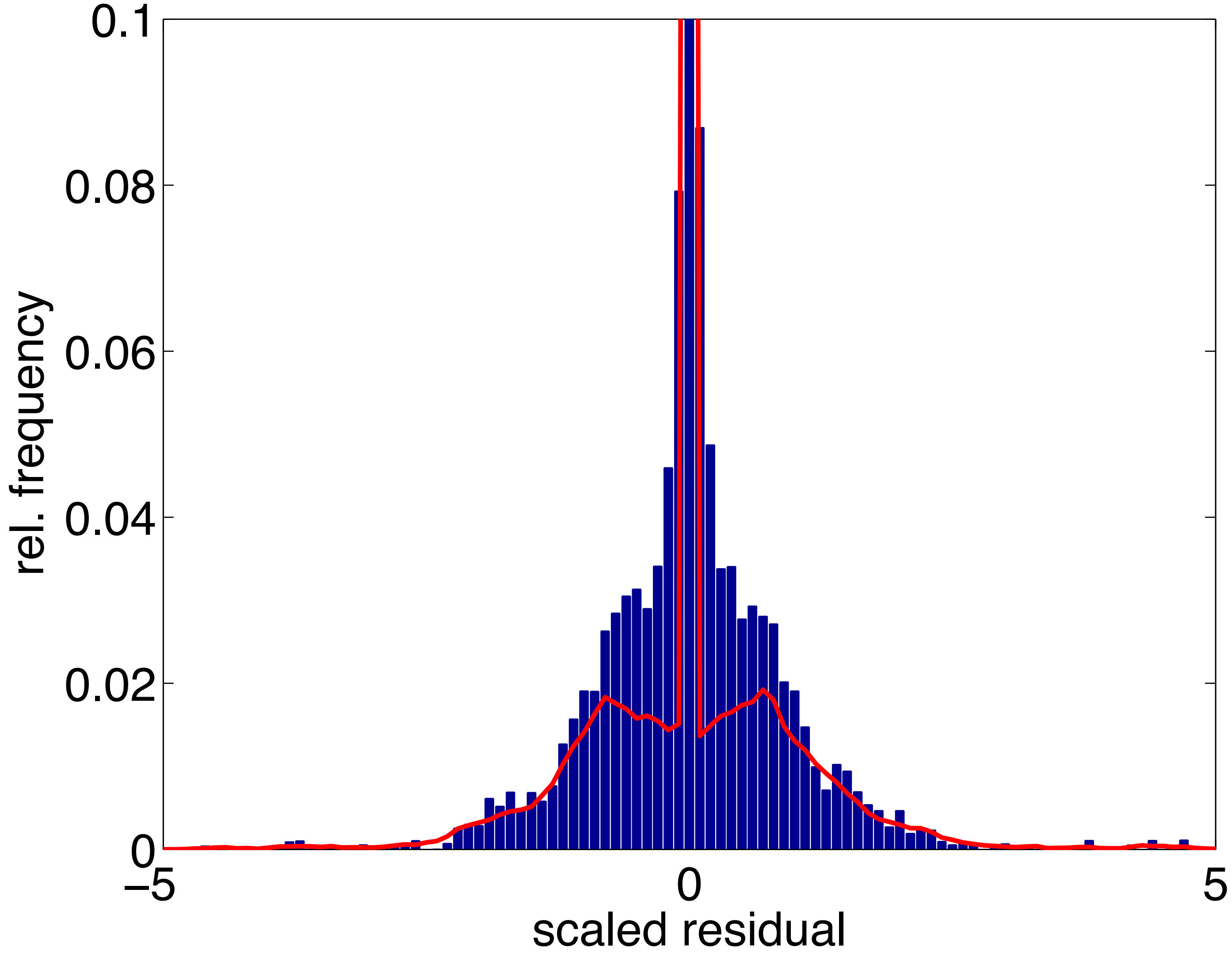


Robust FWI



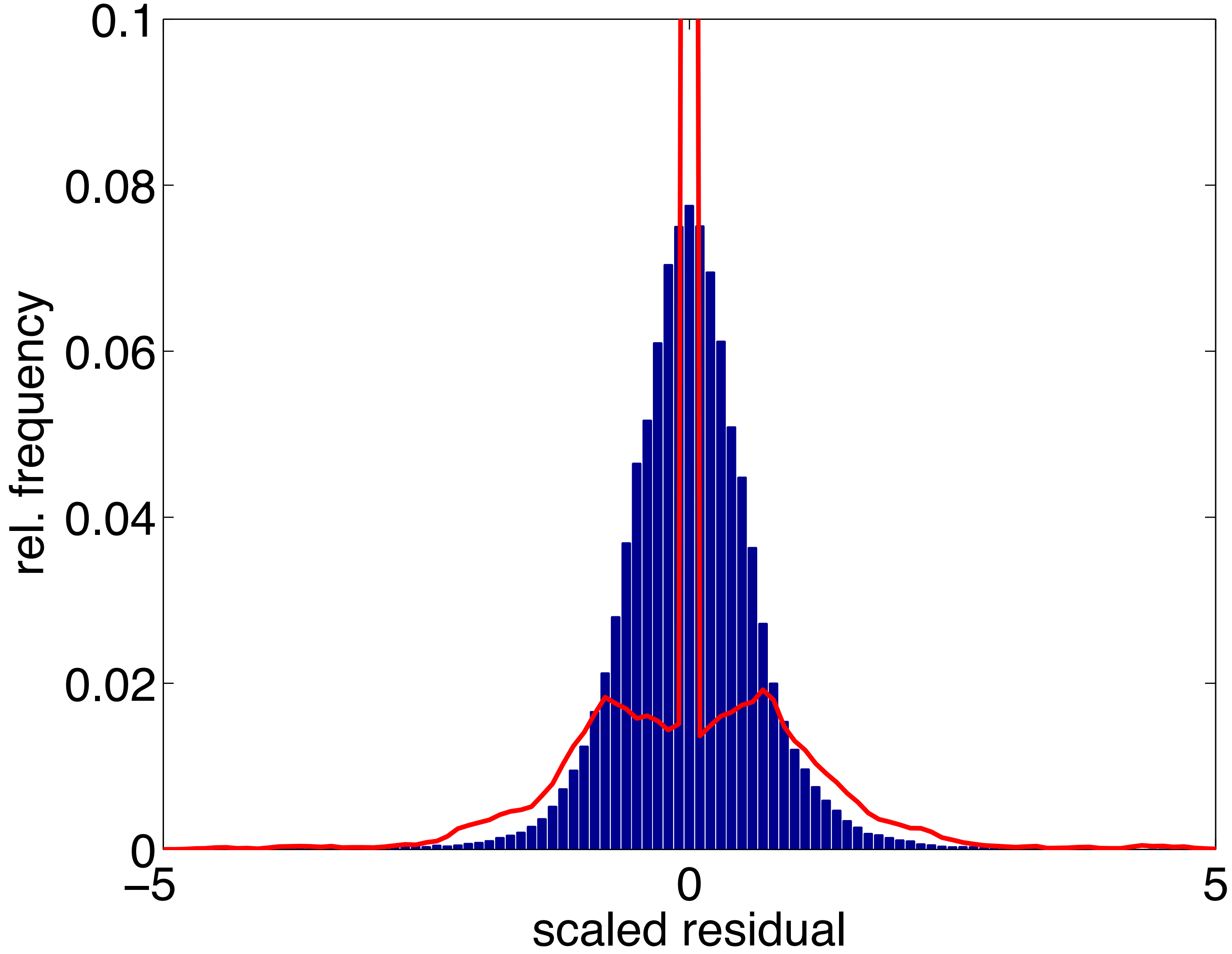
“True” Residual for Missing Data Example

Robust FWI



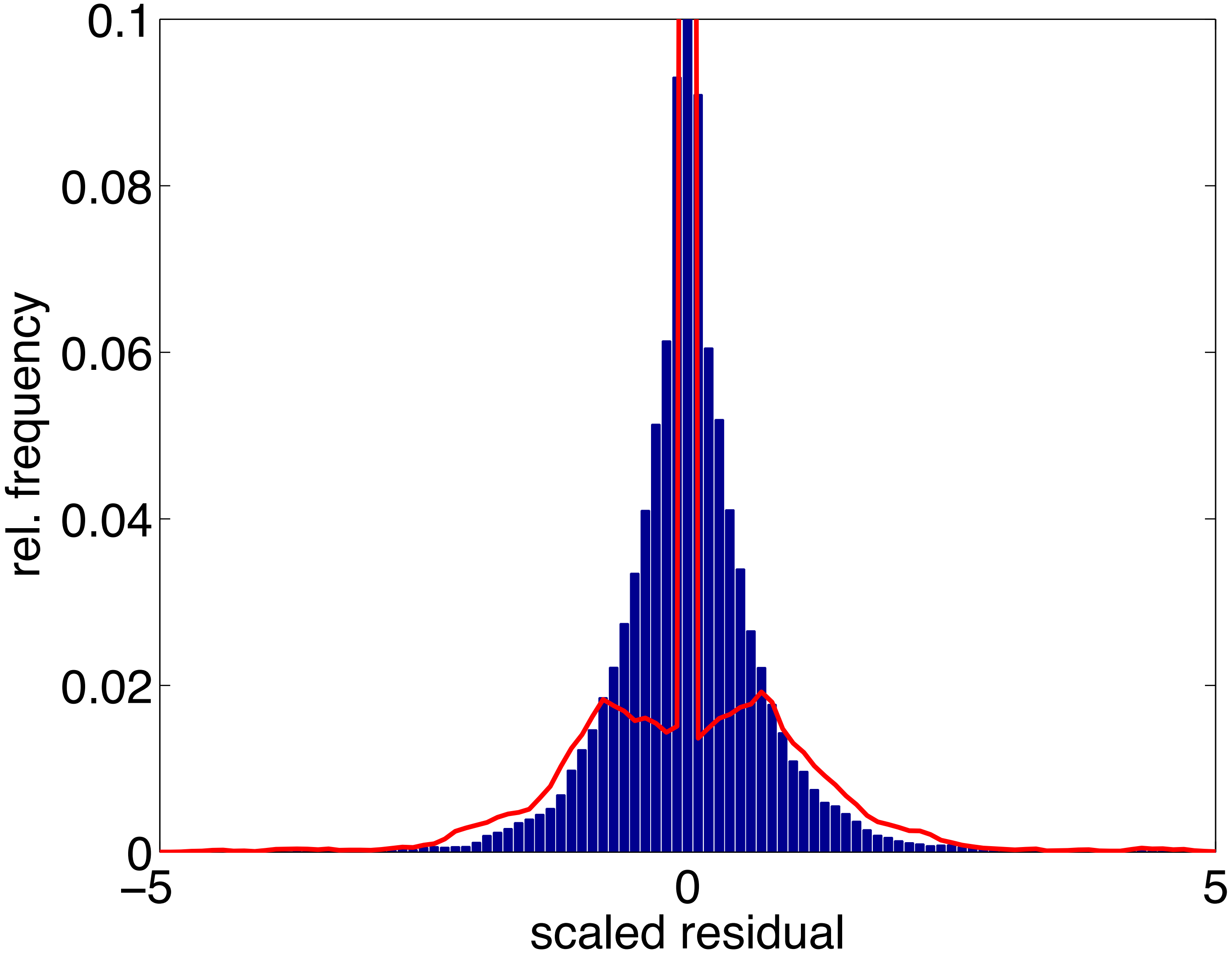
Initial Residual
for Missing Data
Example

Robust FWI



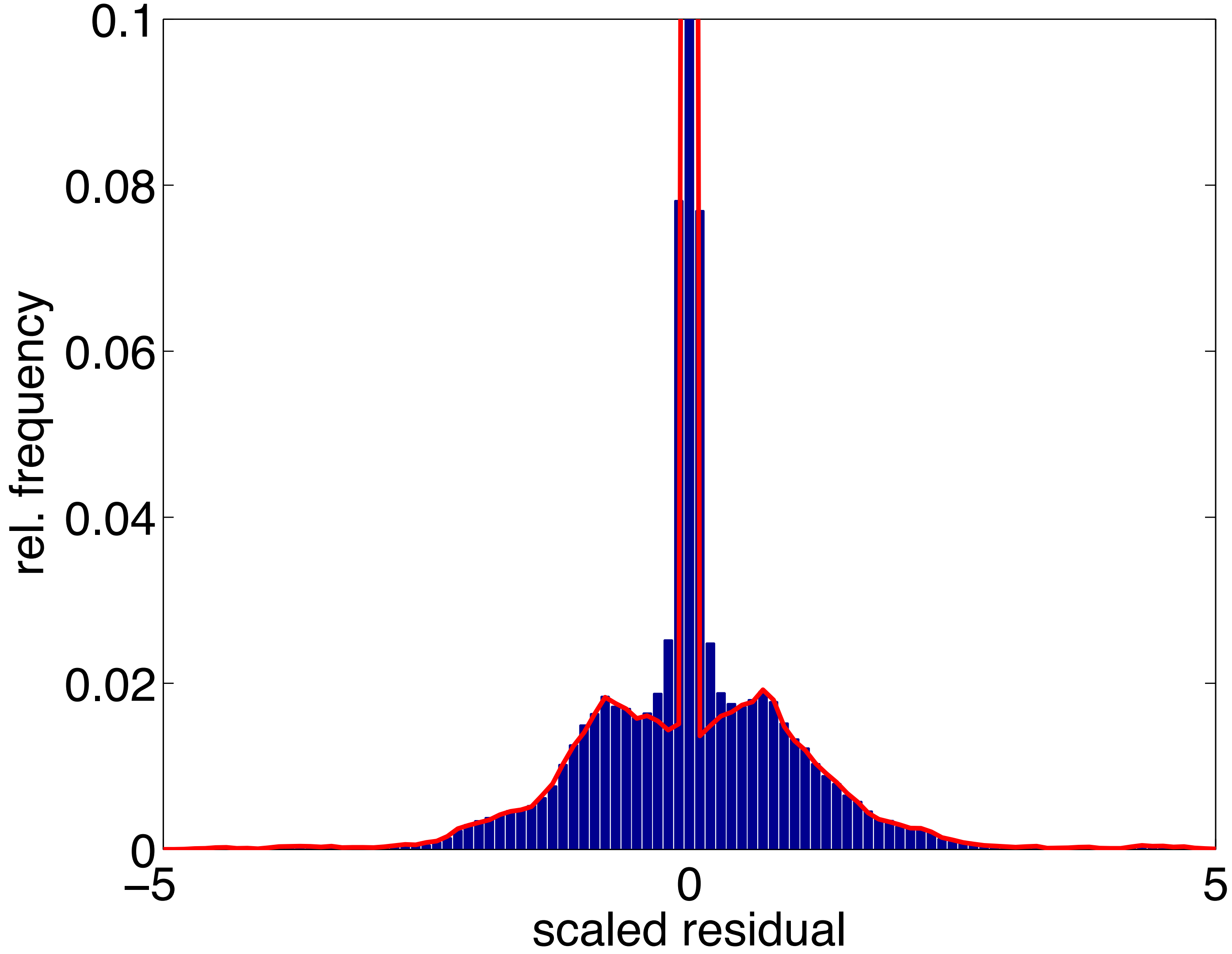
LS Fit Residual for Missing Data Example

Robust FWI



Huber Fit Residual for Missing Data Example

Robust FWI



T Fit Residual
for Missing Data
Example

Key contributions

Practical & *easy* to implement extensions of FWI

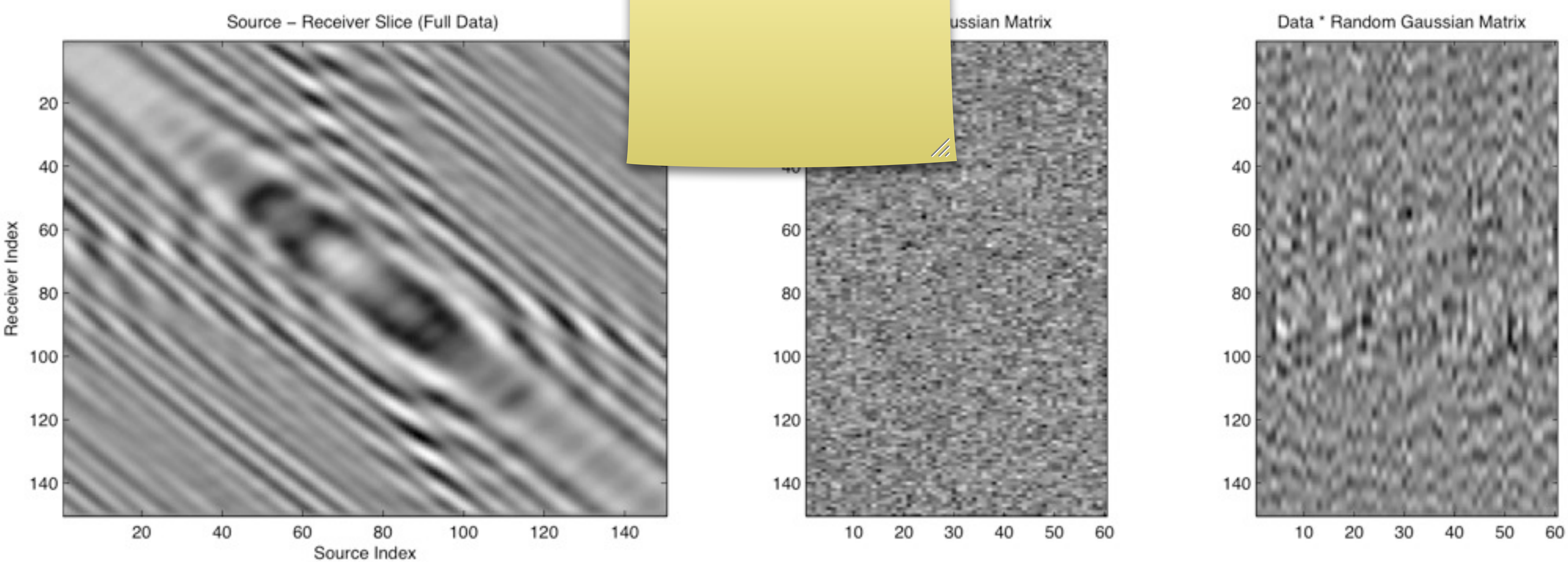
- control of the error related to the randomized batches
- control over unmodelled events in the data

Challenge: upscale to *full 3D* but collaboration with Mike will take care of that...

Experiment I

link with CS

$\mathbf{D}$ ... $\mathbf{W}$ ... $\underline{\mathbf{D}} = \mathbf{D}\mathbf{W}$



SLIM

[Demanet et. al.]

[FJH et. al., 2008-]

Multiscale structure
[model]

Multiscale & *multidirectional* structure of the Earth & wavefields

- *compressibility* w.r.t. *curvelet* frames

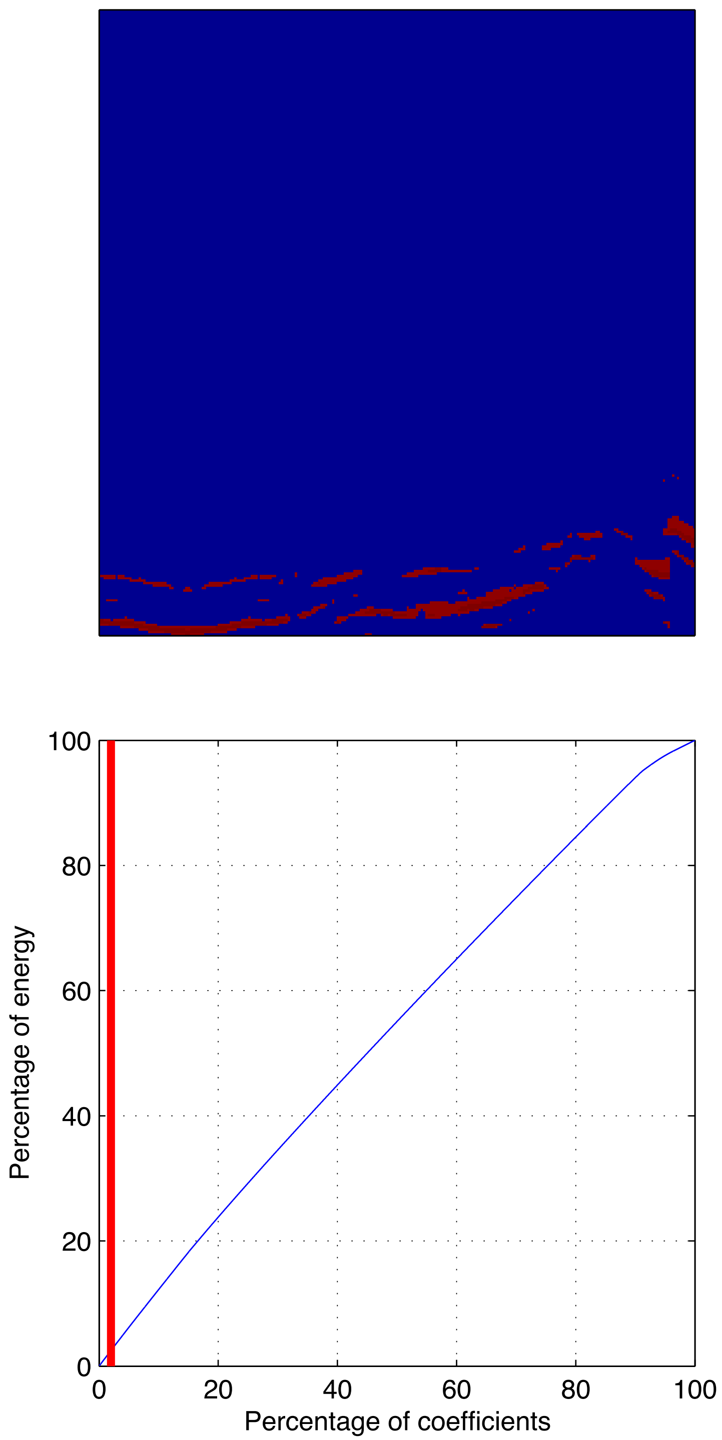
Invariance of *curvelets* under *action* wave-equation *Hessian*

FWI is *amenable* to *sparsity* promotion:

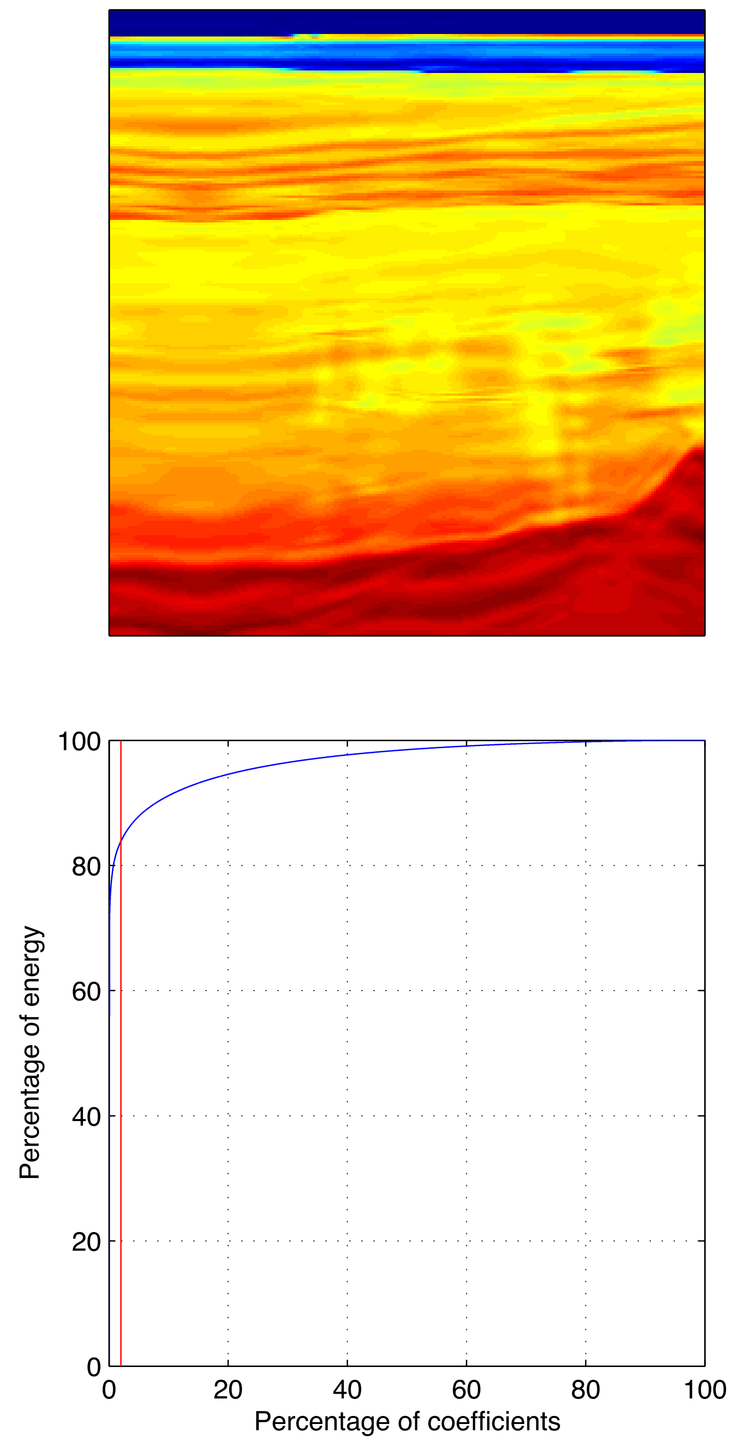
- remove source *crosstalk* & *restore* leaked *energy*
- fill in the *nullspace* of the Hessian
- regularize Gauss-Newton *updates*

Transform-domain sparsity

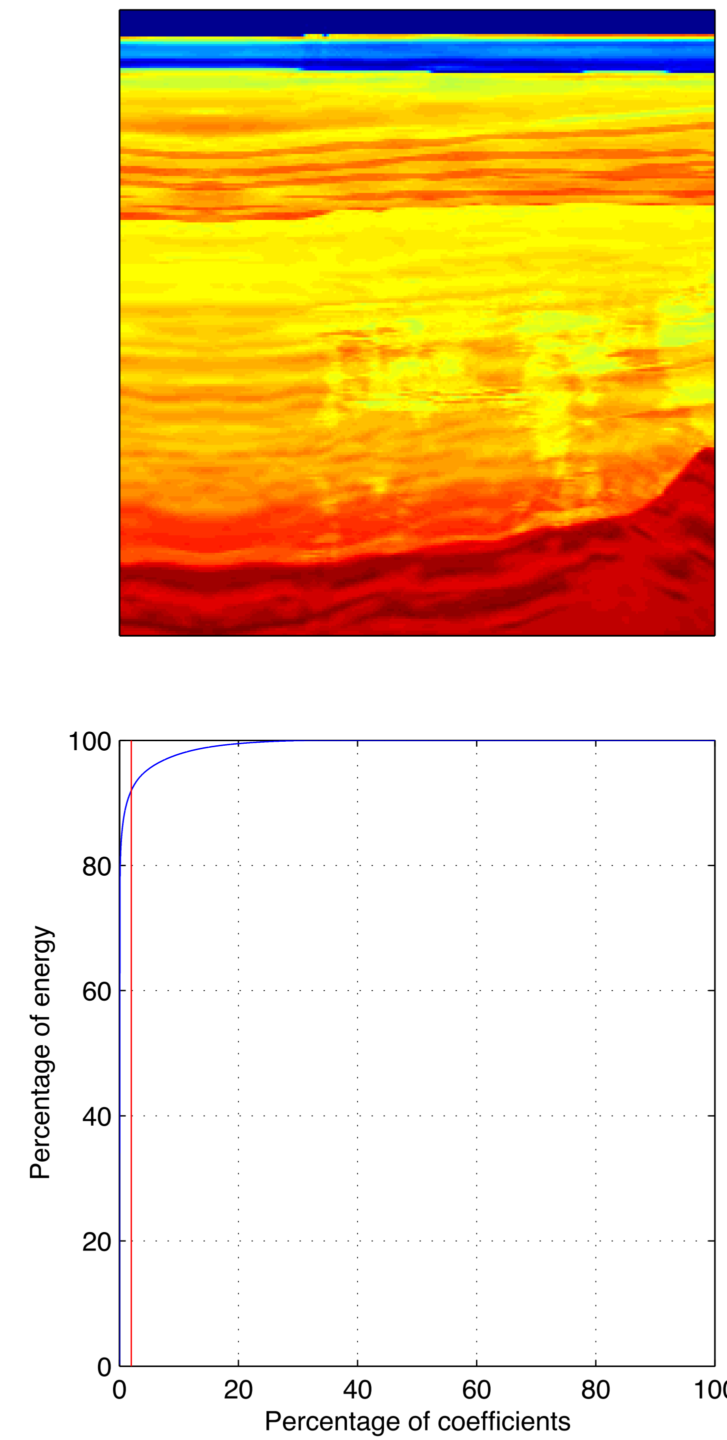
Dirac



curvelet analysis

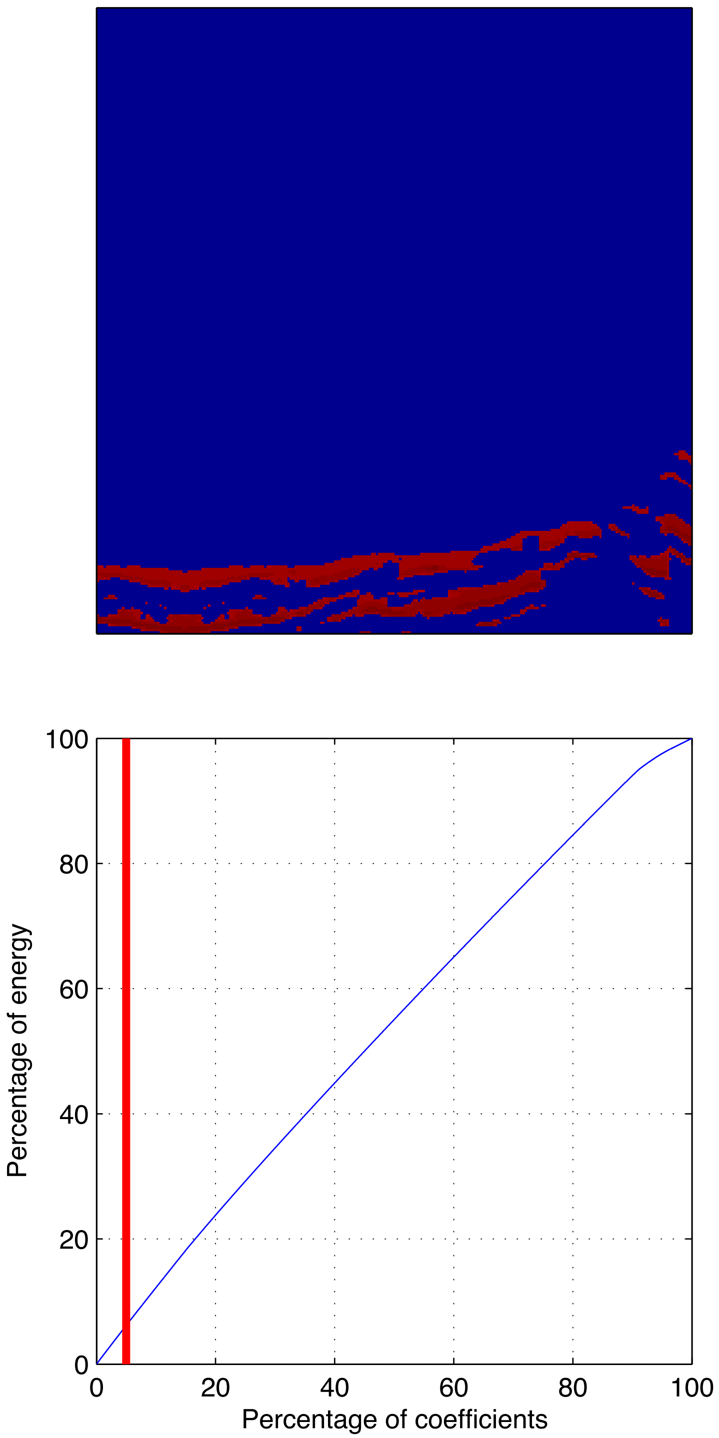


curvelet synthesis

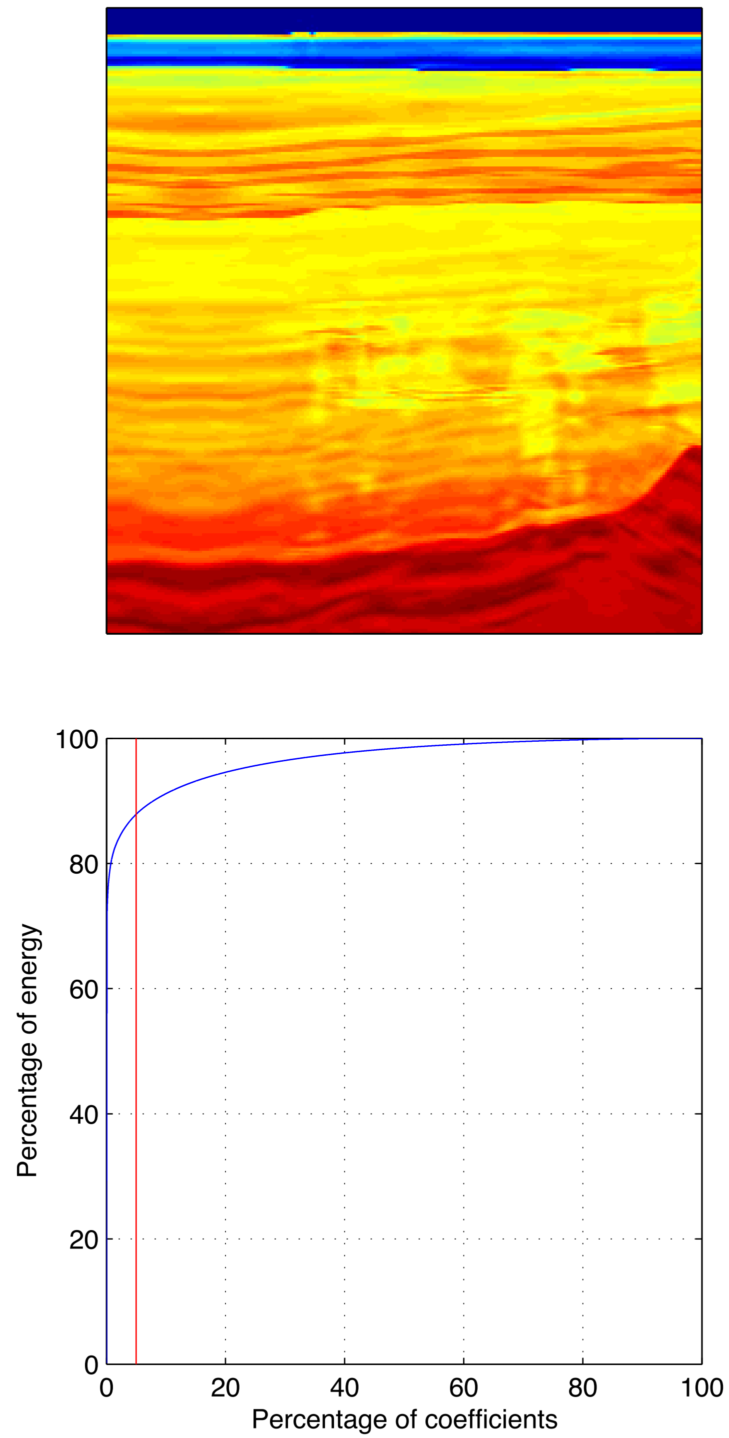


Transform-domain sparsity

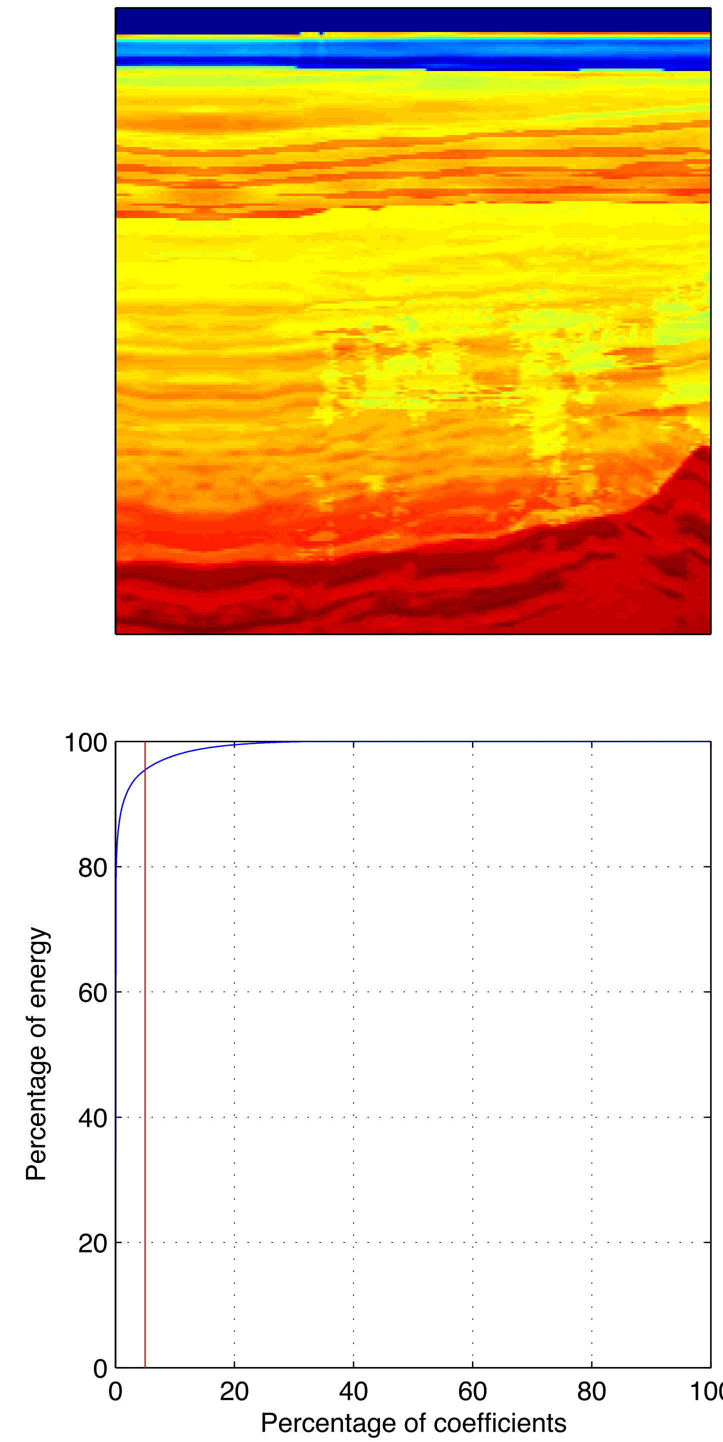
Dirac



curvelet analysis

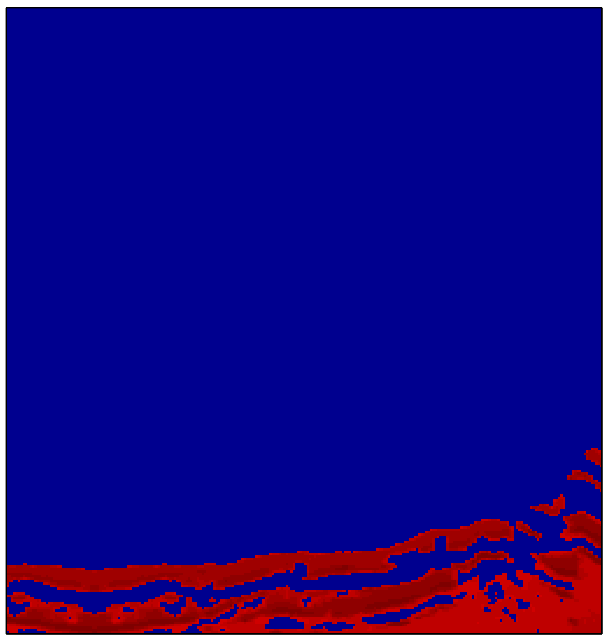


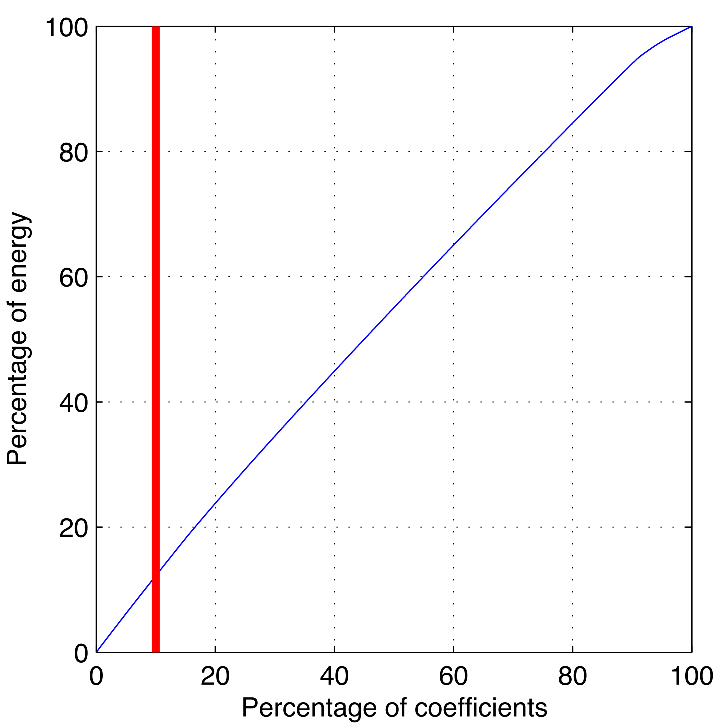
curvelet synthesis



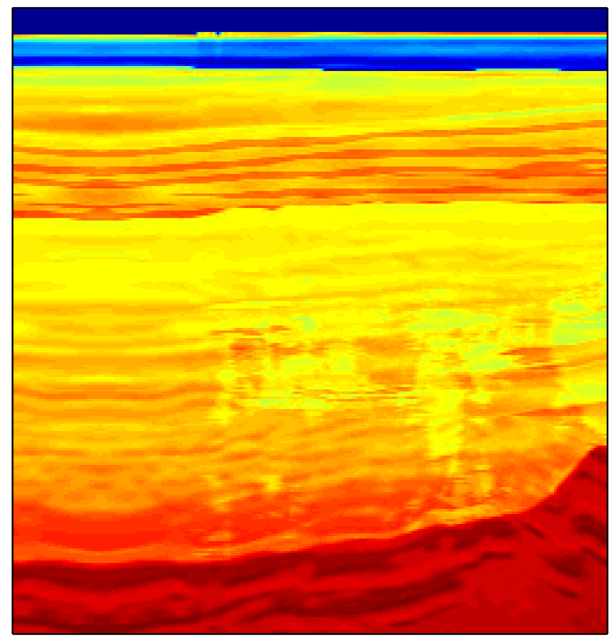
Transform-domain sparsity

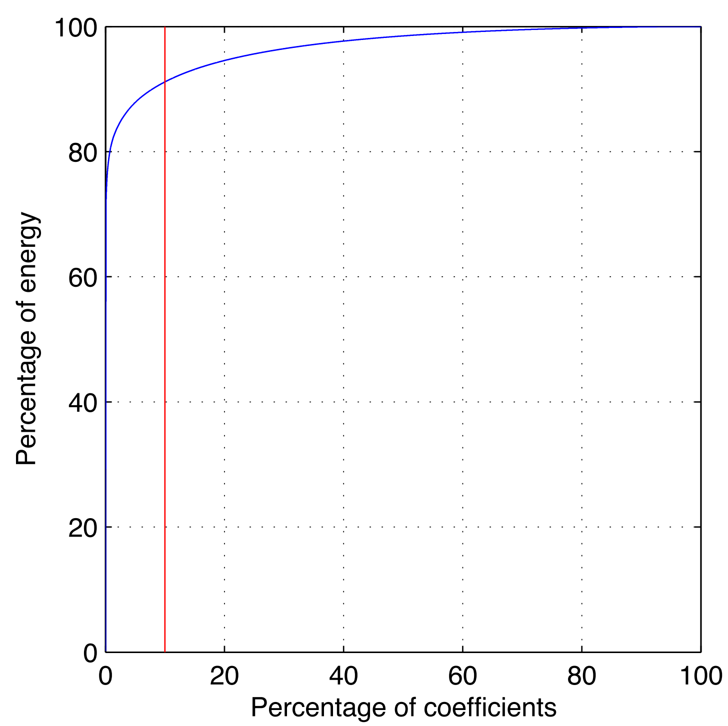
Dirac



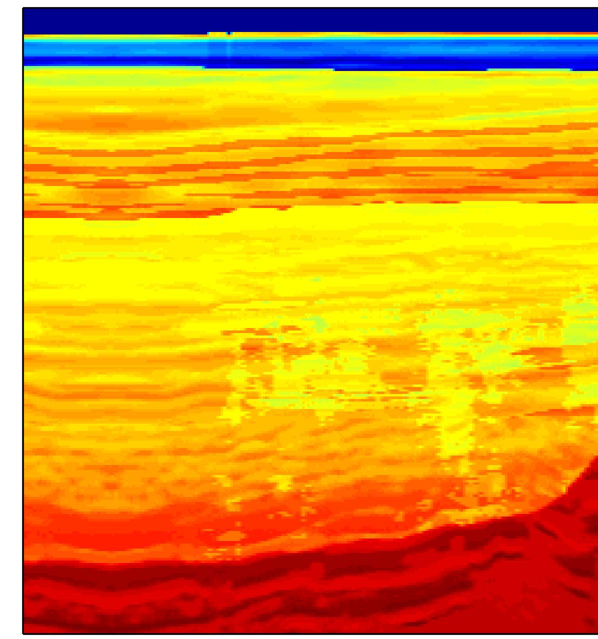


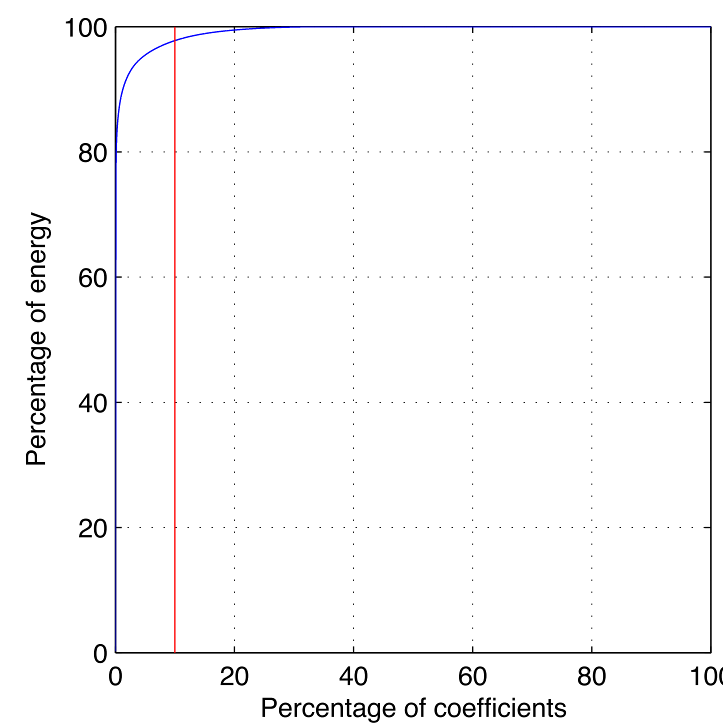
curvelet analysis





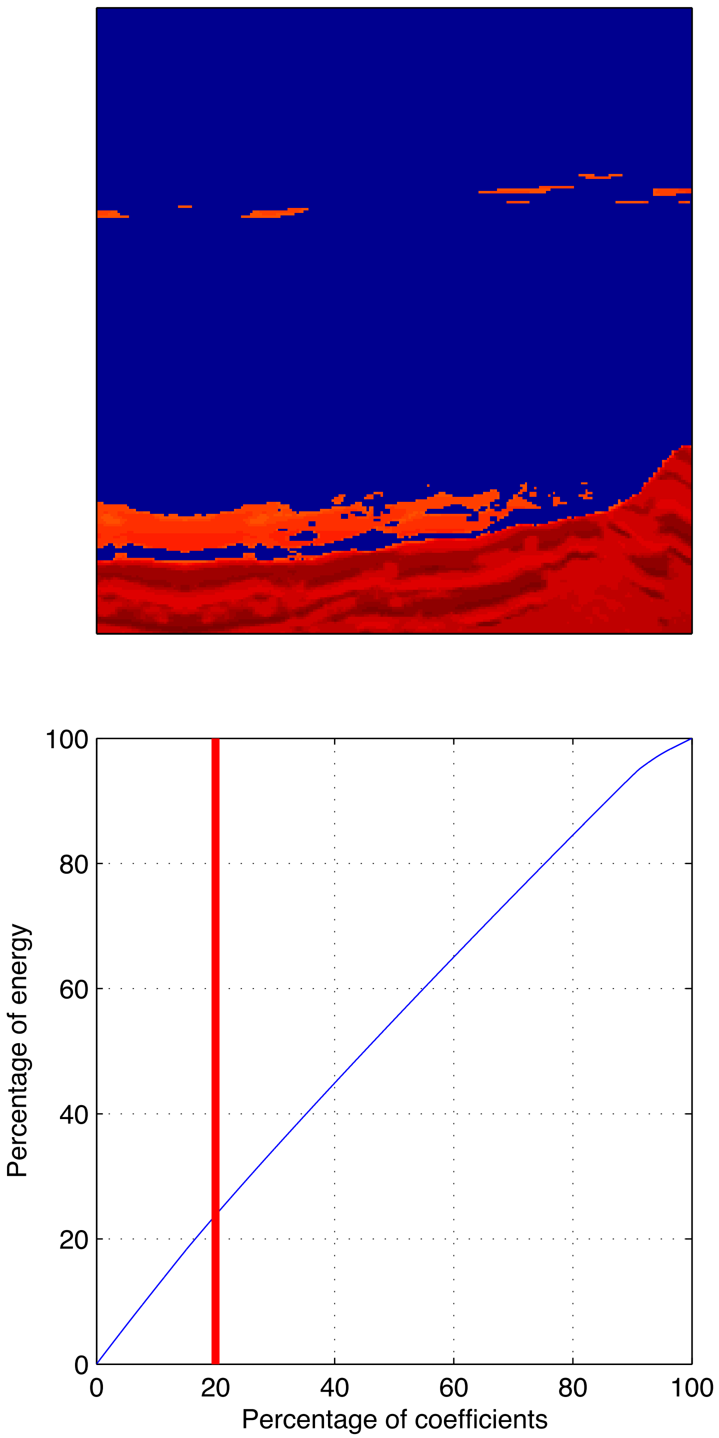
curvelet synthesis



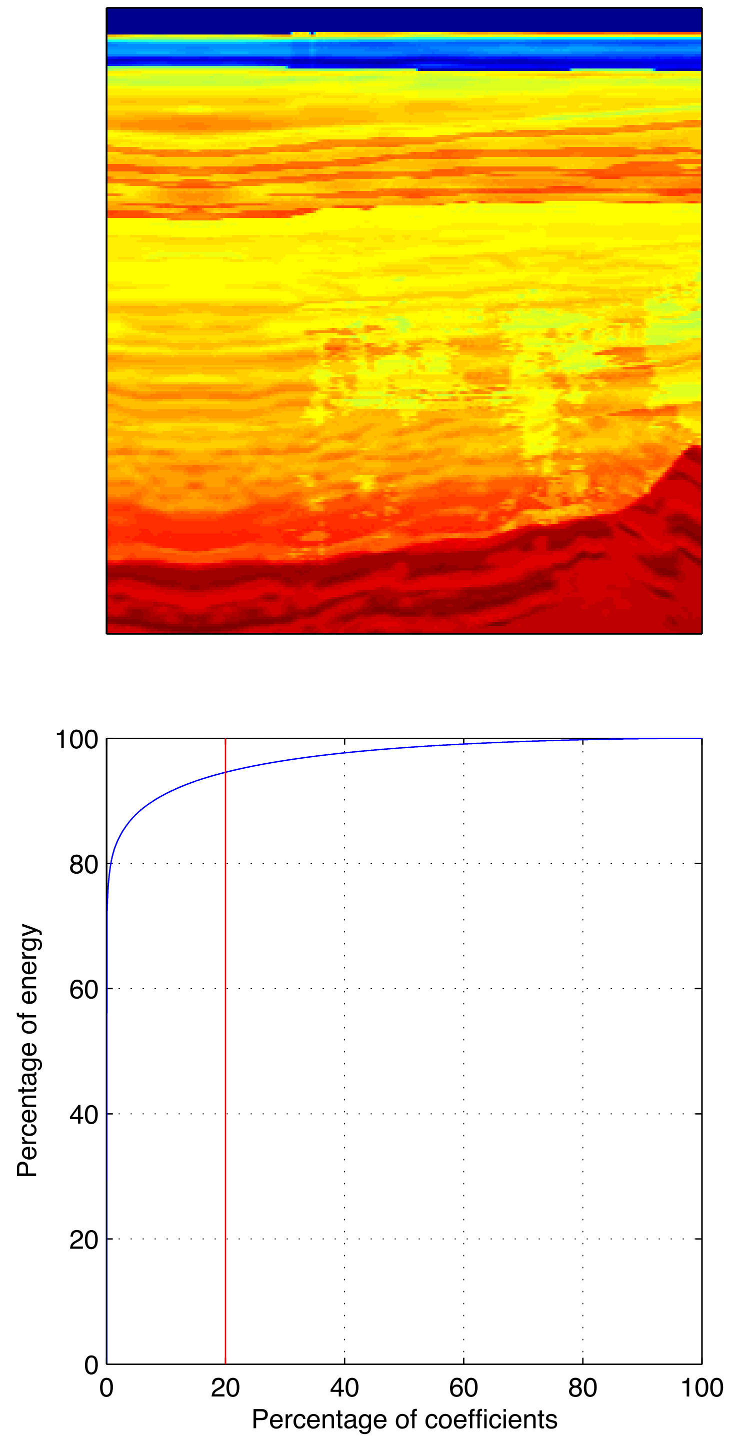


Transform-domain sparsity

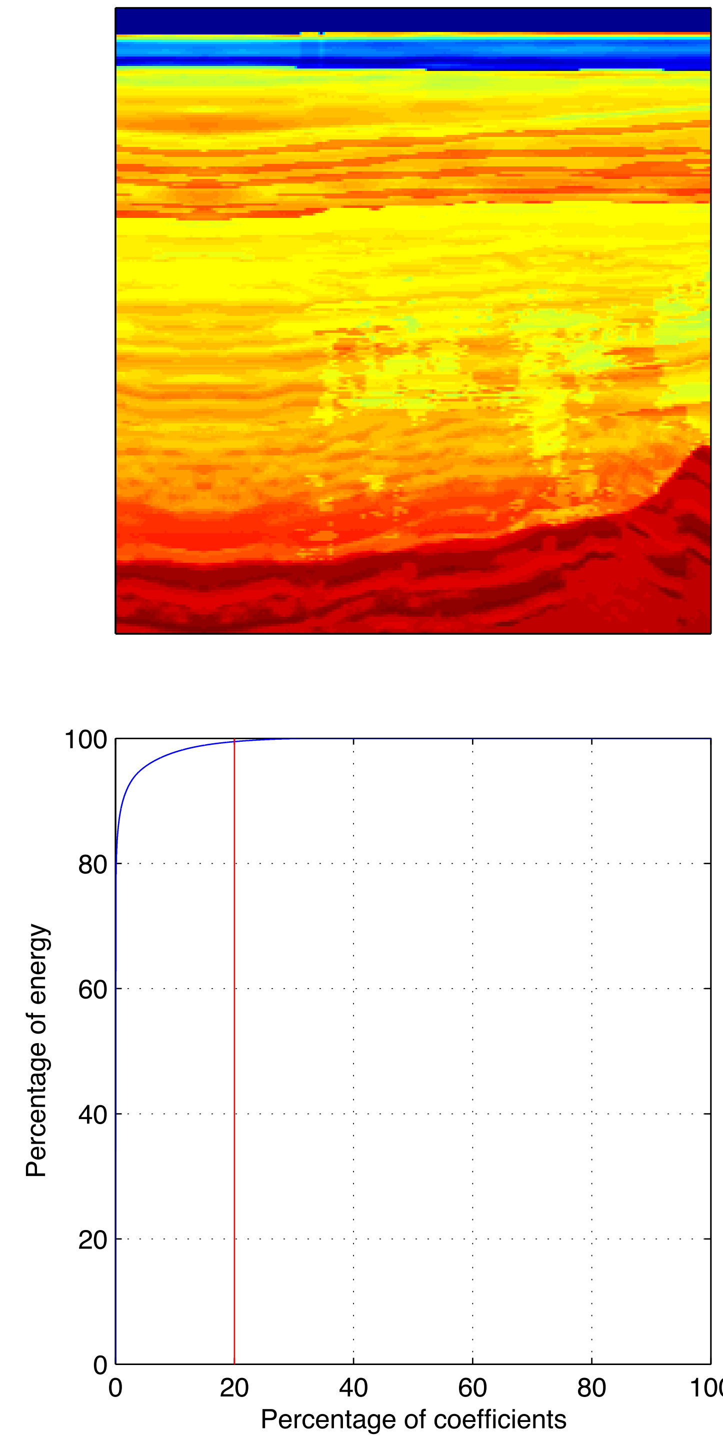
Dirac



curvelet analysis



curvelet synthesis

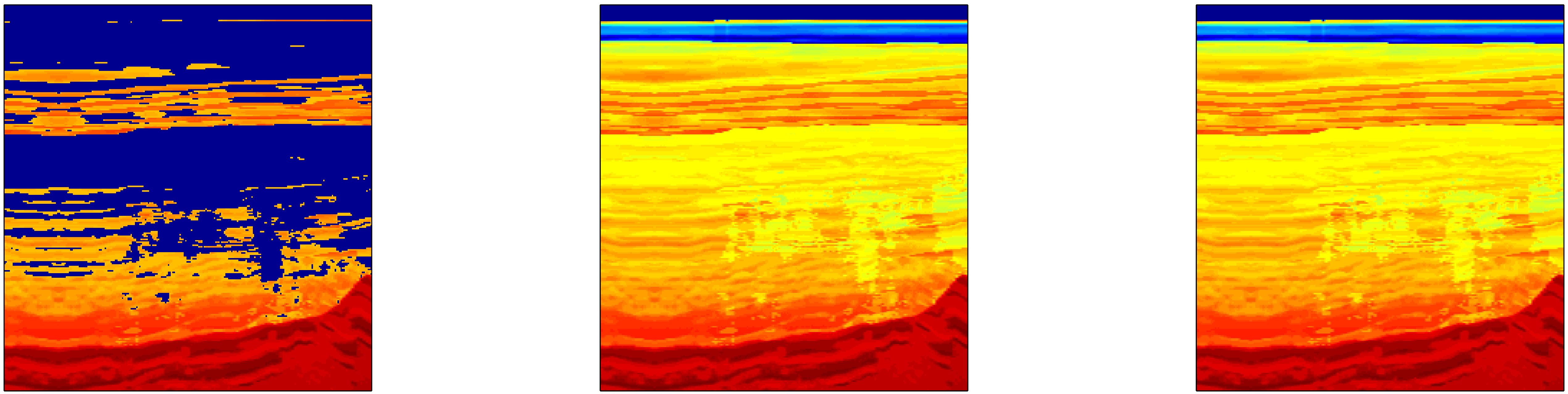


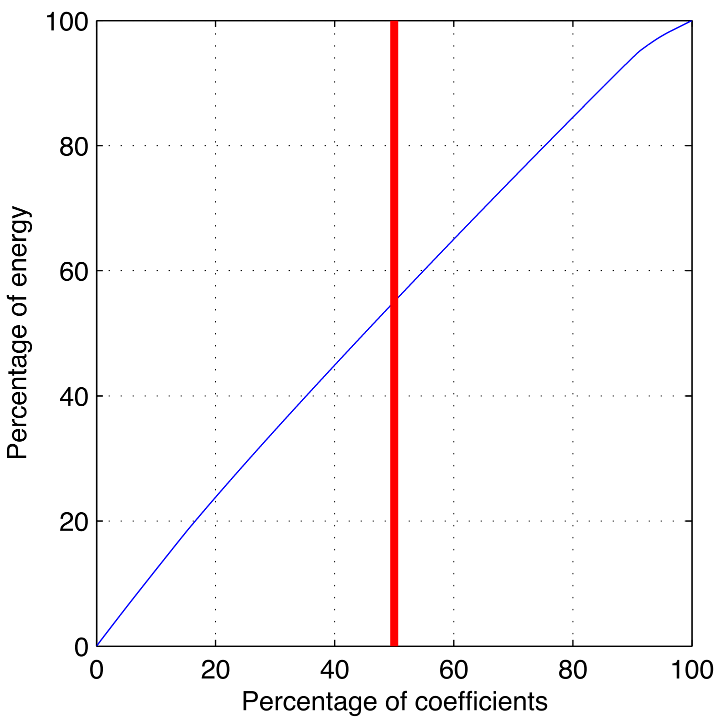
Transform-domain sparsity

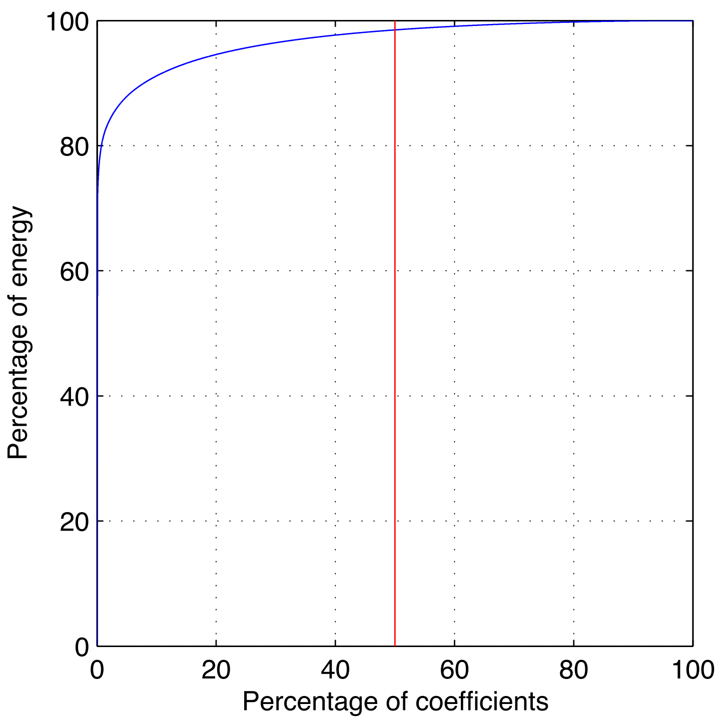
Dirac

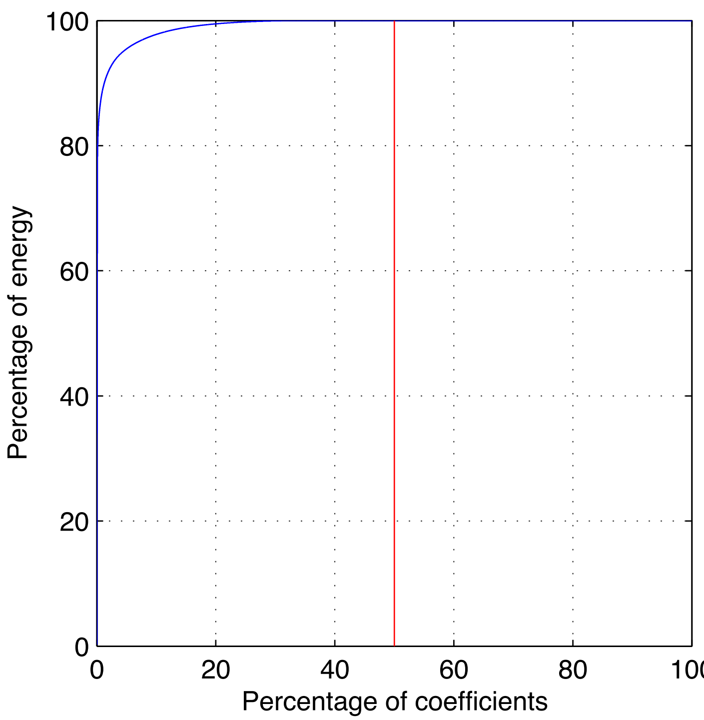
curvelet analysis

curvelet synthesis









Thursday, November 17, 2011

Convex composite structure [Burke & Ferris, '95.]

FWI:

$$\min_{\mathbf{m}} \quad \phi(\mathbf{m}) := \underbrace{\frac{1}{2}\| \overbrace{\mathbf{D} - \boldsymbol{\mathcal{F}}[\mathbf{m};\mathbf{Q}]}^{\text{smooth}} \|_F^2}_{\text{convex}}$$

- exploit *convexity* by linearizing *within*

$$\min_{\mathbf{m}} \quad \phi(\mathbf{m}) := \frac{1}{2}\|\mathbf{D} - \boldsymbol{\mathcal{F}}[\mathbf{m};\mathbf{Q}] - \nabla\boldsymbol{\mathcal{F}}[\mathbf{m};\mathbf{Q}]\boldsymbol{\delta}\mathbf{m}\|_F^2$$

- control the norm of the updates to *guarantee* convergence

Convex composite structure [Burke & Ferris, '95.]

FWI:

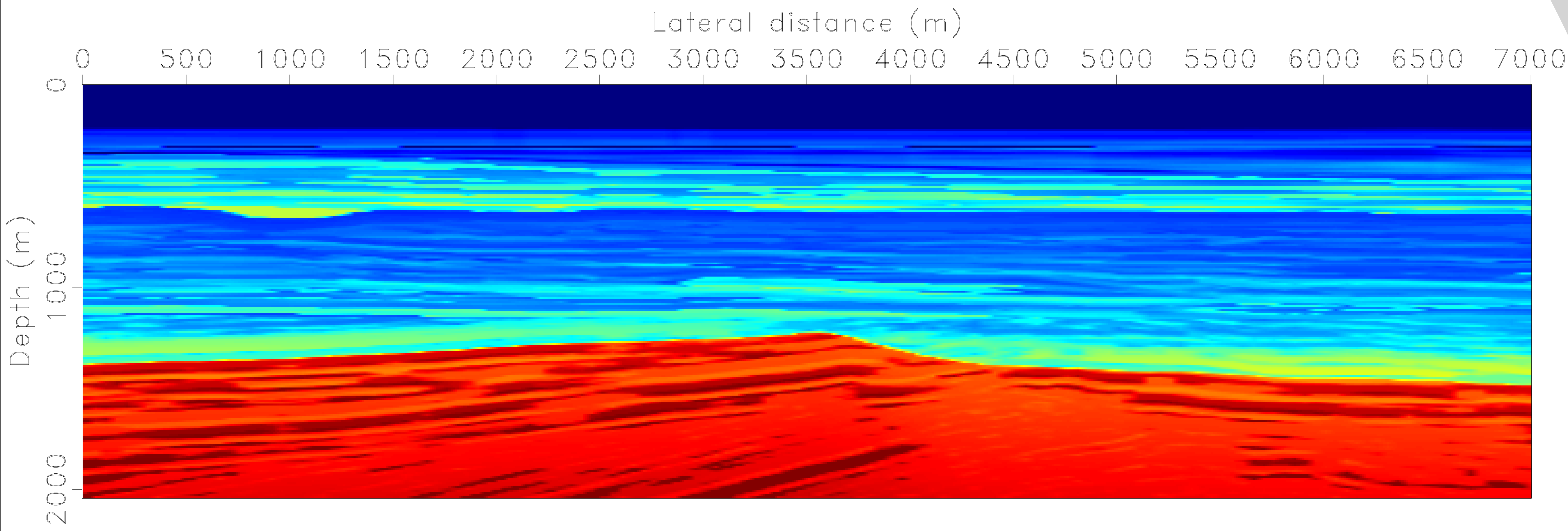
$$\min_{\mathbf{m}} \quad \phi(\mathbf{m}) := \underbrace{\frac{1}{2}\|\overbrace{\mathbf{D} - \boldsymbol{\mathcal{F}}[\mathbf{m};\mathbf{Q}]}^{\text{smooth}}\|_F^2}_{\text{convex}}$$

- exploit *convexity* by linearizing *within*

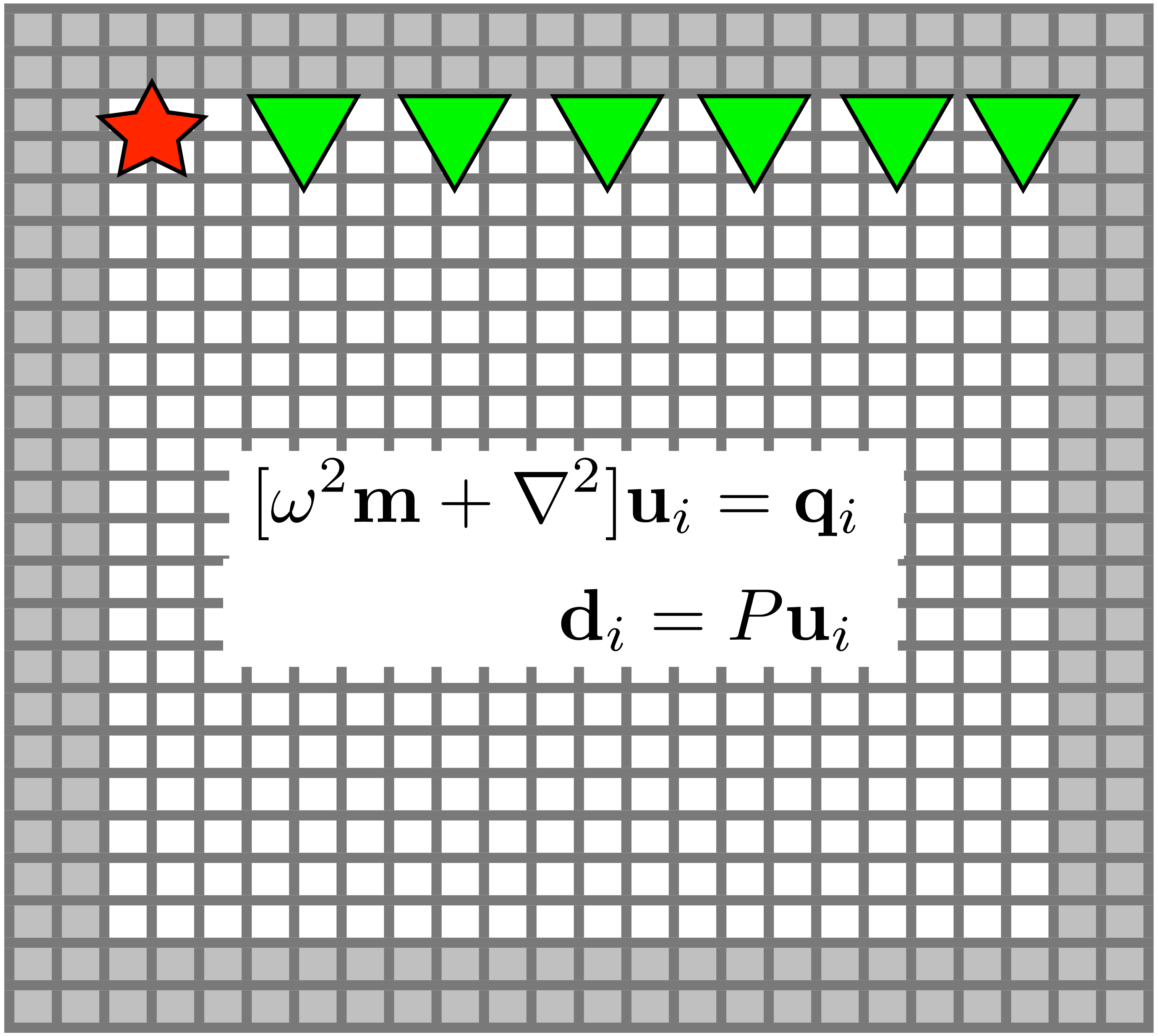
$$\min_{\mathbf{m}} \quad \phi(\mathbf{m}) := \frac{1}{2}\|\underline{\mathbf{D}} - \boldsymbol{\mathcal{F}}[\mathbf{m};\underline{\mathbf{Q}}] - \boldsymbol{\nabla}\boldsymbol{\mathcal{F}}[\mathbf{m};\underline{\mathbf{Q}}]\boldsymbol{\delta}\mathbf{m}\|_F^2$$

- control the norm of the updates to *guarantee* convergence

Example
BG Compass model



FWI results



$$[\omega^2\mathbf{m} + \nabla^2]\mathbf{u}_i = \mathbf{q}_i$$

$$\mathbf{d}_i = P\mathbf{u}_i$$

FWI results

Time-harmonic Helmholtz:

- 205 X 701 with mesh size of 10m
- 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

FWI 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 12Hz
- Recording time for each shot is 3.6s

SLIM

FWI results

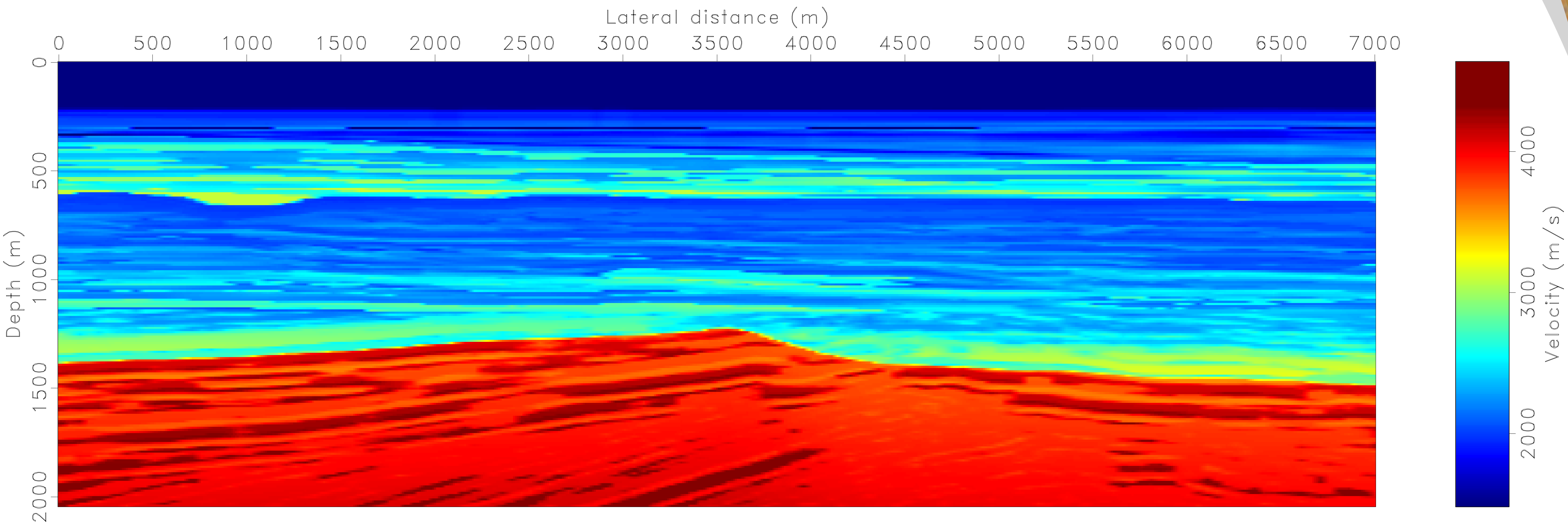
FWI:

- 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]

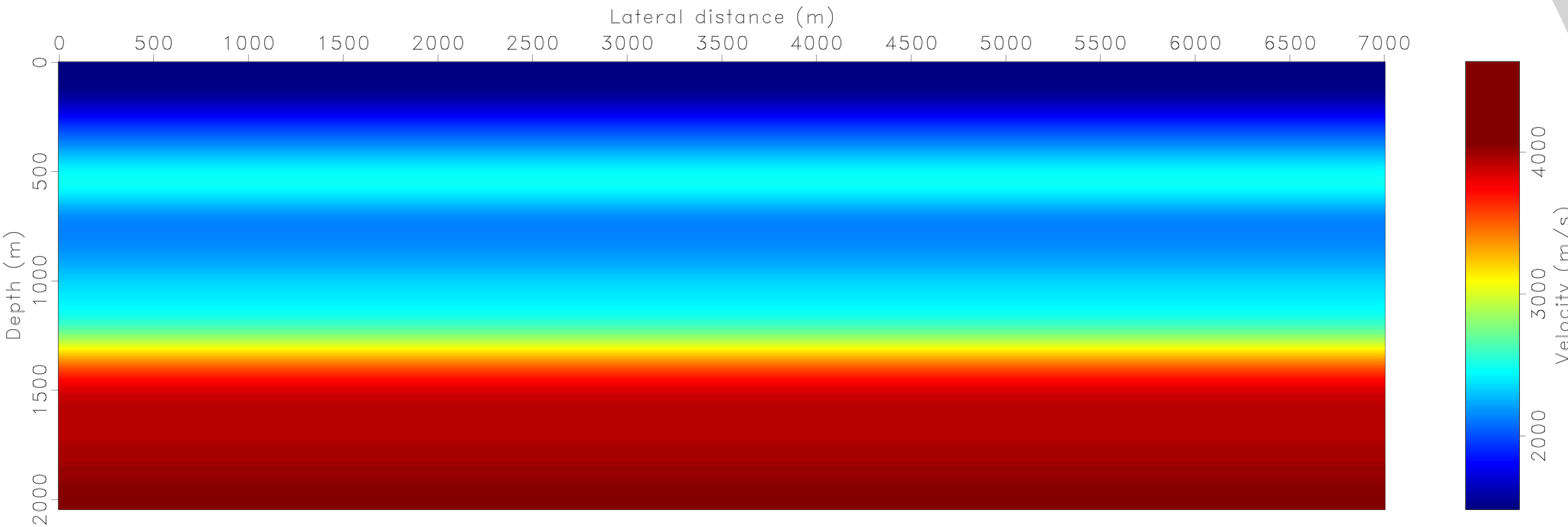
Results

True model



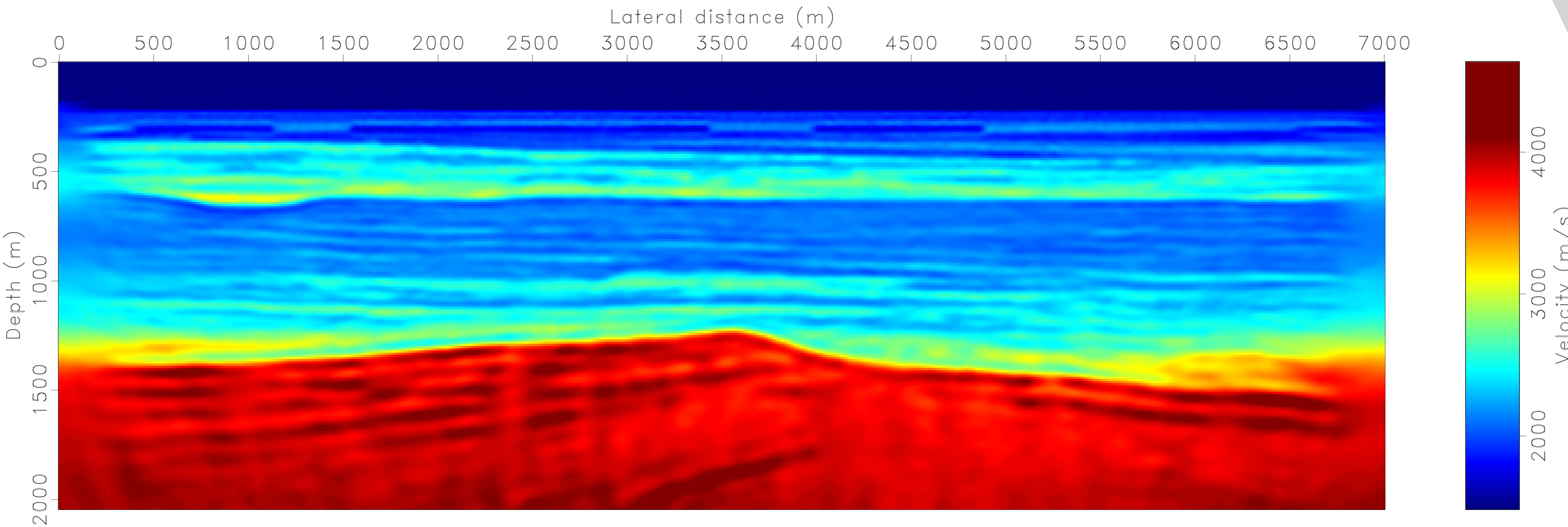
Results

Initial model



Results

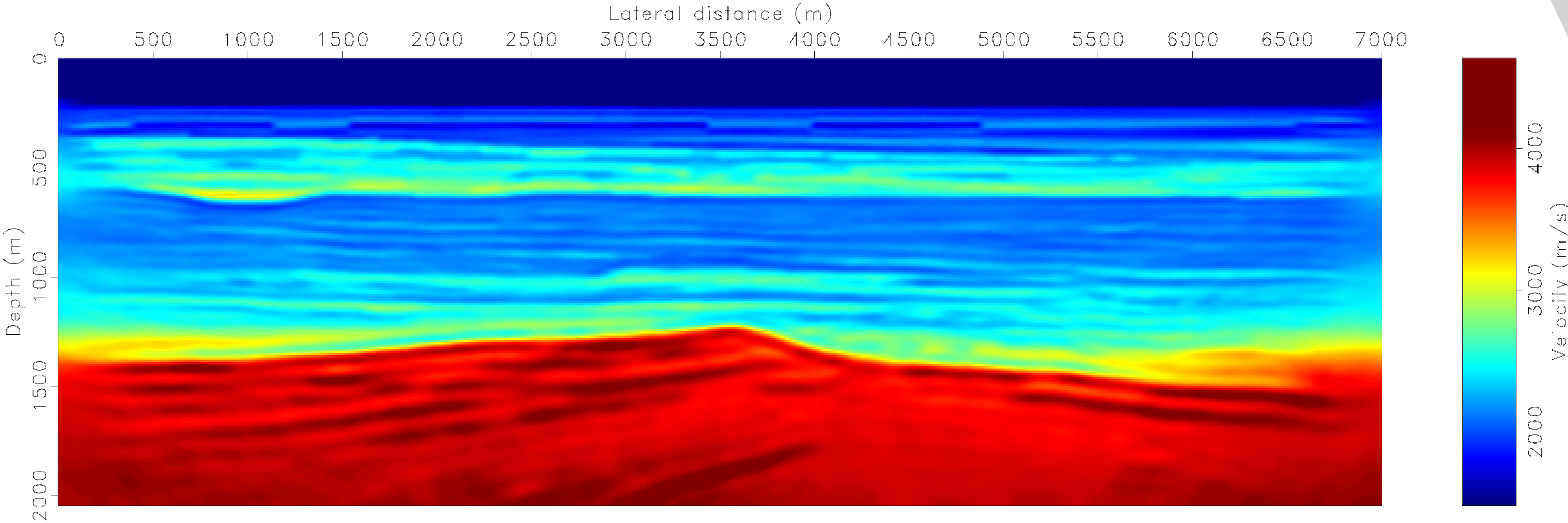
Modified GN 7 sim. shots



25 times speedup compared to full GN

Results

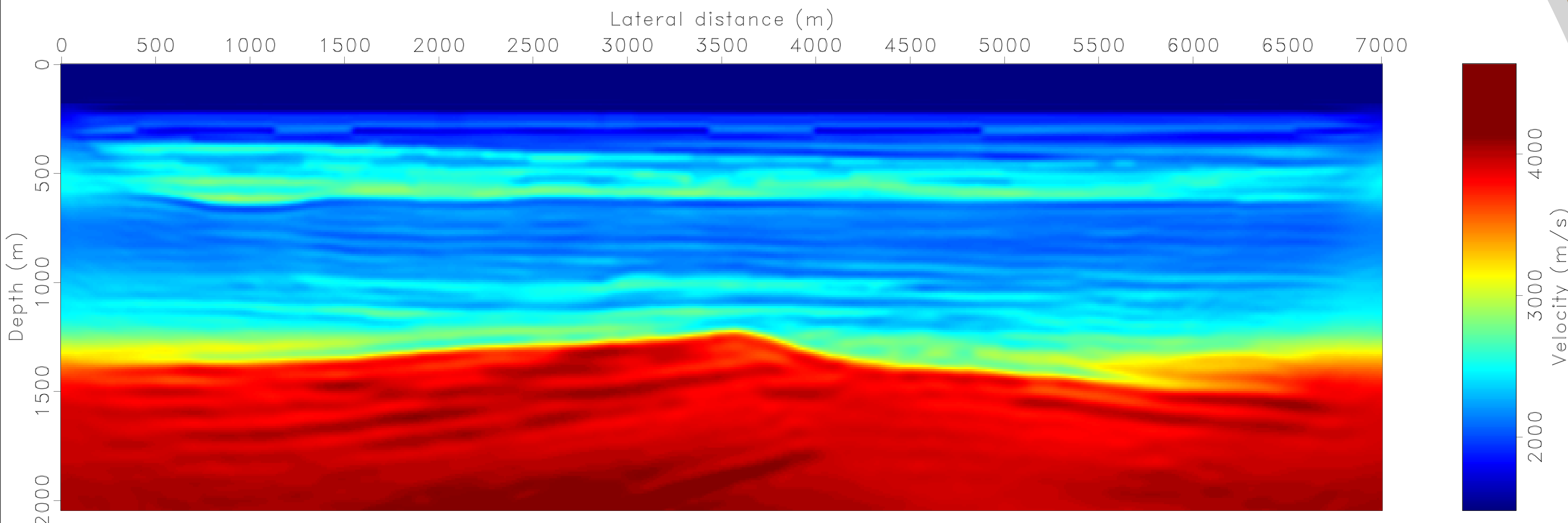
Modified GN 7 sim. shots *with renewals*



25 times speedup compared to full GN

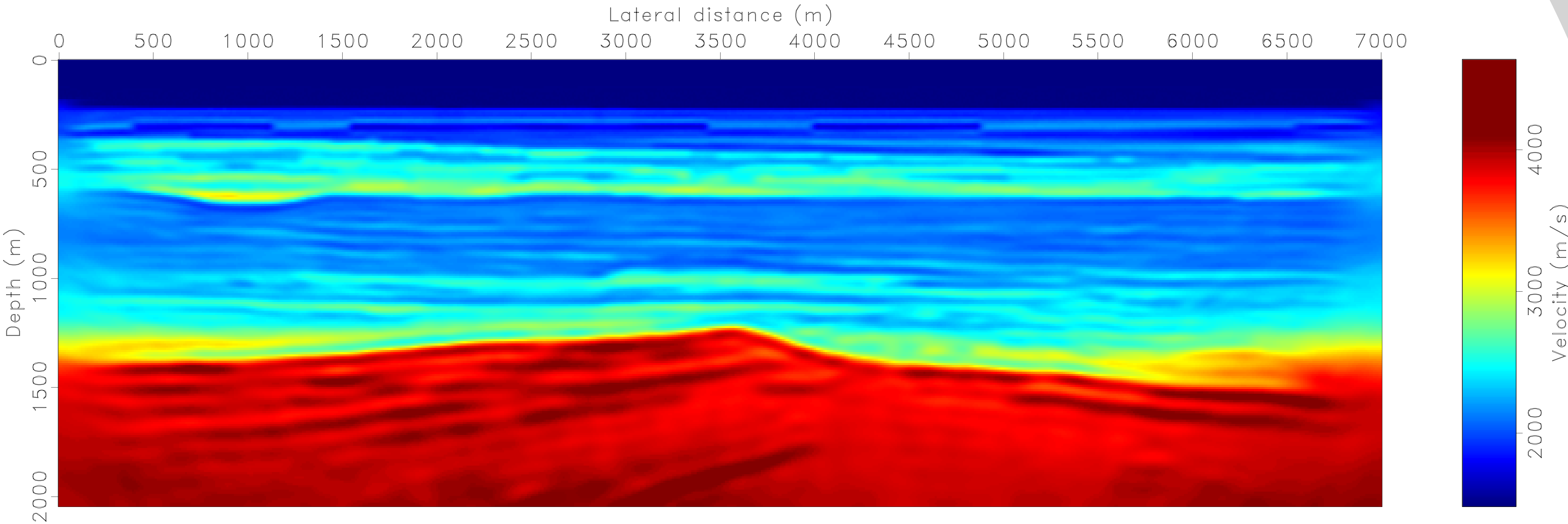
Results

quasi-Newton (l-BFGS)



Results

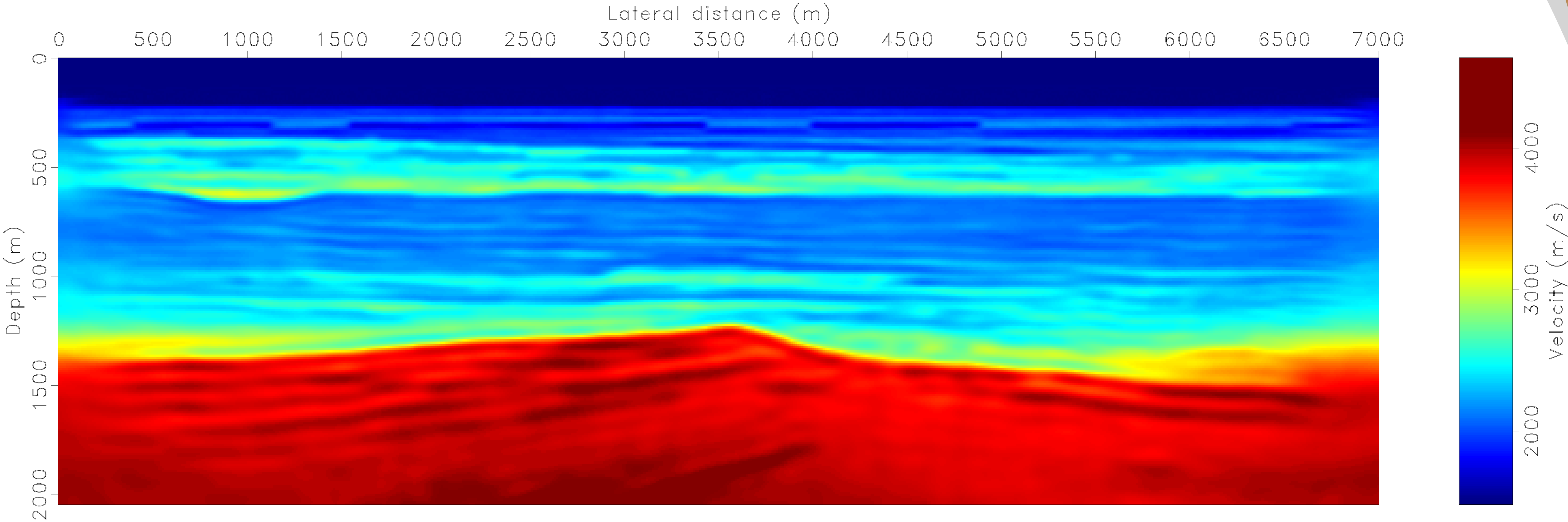
Modified GN 7 sim. shots *with renewals*



25 times speedup compared to full GN

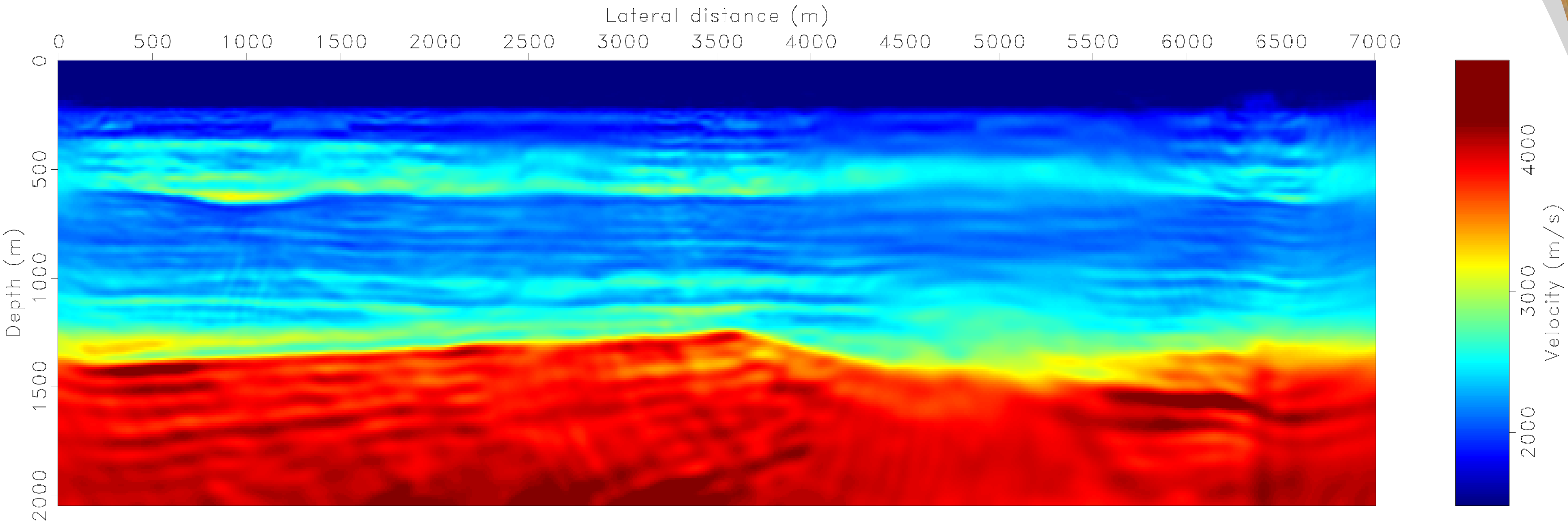
Results

Modified GN 7 *sequential* shots *with renewals*



Results

Modified GN 7 *sequential* shots *w/o renewals*



Migration 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

Migration 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

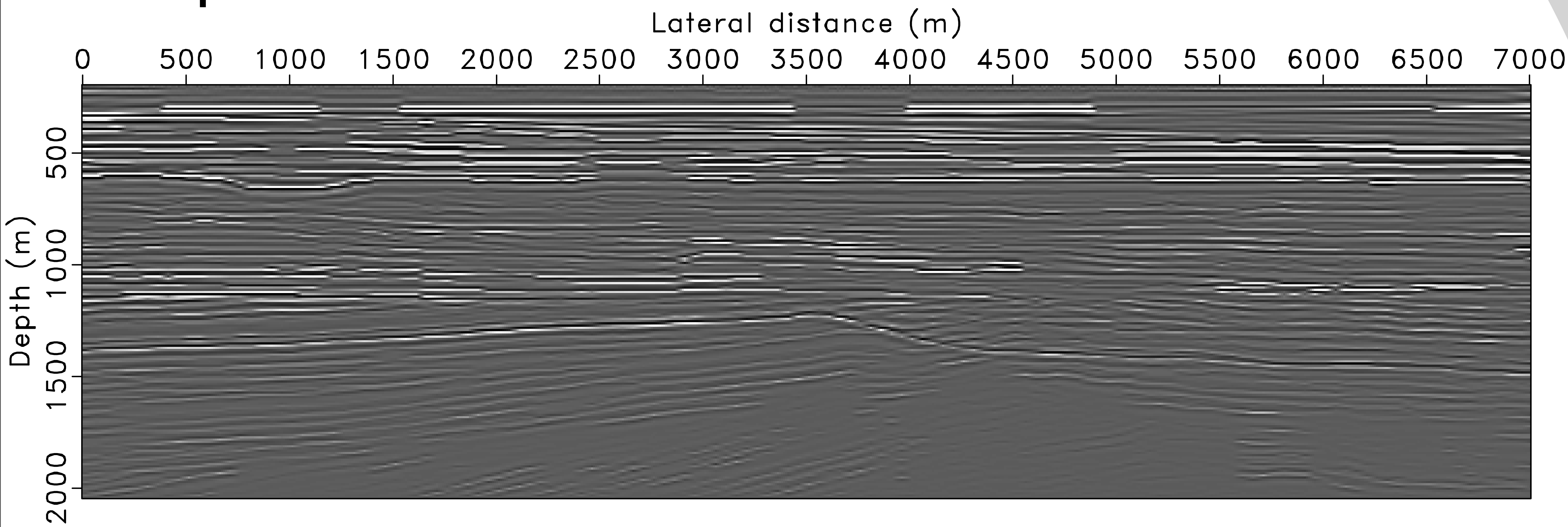
Migration:

- 10 random frequencies
  (20Hz-50Hz)
- 17 *simultaneous* shots (*versus 350 sequential shots*)
- LASSO problems determined by SPGL1

[Herrmann & Xi, 2011]

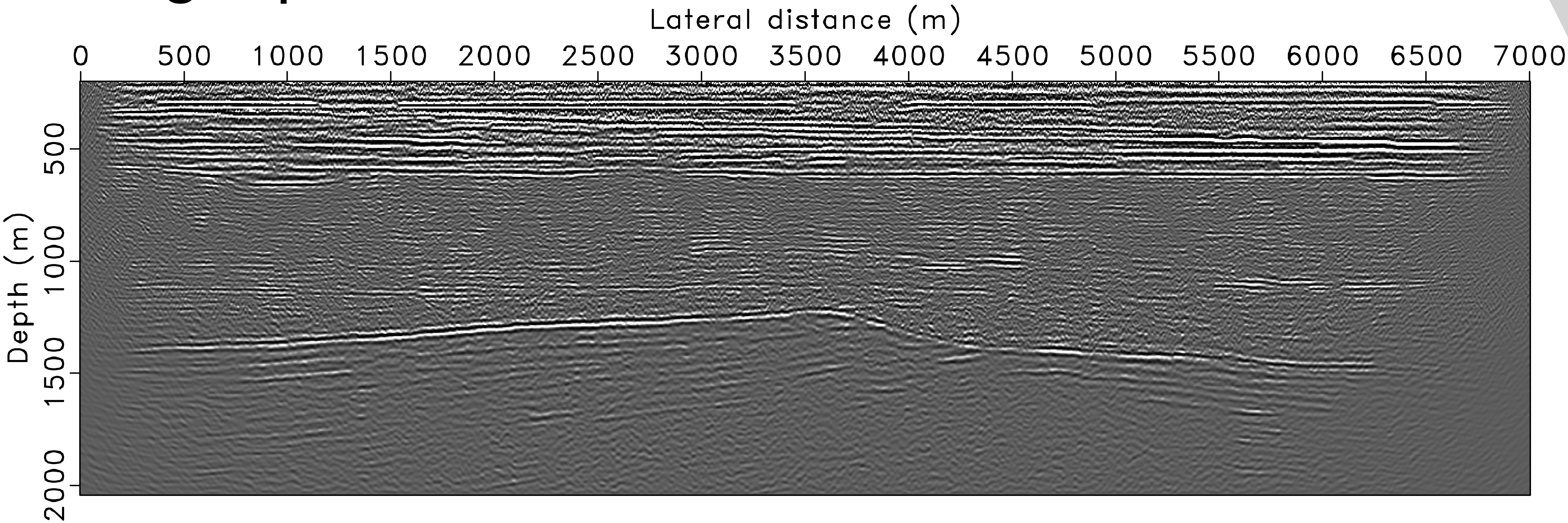
Migration results

true perturbation



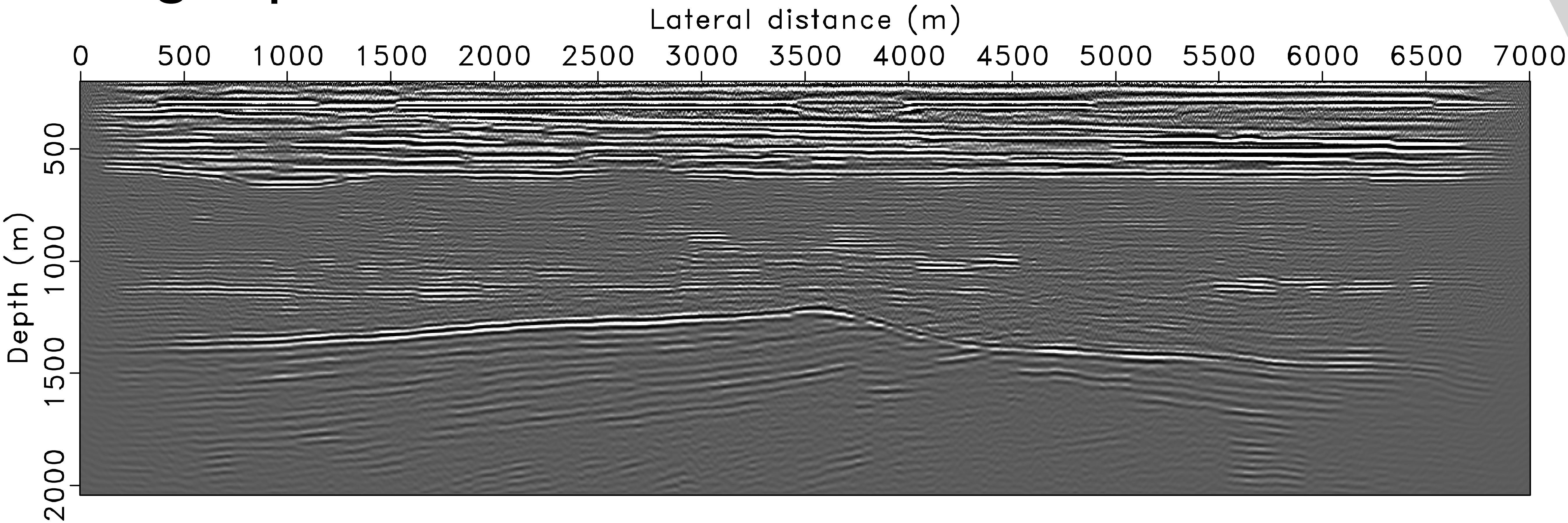
Migration results

imaged perturbation



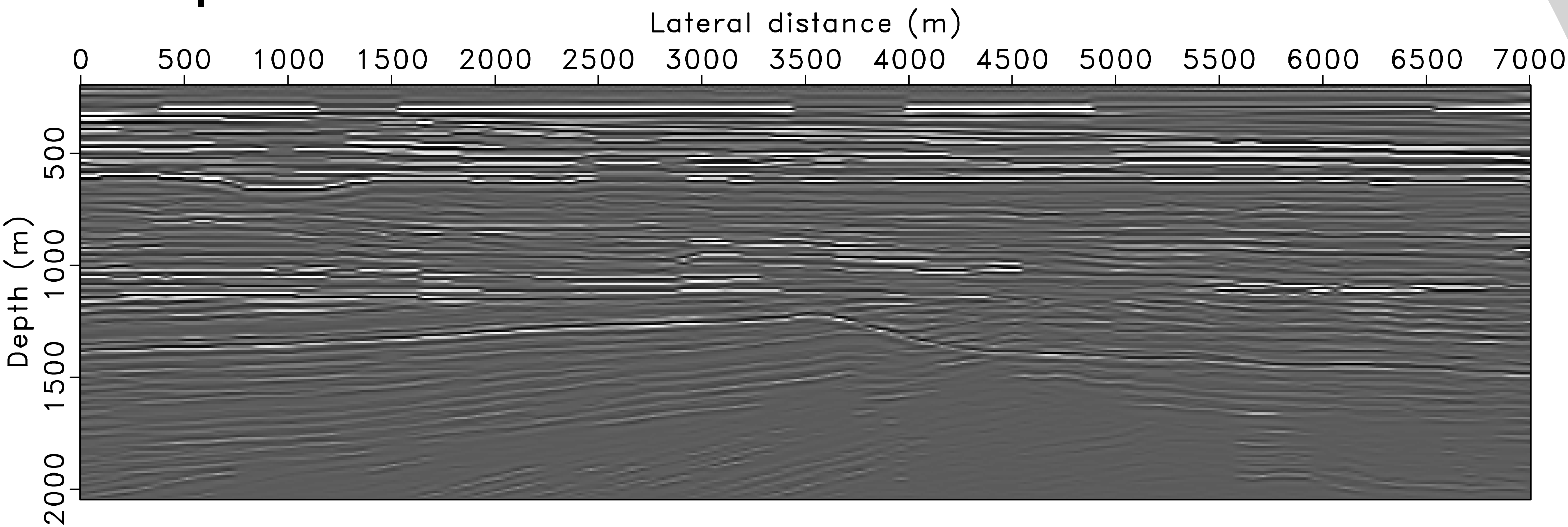
Migration results

imaged perturbation *with renewals*



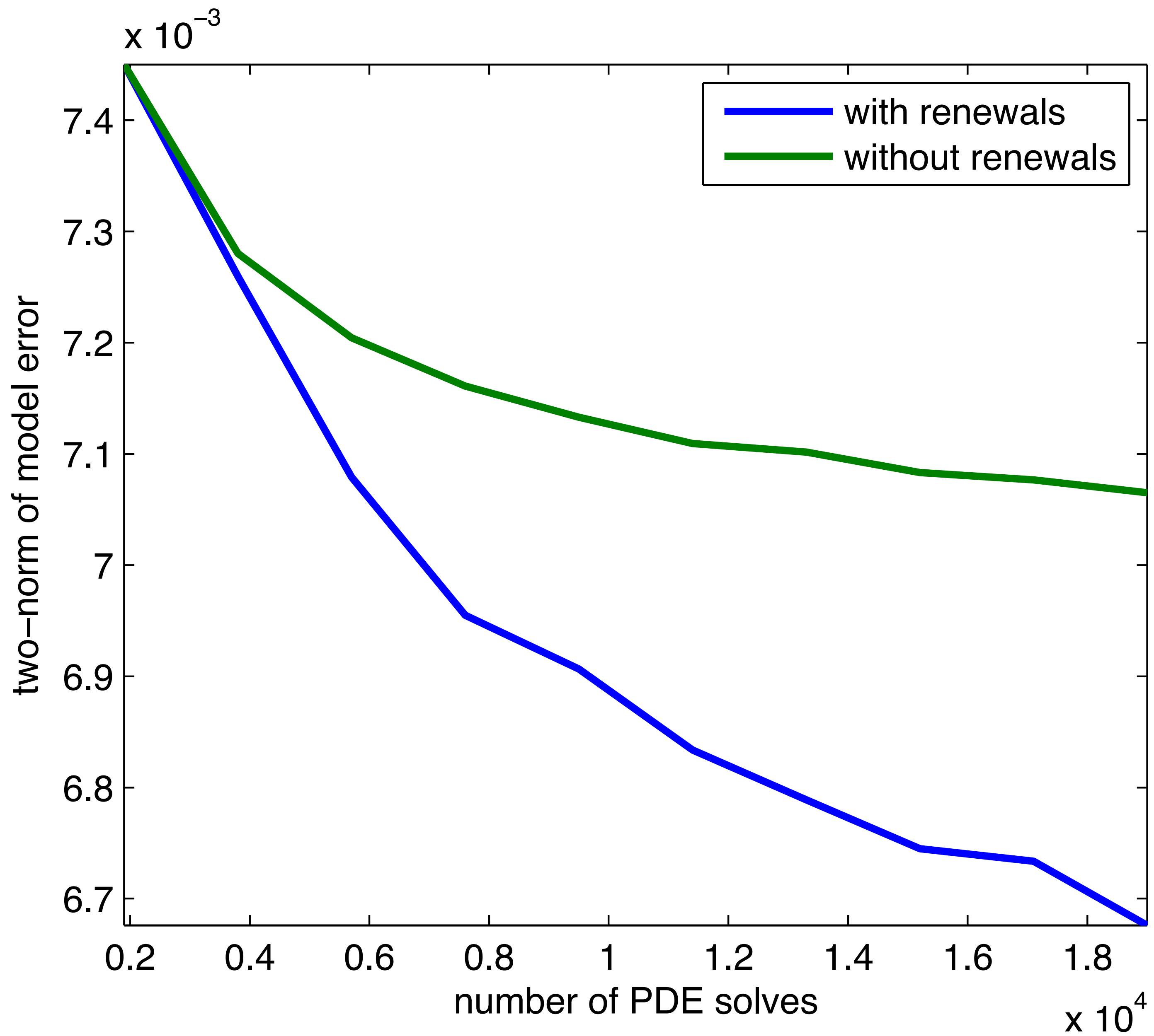
Migration results

true perturbation



Migration results

Model error



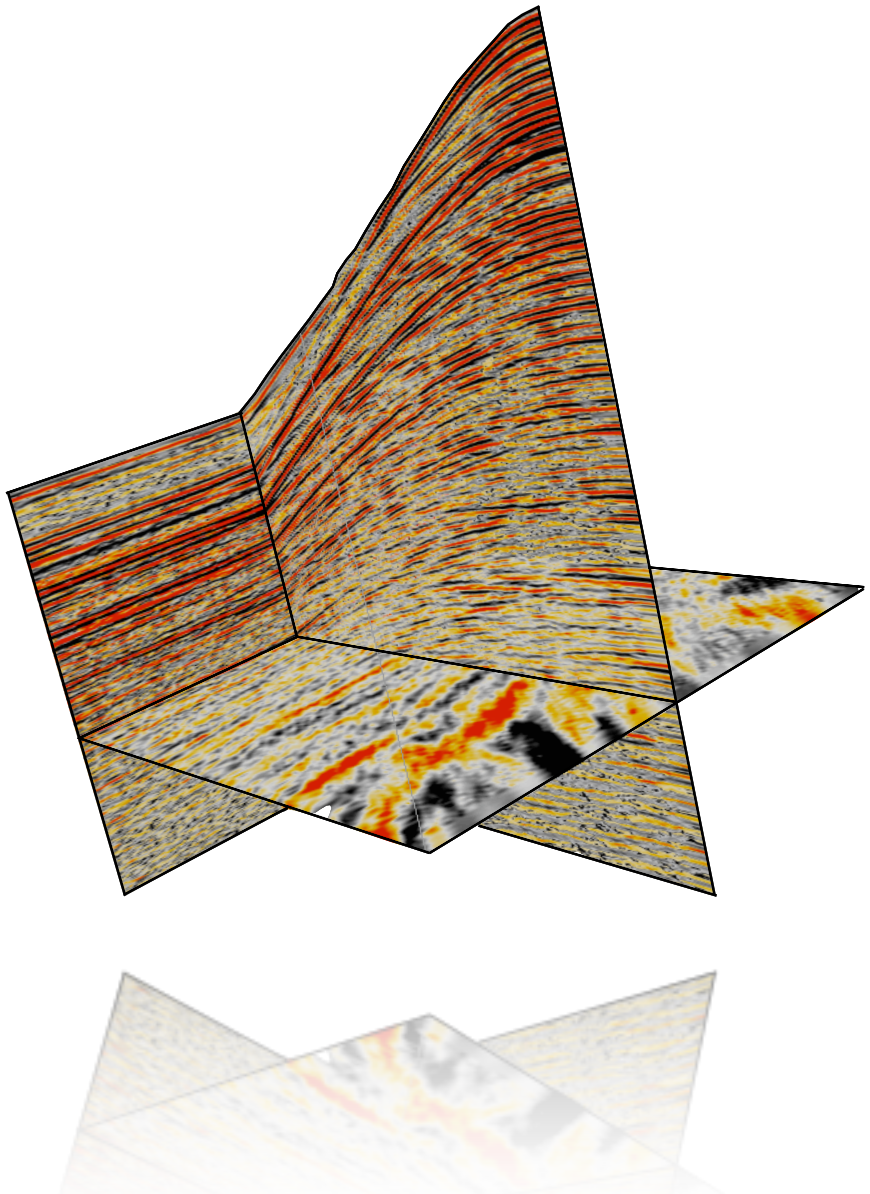
Key contributions

More *challenging* to implement extensions of FWI

- works with *randomized* sequential or simultaneous source experiments
- control of the error related to the randomized batches by sparsity promotion & batch size
- control over null space of the wave-equation Hessian

Challenge: upscale to *full 3D* but careful coordination with Mike will take care of that...

Sparsity inducing imaging with surface-related *multiples*



Key goals

Use information in *surface-related* multiples

- estimate the source function
- "fill in" missing data

Improve imaging results by

- *exploiting* transform-domain sparsity
- incorporating *physics* in the formulation

SLIM

Key strategies

Use *sparsity* promotion to stabilize *wavefield* inversion

Combine with *sparsity* promoting *imaging*

Use randomized *dimensionality* reduction

EPSI Model

SLIM

Estimation of Primaries by Sparse Inversion (van Groenestijn and Verschuur, 2009)

recorded data   predicted data from primary IR

$$\mathbf{P} = \mathbf{G}(\mathbf{Q} + \mathbf{R}\mathbf{P})$$

$\mathbf{P}$ total recorded up-going wavefield

$\mathbf{Q}$ source signature (incl. src ghosts)

$\mathbf{R}$ reflectivity of free surface (assume -1)

$\mathbf{G}$ primary impulse response

(all monochromatic data matrix, implicit ω)

EPSI Inversion

Estimation of Primaries by Sparse Inversion (van Groenestijn and Verschuur, 2009)

recorded data predicted data from primary IR

$$\mathbf{P} = \mathbf{G}(\mathbf{Q} + \mathbf{R}\mathbf{P})$$

Inversion objective:

$$f(\mathbf{G}, \mathbf{Q}) = \frac{1}{2}\|\mathbf{P} - \mathbf{G}(\mathbf{Q} + \mathbf{R}\mathbf{P})\|_2^2$$

+ *Sparse* inversion

Convolution model

SLIM

Convolution Model

$$\text{Up-going Primary} = \mathbf{GQ}$$

EPSI Model

$$\text{Up-going Primary} + \text{Multiples} = \mathbf{GQ} + \mathbf{GRP}$$

additional info on G

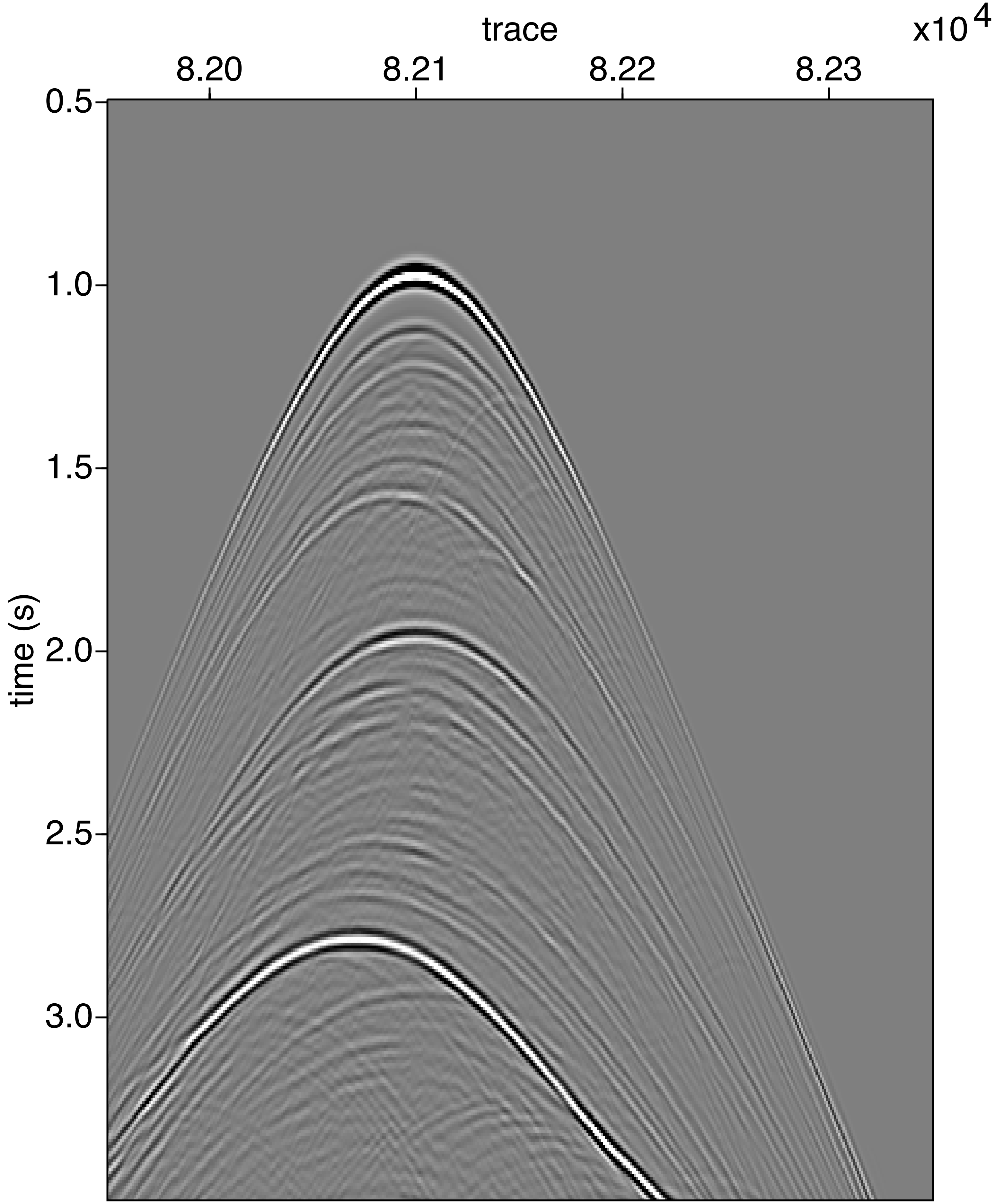
$\mathbf{P}$ total recorded up-going wavefield

$\mathbf{Q}$ source signature (incl. src ghosts)

$\mathbf{R}$ reflectivity of free surface (assume -1)

$\mathbf{G}$ primary impulse response

(all monochromatic data matrix, implicit ω)

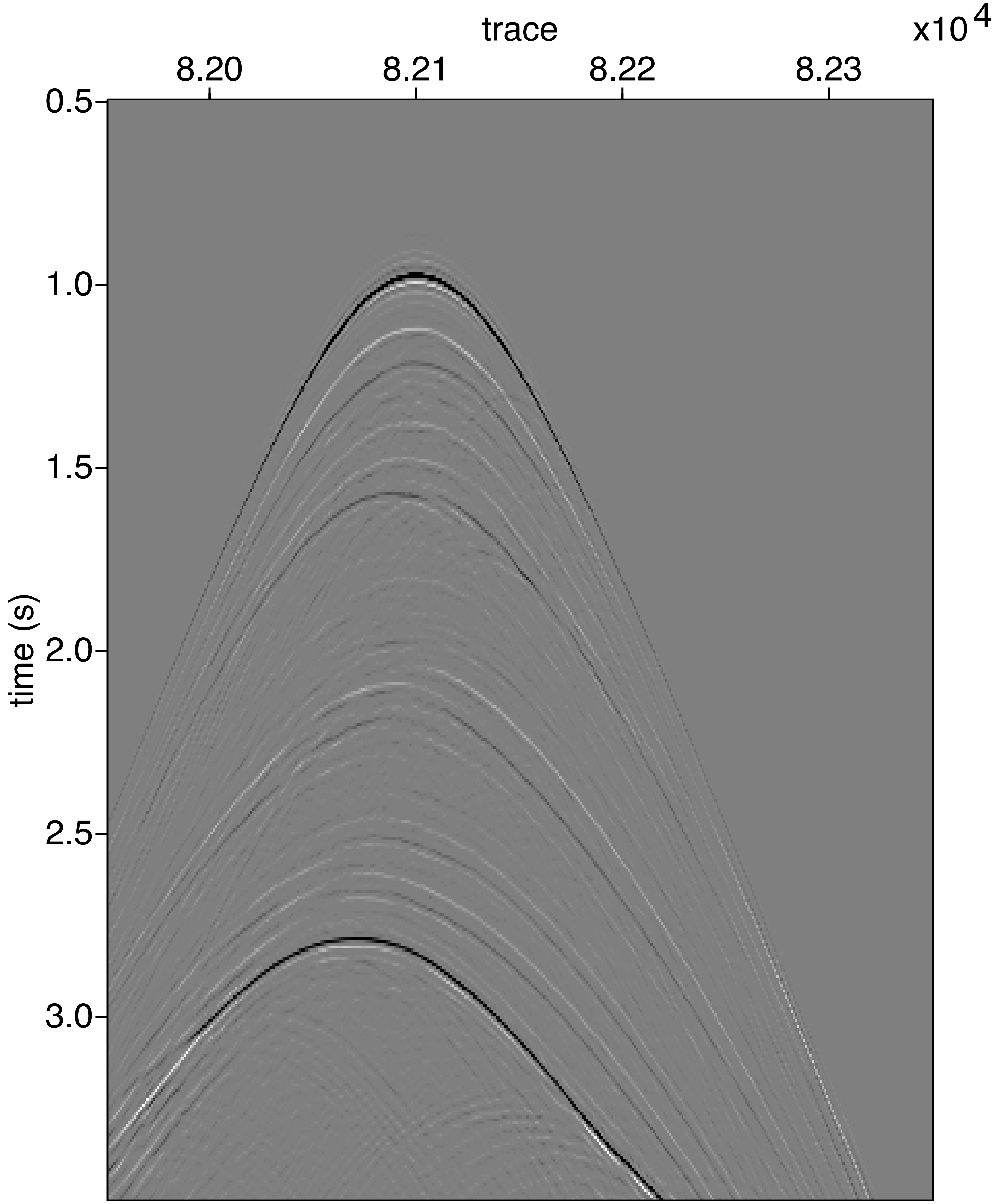


Pluto15
data

Elastic FD Modeling

muted

no deghosting



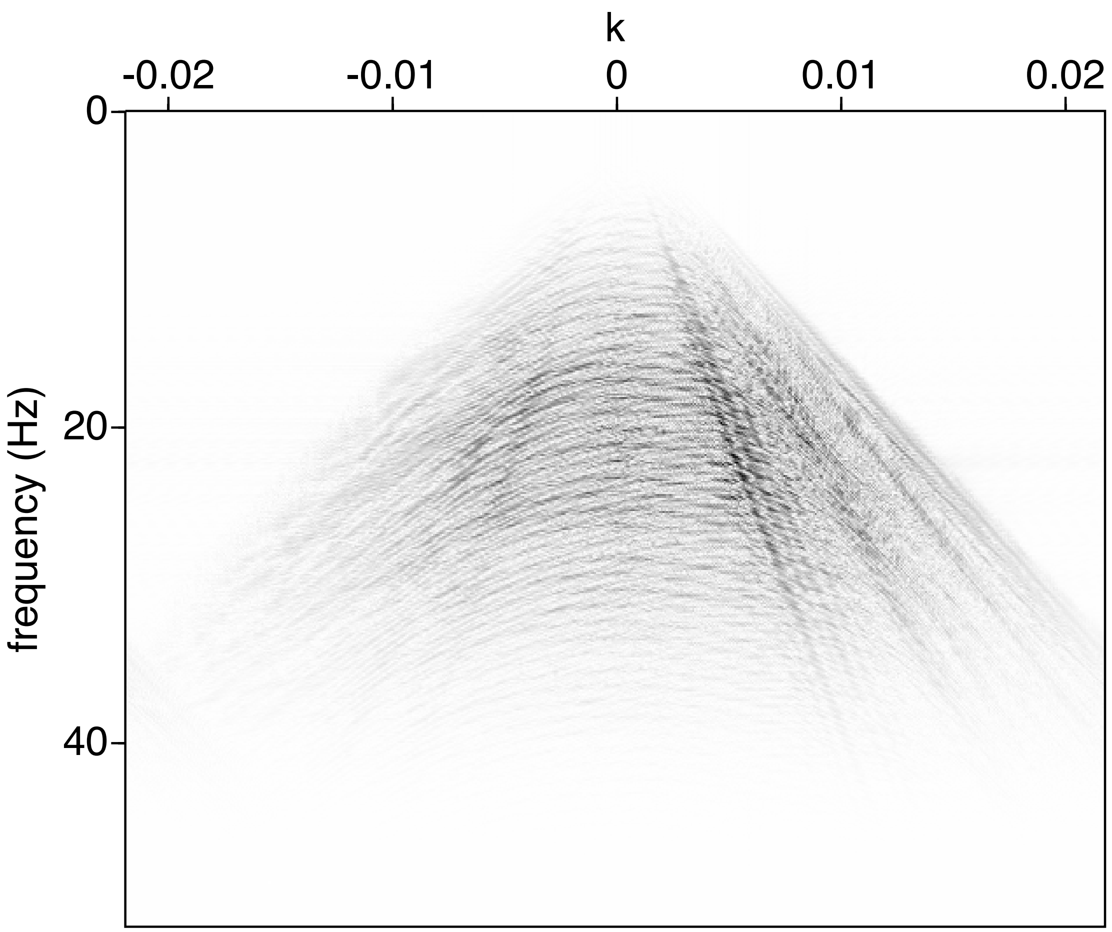
Pluto15

REPSI

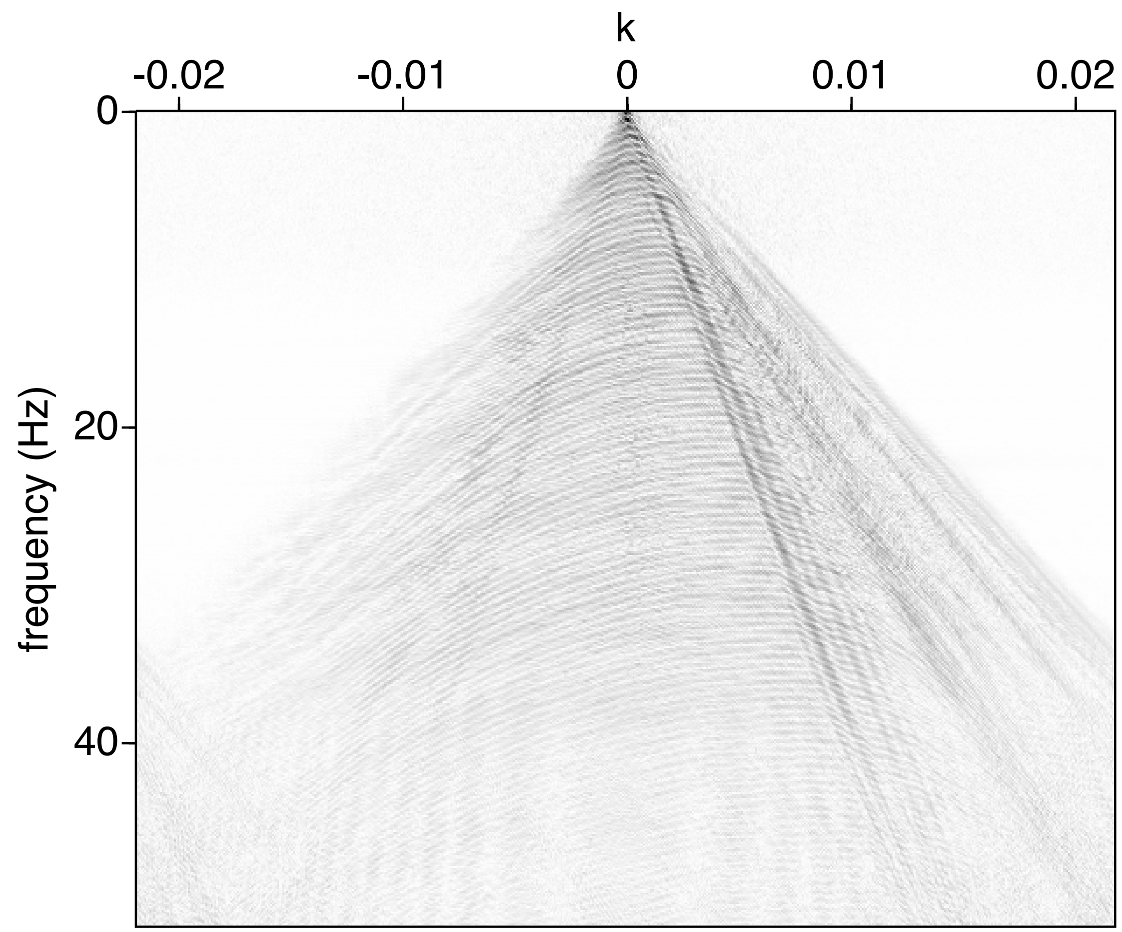
Primary IR (G)

no transform used

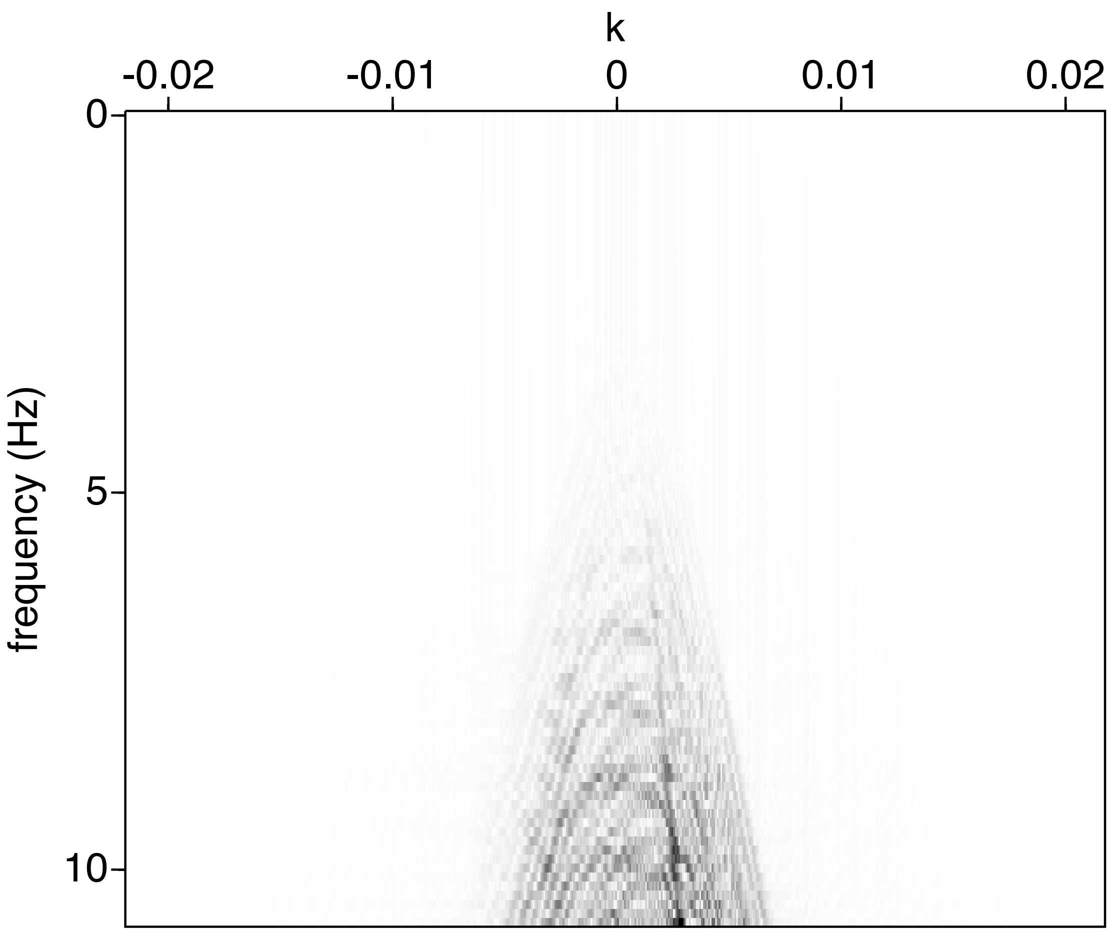
80 iters



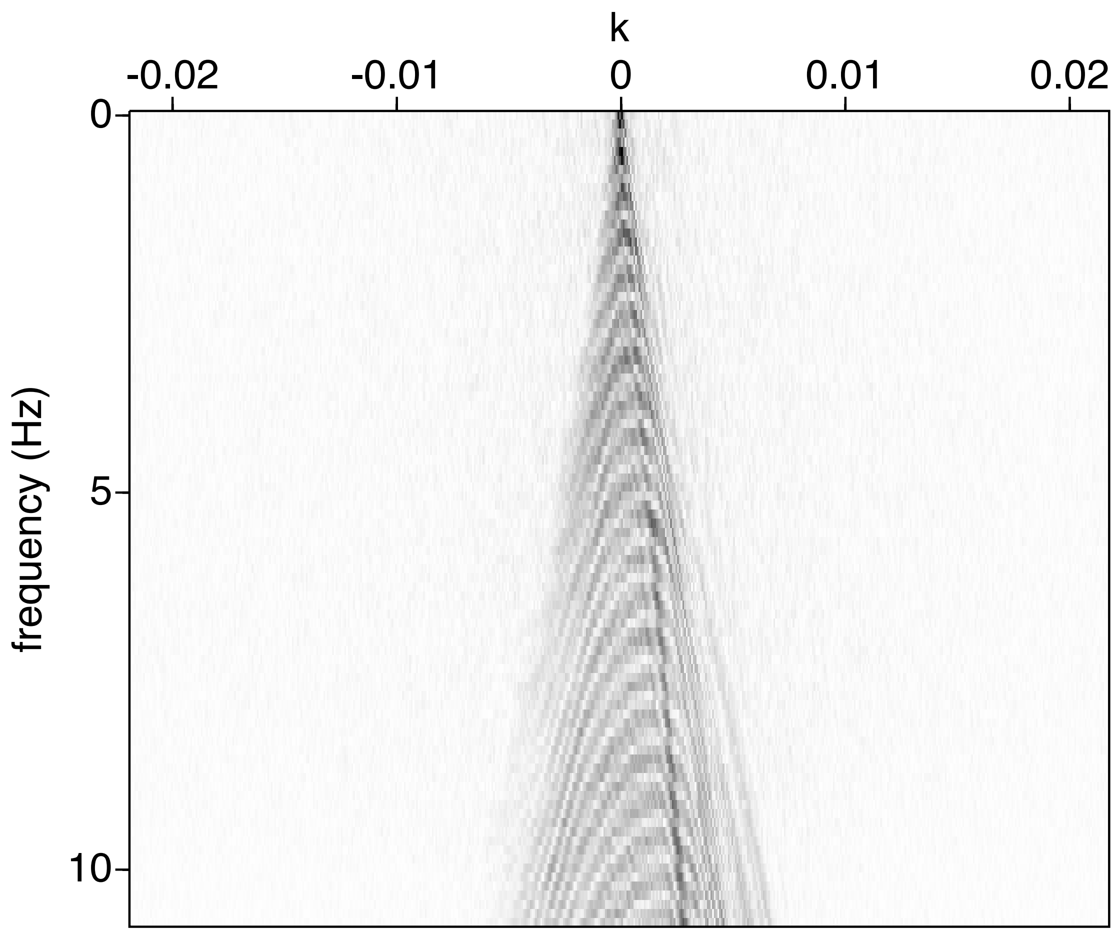
F-K Spectrum of data



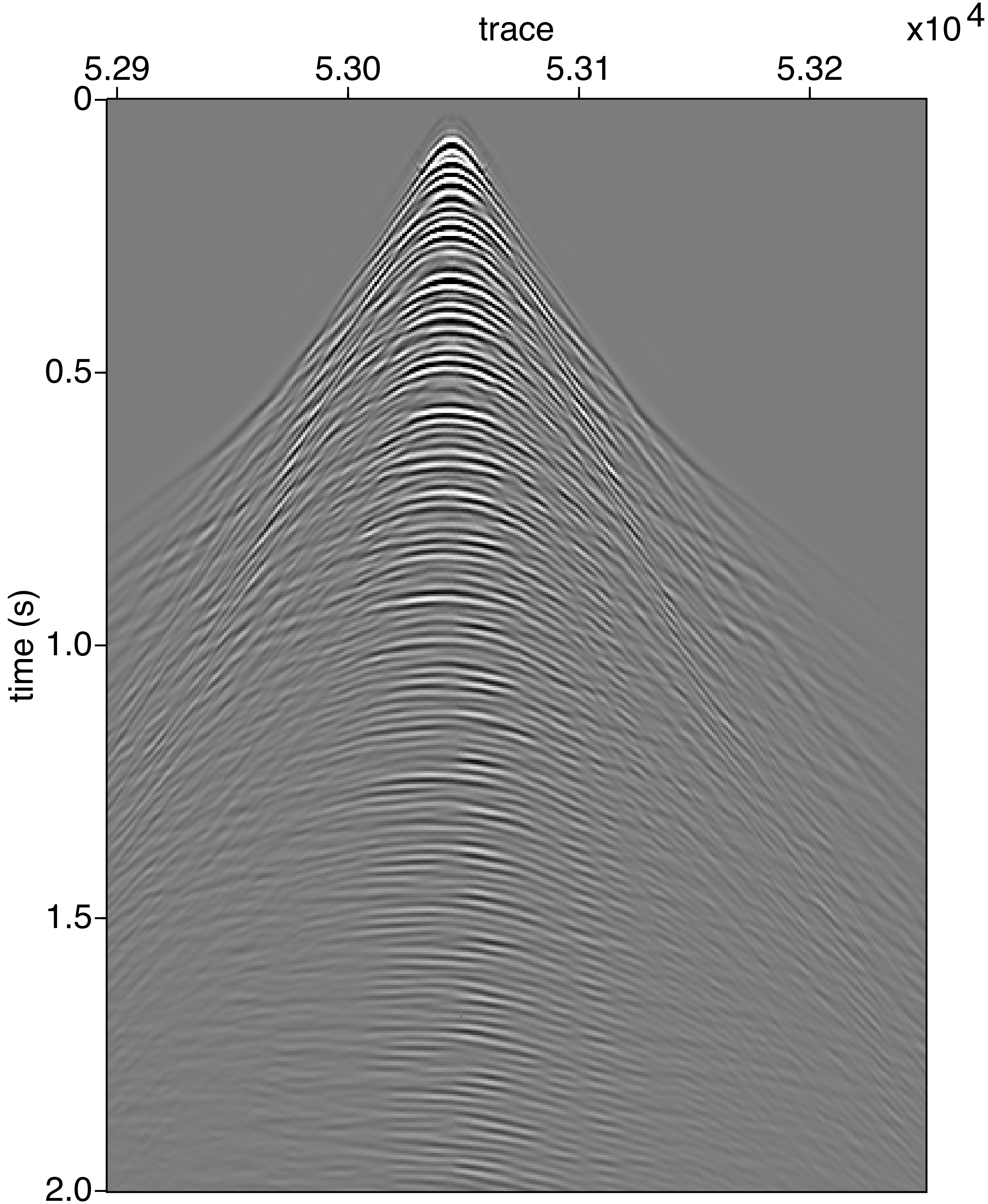
F-K Spectrum of REPSI Primary IR



F-K Spectrum of data



F-K Spectrum of REPSI Primary IR



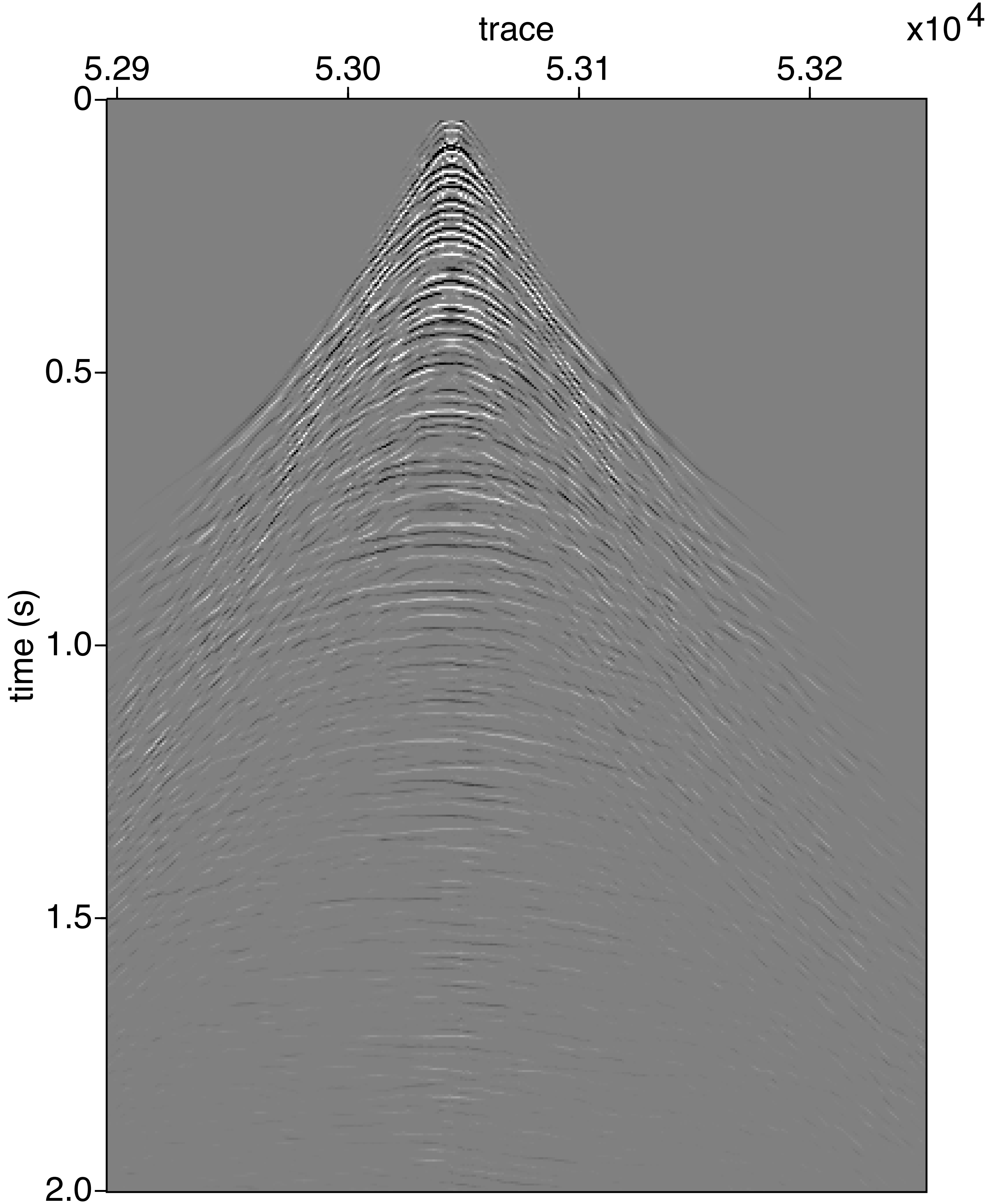
Gulf of Suez
data

shot gather

interpolated, muted

reciprocity

no deghosting



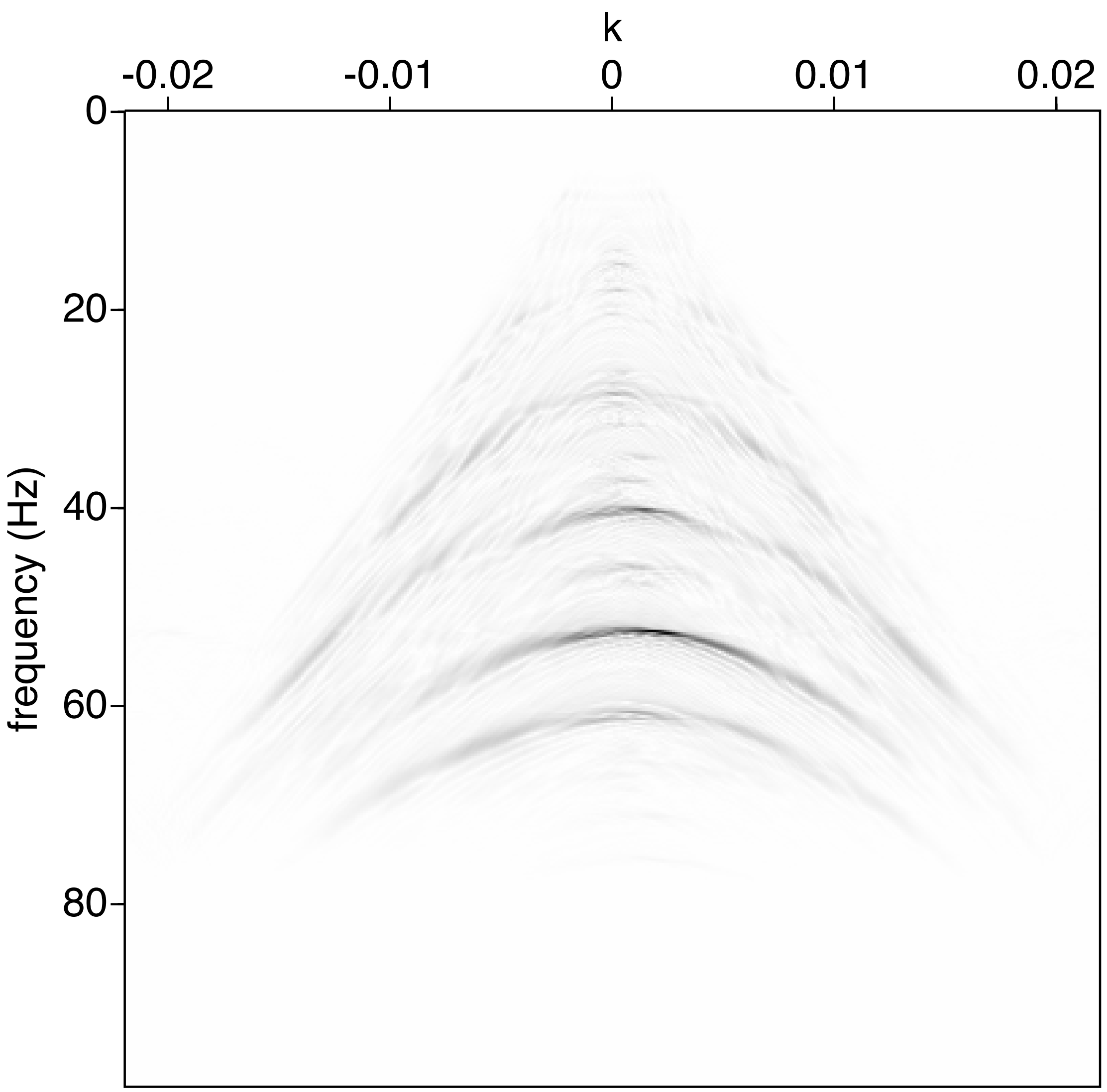
Gulf of Suez

REPSI

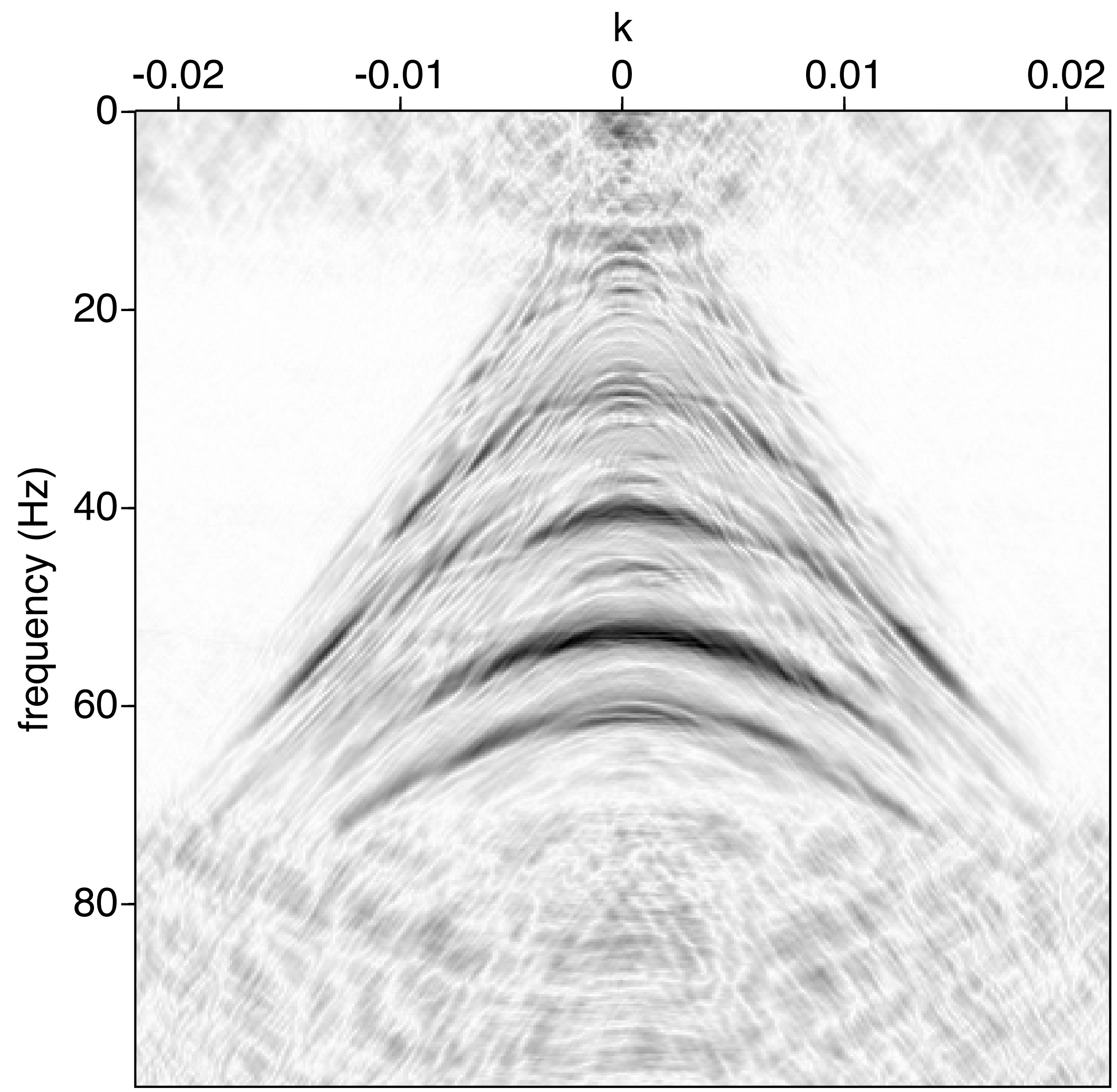
Primary IR (G)

shot gather

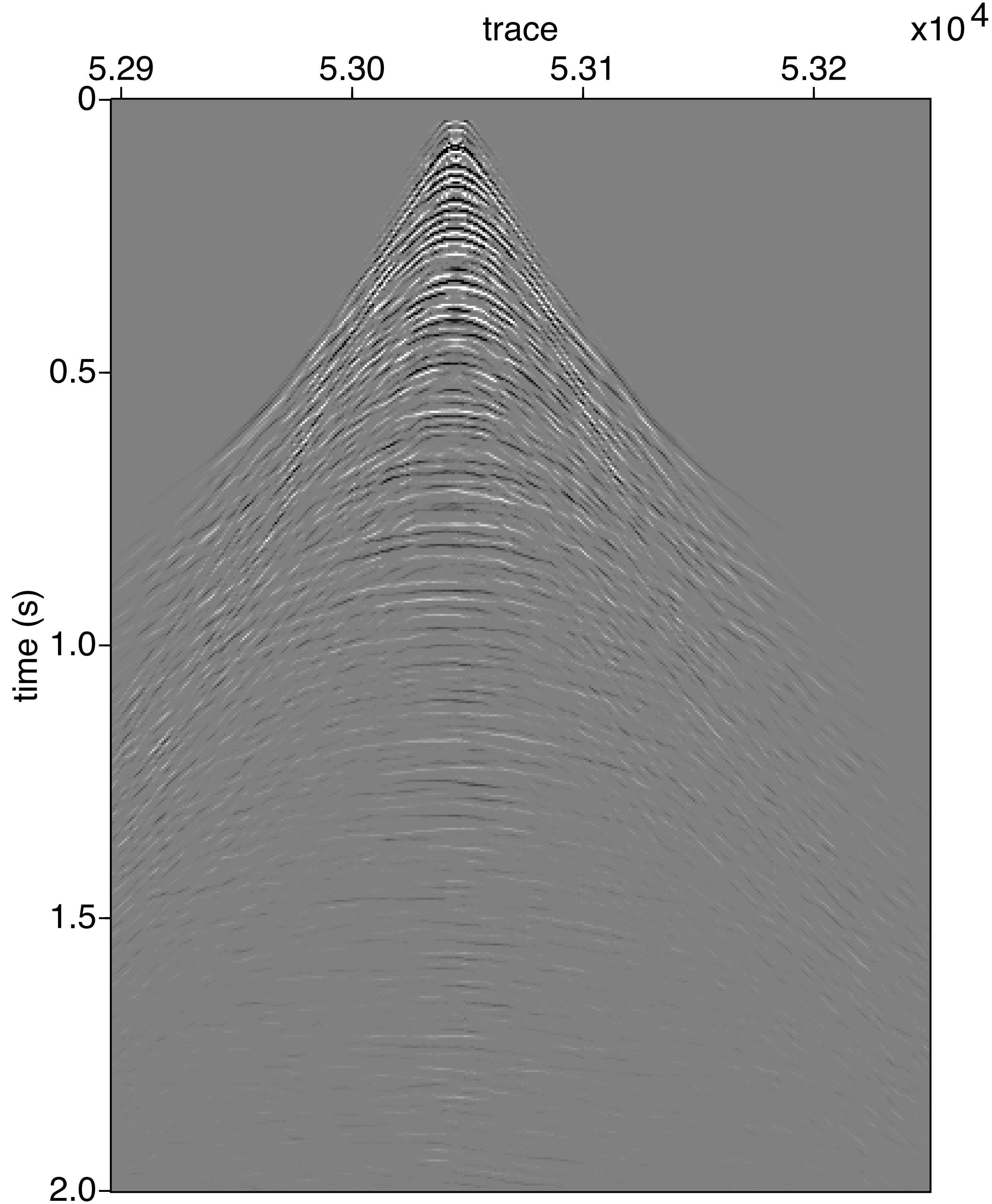
80 SPG grad. iterations



F-K Spectrum of data



F-K Spectrum of REPSI+Transform Primary IR



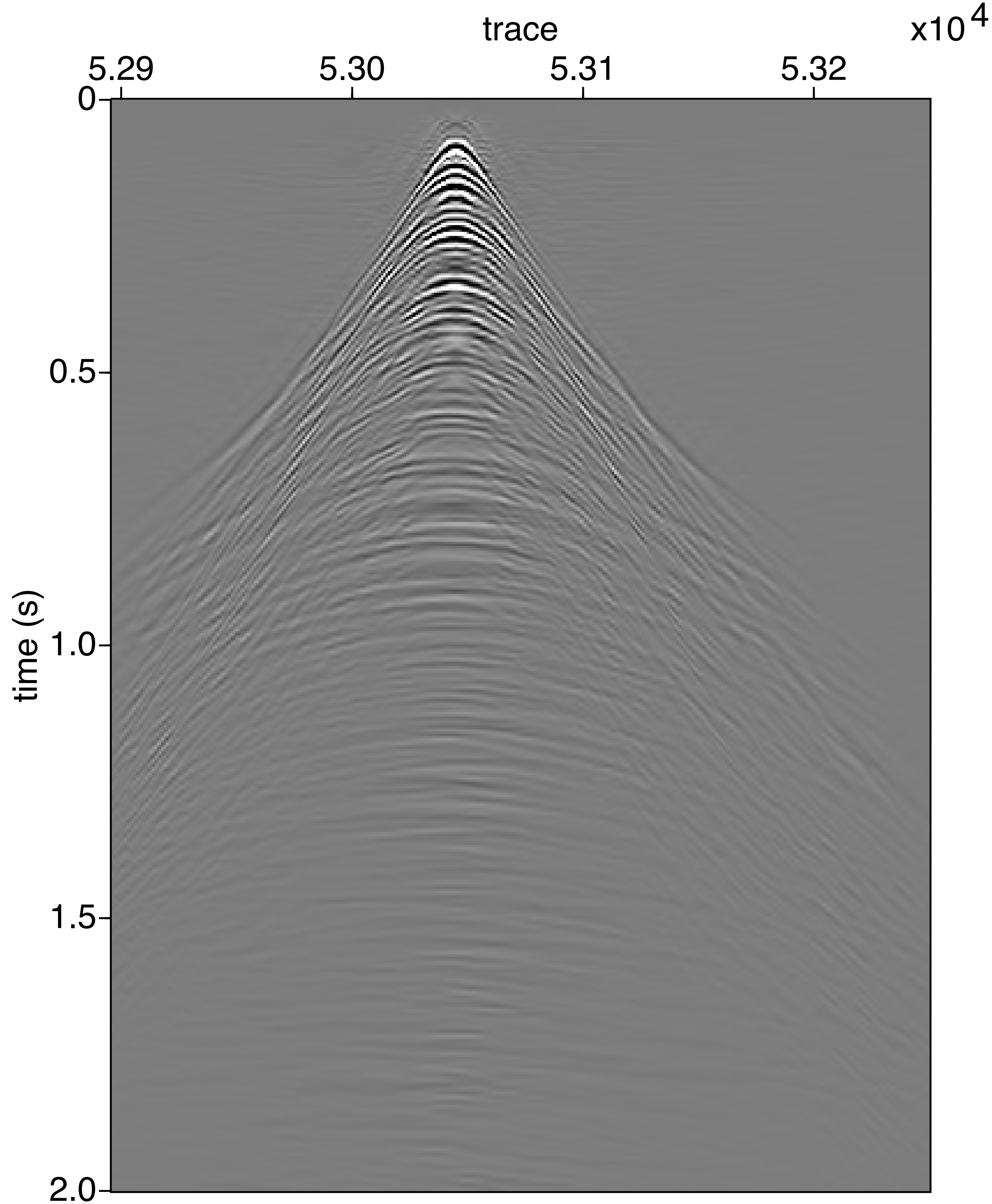
Gulf of Suez

REPSI

Primary IR (G)

shot gather

80 SPG grad. iterations



Gulf of Suez

REPSI + Transform

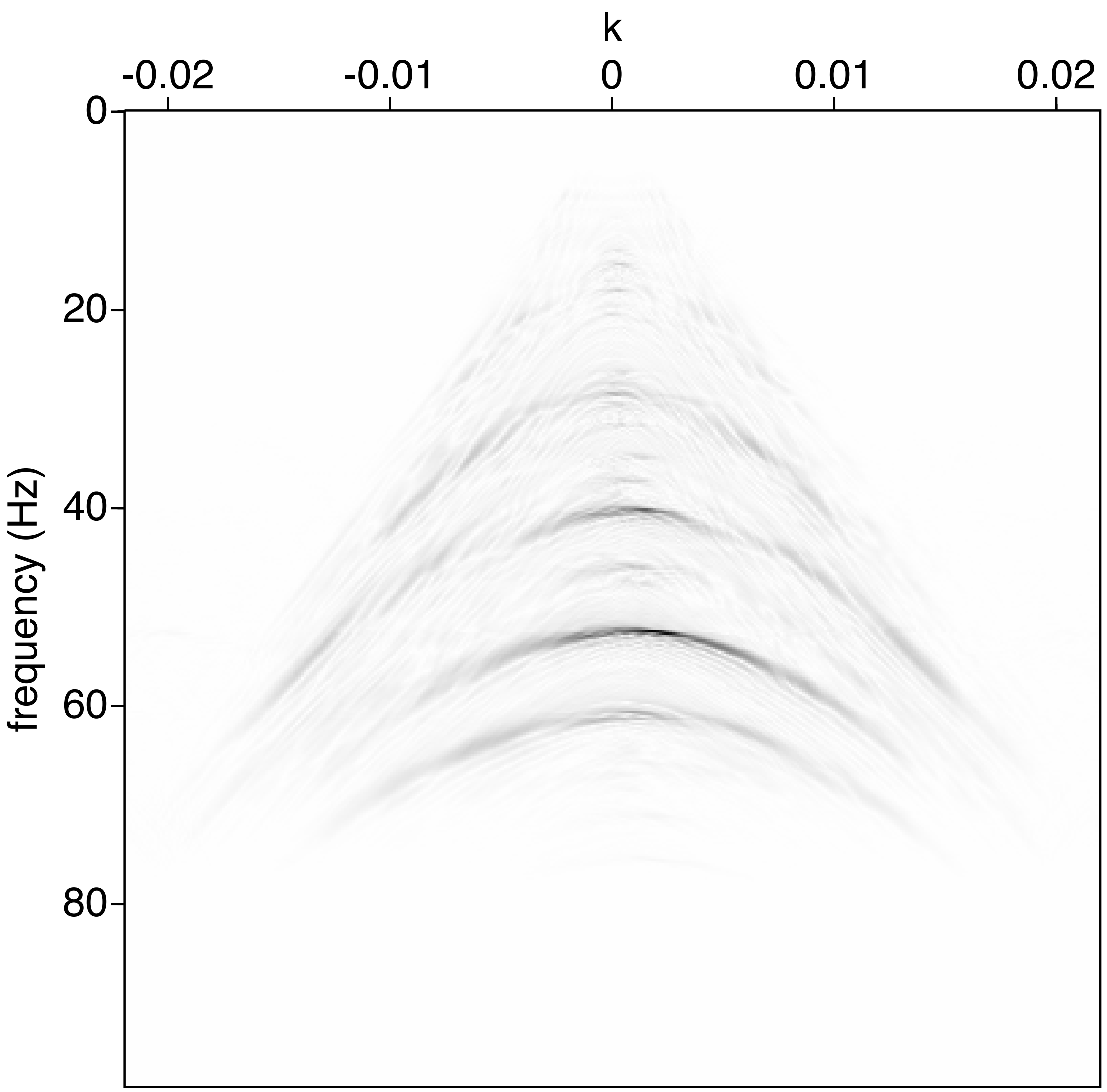
Primary IR (G)

shot gather

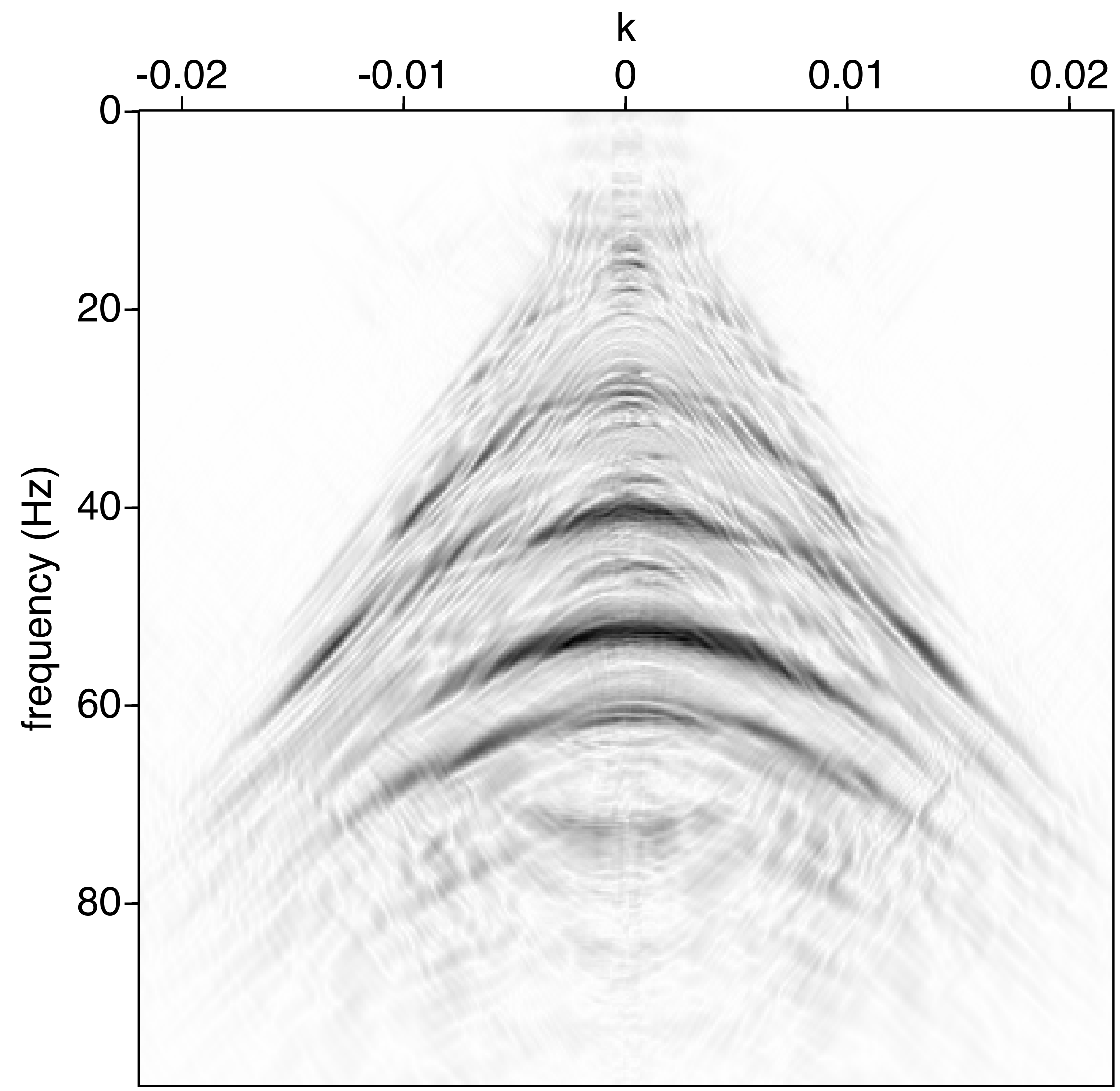
2D Curvelet (Src-Rcv)

Spline a=3.0 DWT (Time)

90 SPG grad. iterations

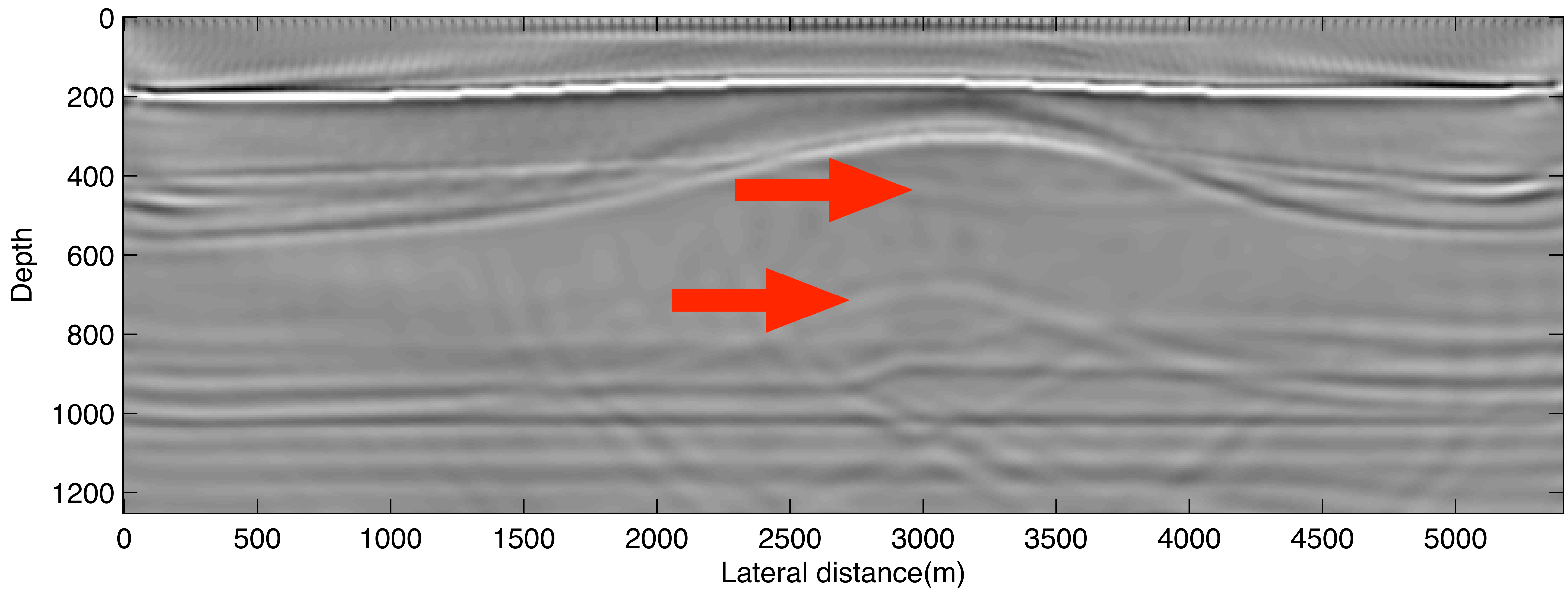


F-K Spectrum of data

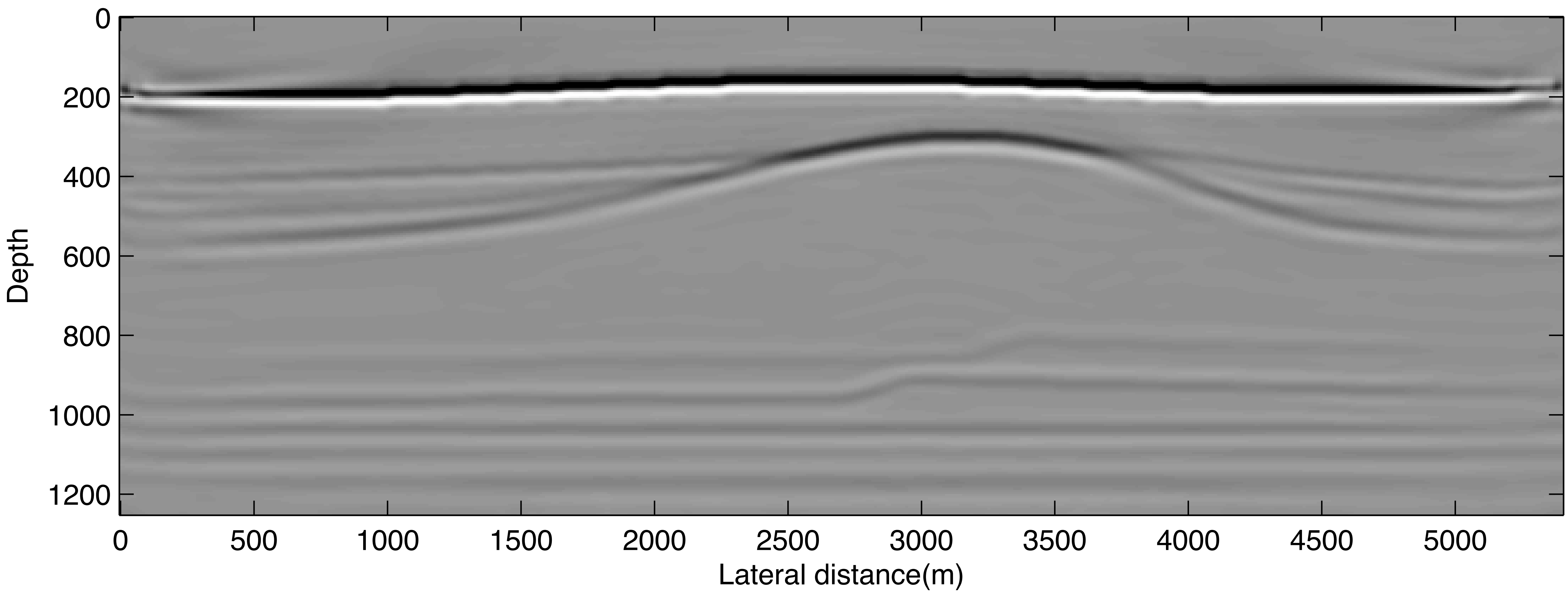


F-K Spectrum of REPSI+Transform Primary IR

Sparse inversion of data with multiples



Sparse inversion of data with multiples with EPSI



EPSI problem

recorded data

predicted data

$$\hat{\mathbf{P}} = \hat{\mathbf{G}}(\hat{\mathbf{Q}} - \hat{\mathbf{P}})$$

$\hat{\mathbf{P}}$ “low-rank” approximation (known)

$\hat{\mathbf{Q}}$ full-rank diagonal matrix (known)

$\hat{\mathbf{R}}$ assume $-\mathbf{I}$

$\hat{\mathbf{G}}$ unknown

Dimensionality-reduction via SVD

Approximate data matrix $\hat{\mathrm{P}}$ with *low-rank* factorization:

$$\hat{\mathbf{P}} = \hat{\mathbf{G}}(\hat{\mathbf{Q}} - \hat{\mathbf{P}})$$

$$\hat{\mathbf{P}} \approx \mathbf{U \Sigma V}^*$$

$\mathbf{U_{n_r \times k}}$ left singular vectors

$\mathbf{\Sigma_{k \times k}}$ singular values

$\mathbf{V_{n_s \times k}}$ right singular vectors

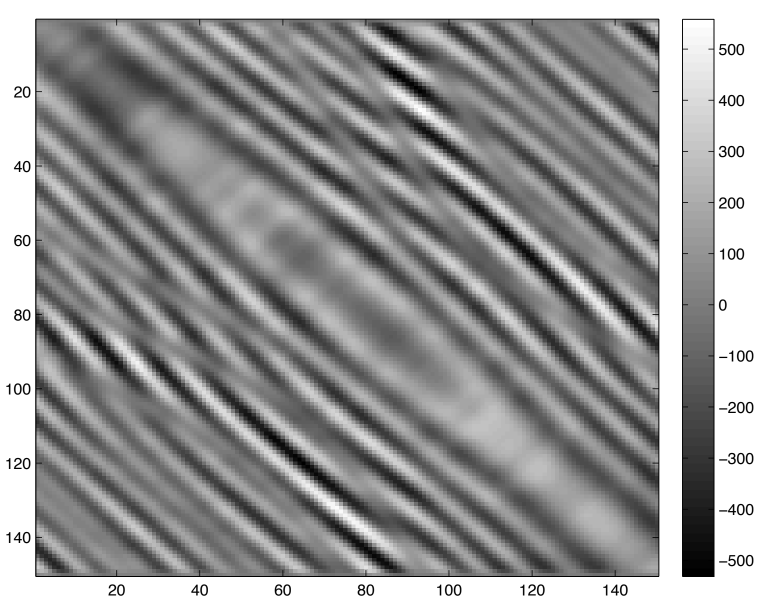
$\mathrm{k}:$ approximate rank

$k << min(n_r, n_s)$

Dimensionality Reduction Via SVD

Approximate data matrix $\hat{\mathbf{P}}$ with *low-rank* factorization:

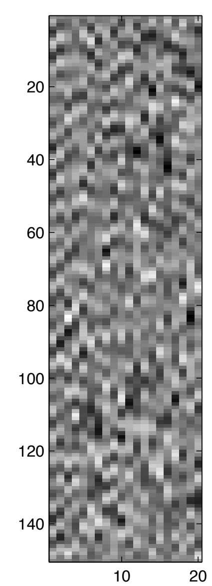
$\hat{\mathbf{P}}$



$n_r \times n_s$

$\approx$

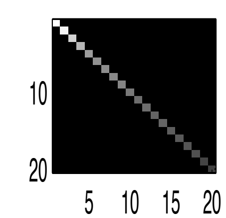
$\mathbf{U}$



$n_r \times k$

$*$

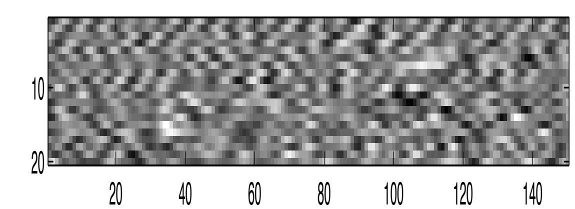
$\mathbf{\Sigma}$



$k \times k$

$*$

$\mathbf{V}^*$



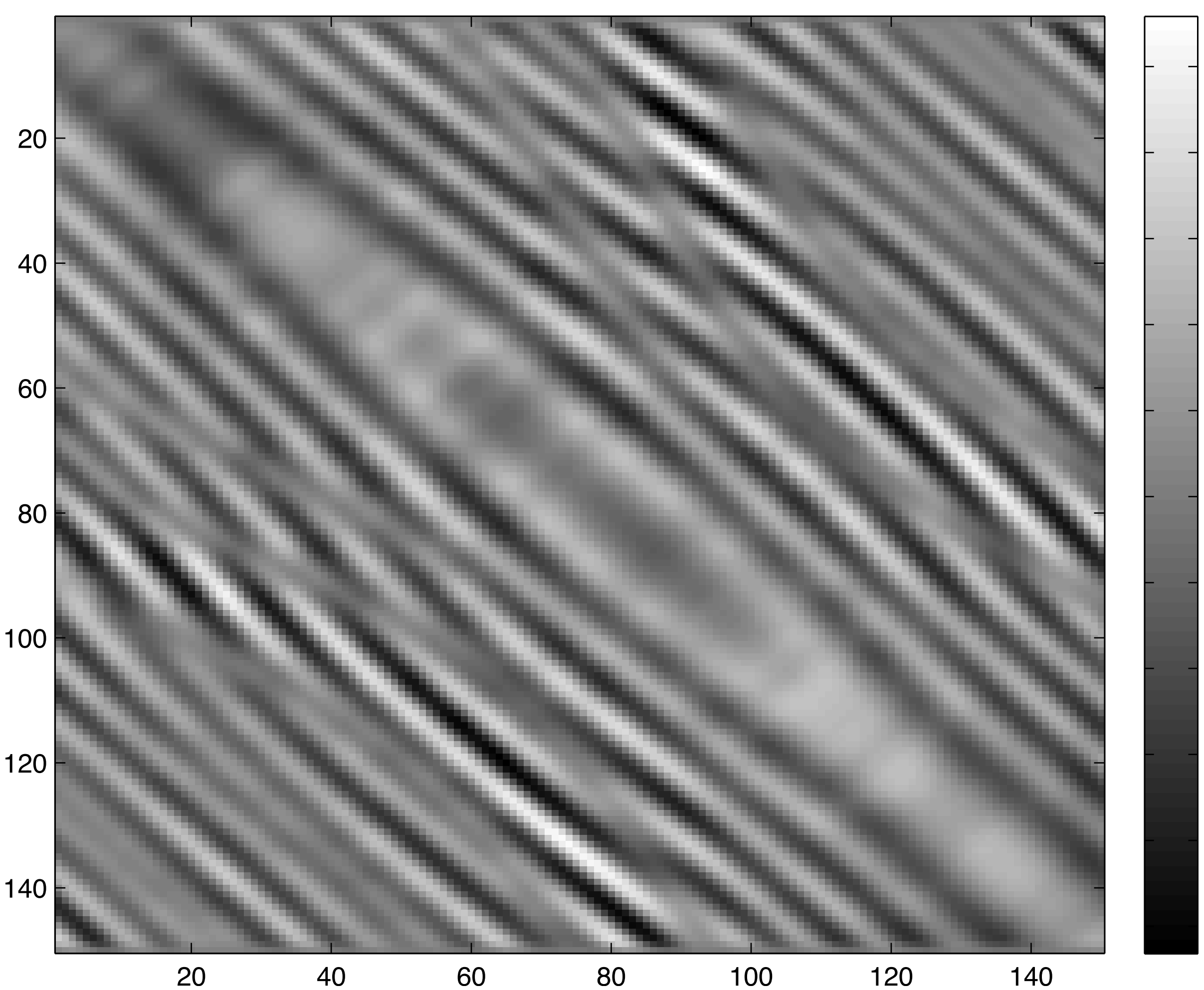
$k \times n_s$

$\mathrm{k\ :\ approximate\ rank}$

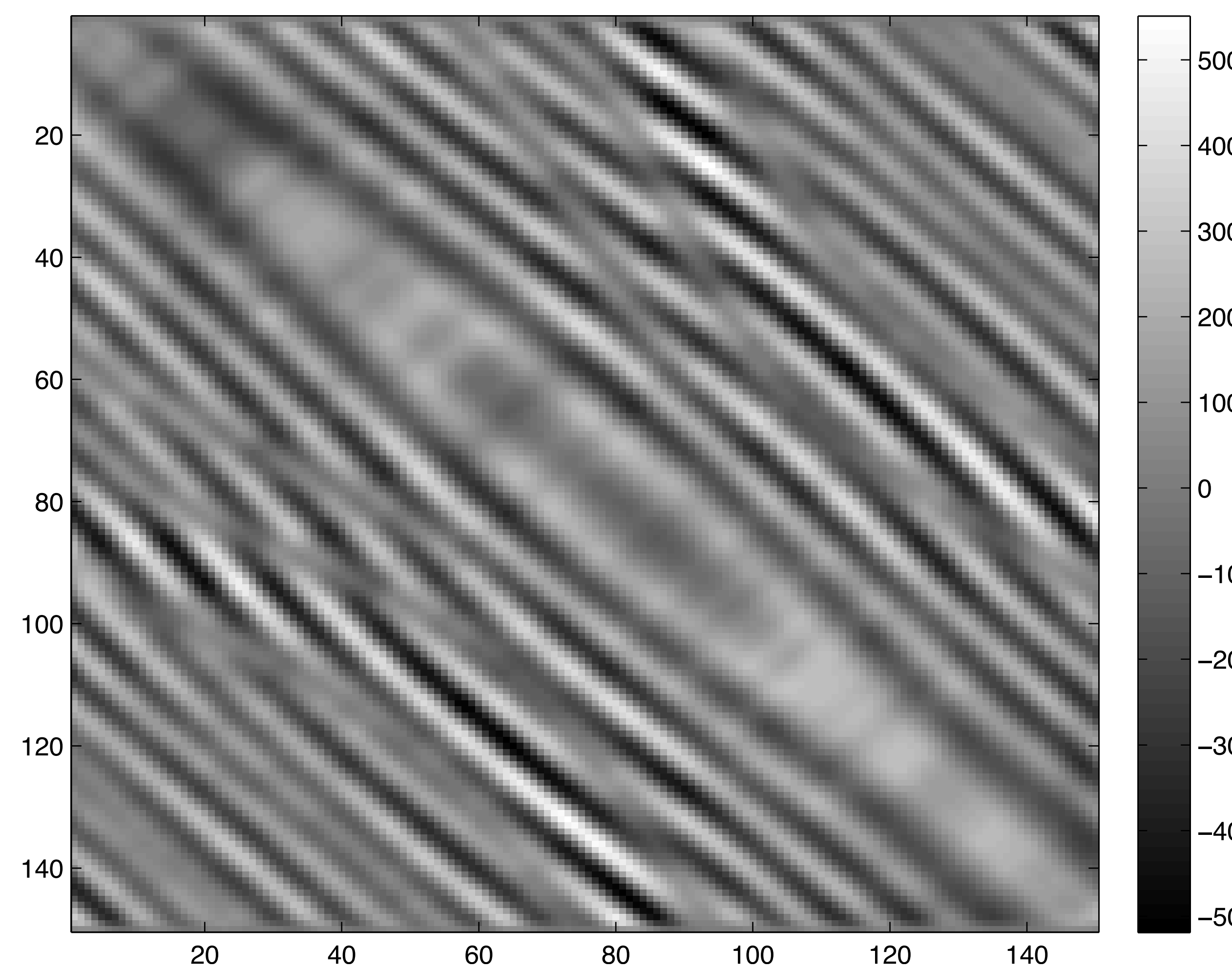
$k << min(n_r, n_s)$

Full vs approximated data

$\hat{\mathbf{P}}$

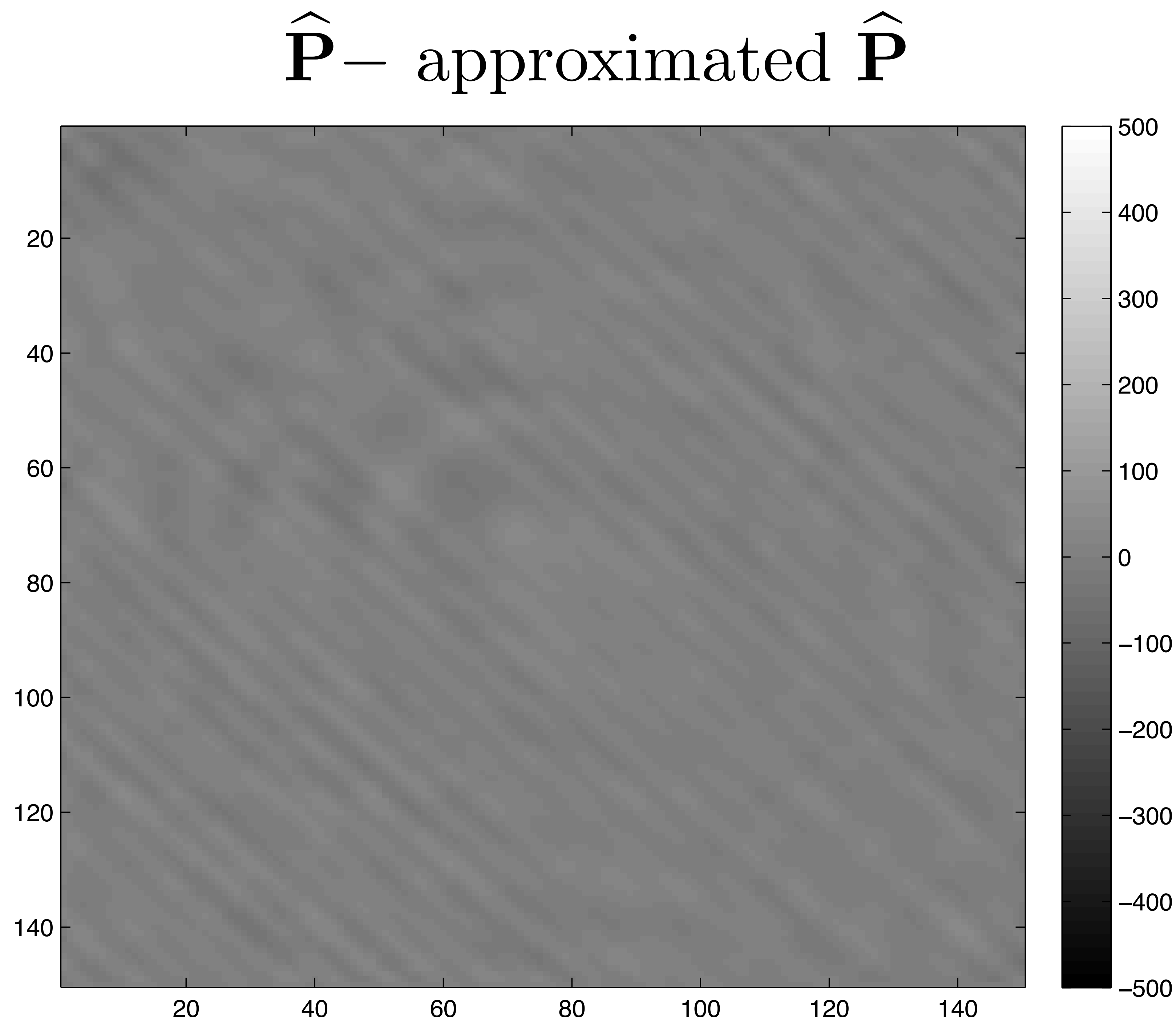


Approximated $\hat{\mathbf{P}}$



$$n_s = n_r = 150$$
$$k = 20 = 14\%$$
$$SNR = 16dB$$

Full vs approximated data

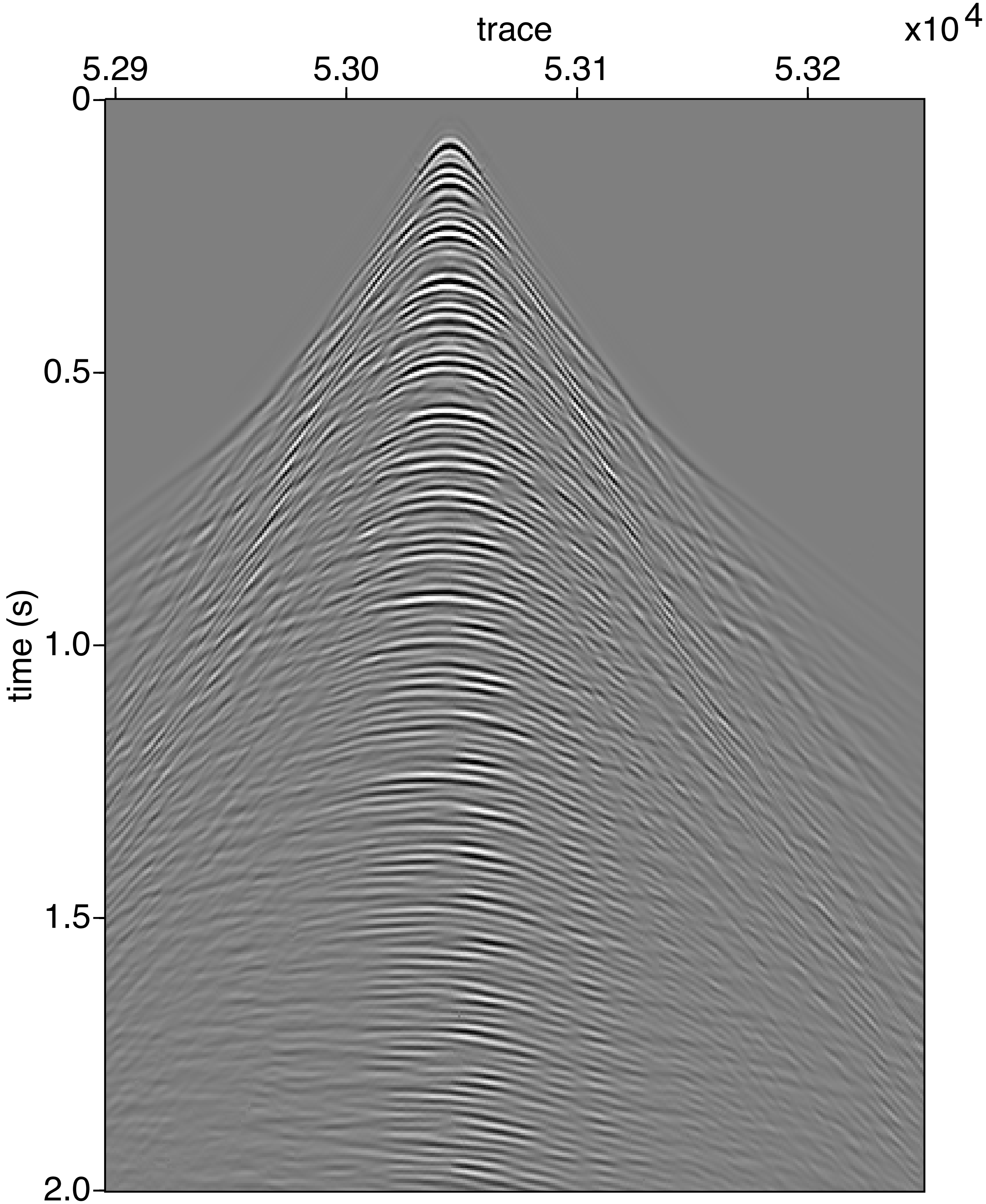


Multiplication speed up

7.5 x

Memory usage

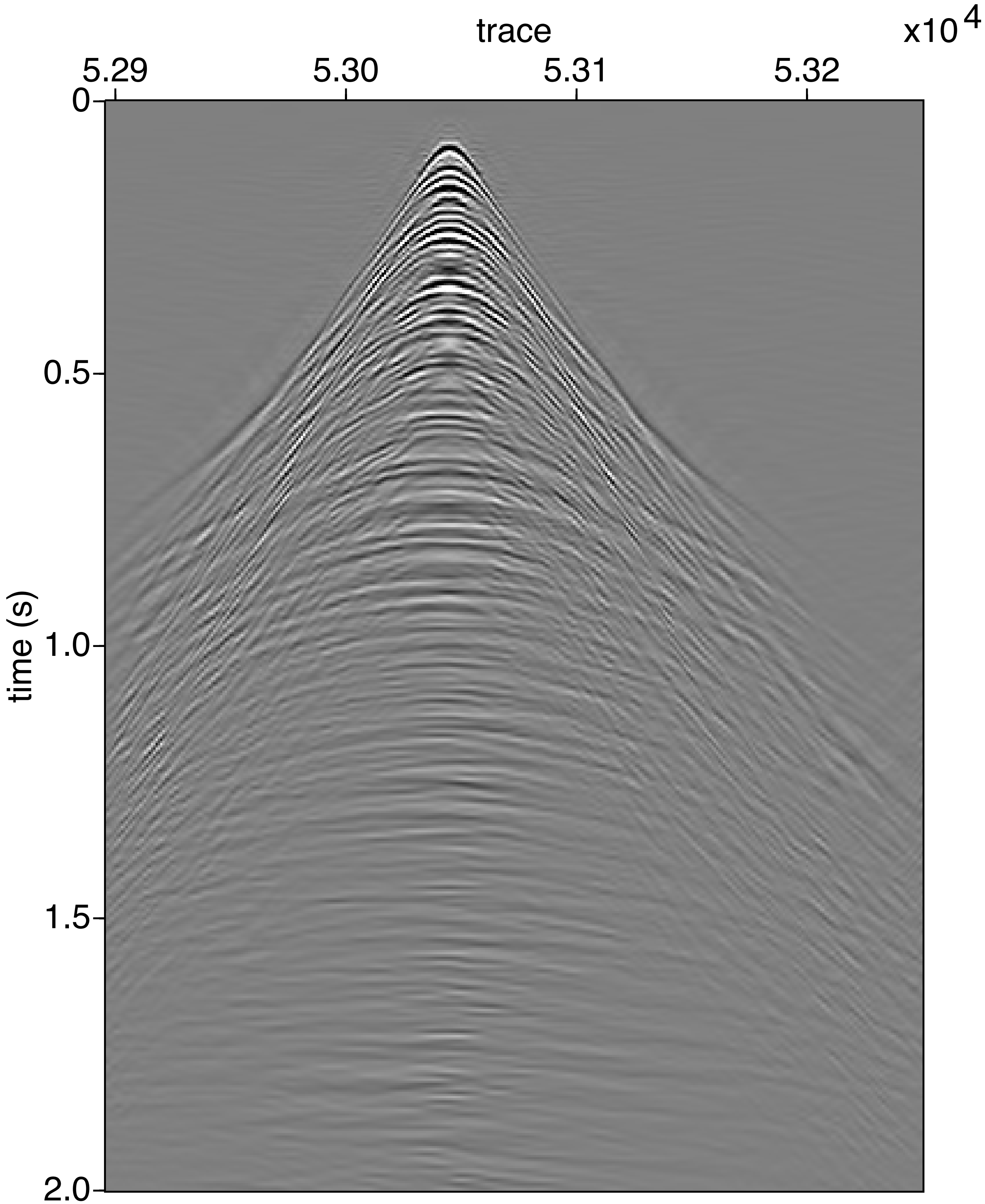
70% less



Gulf of Suez
Total Data

shot gather

$$n_r = 355$$
$$n_s = 355$$
$$n_t = 1024$$
$$dt = .004s$$



Gulf of Suez

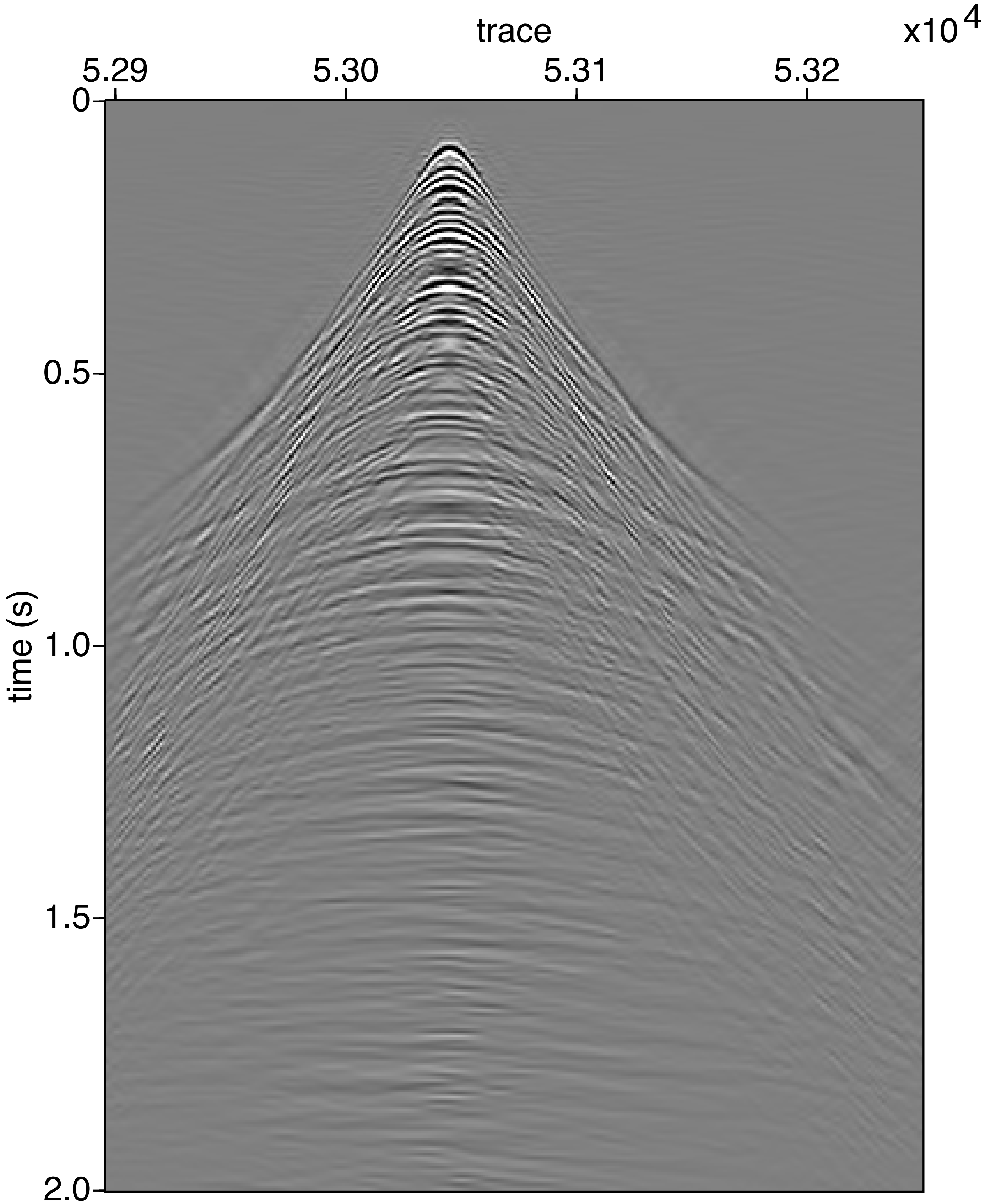
Full Data

Primary IR (G)

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

20% of rank budget

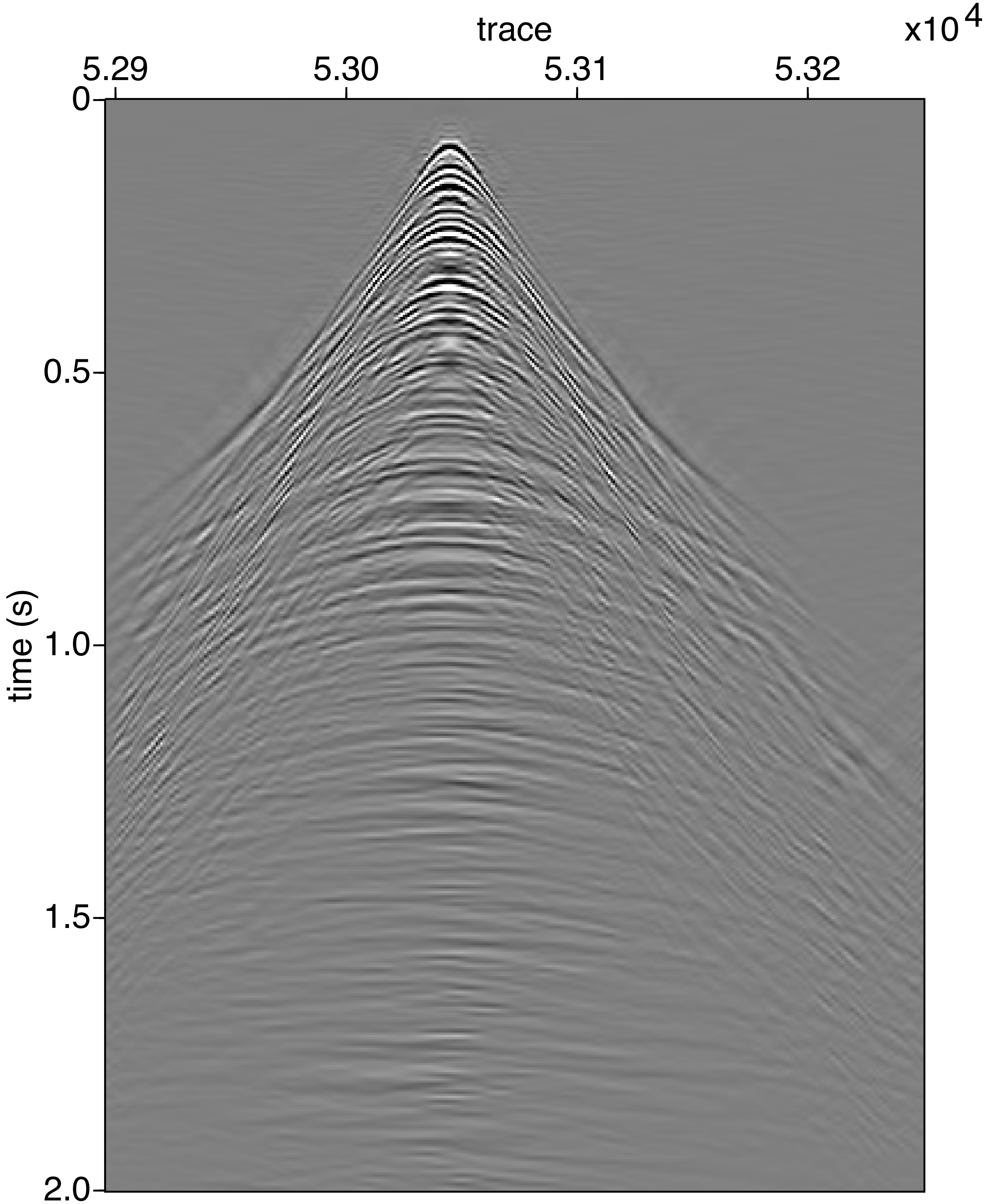
Primary IR (G)

SNR = 27dB

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

12% of rank budget

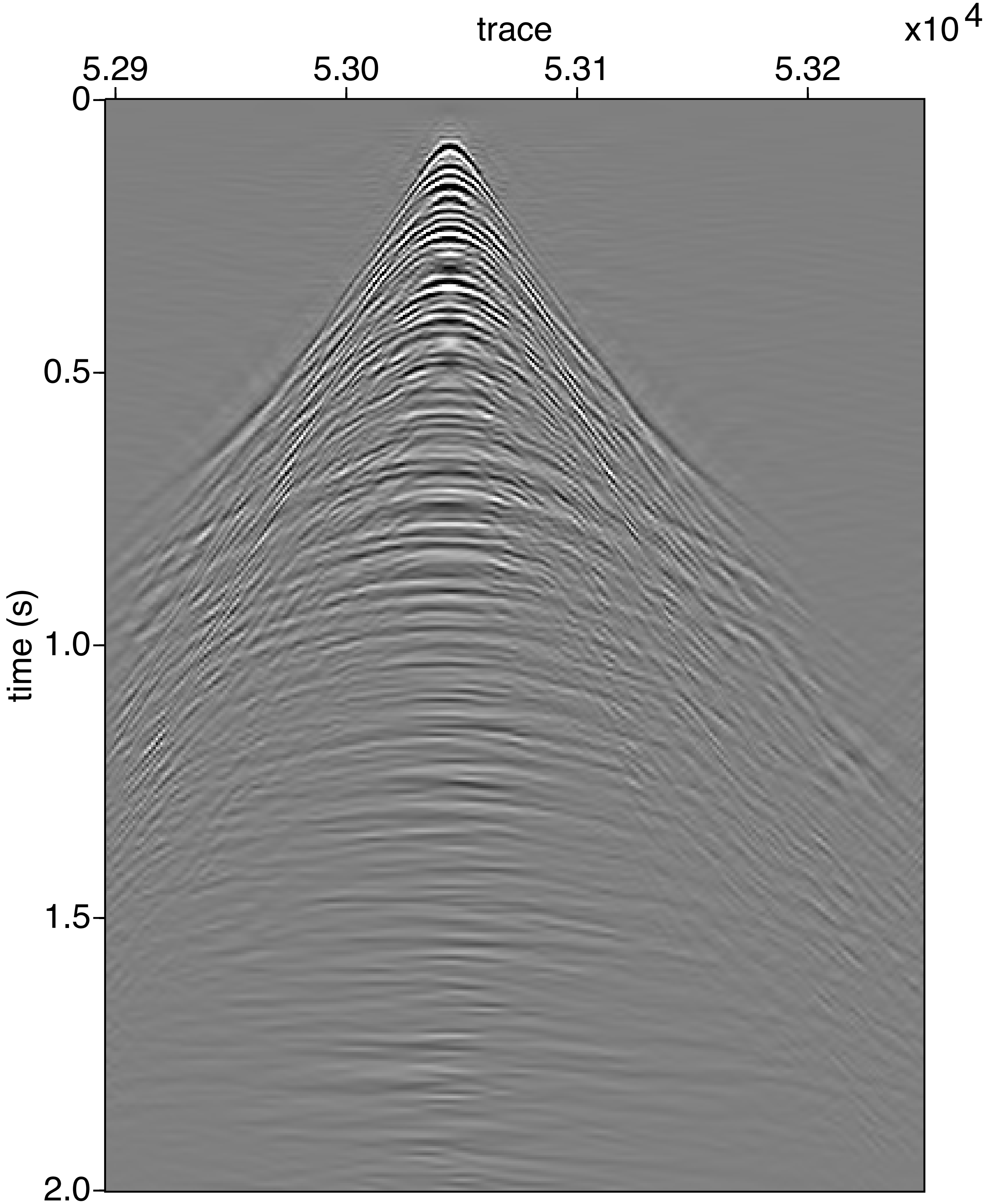
Primary IR (G)

SNR = 17dB

shot gather

2D Curvelet (Src-Rcv)

150 iterations



Gulf of Suez

8% of rank budget

Primary IR (G)

SNR = 12dB

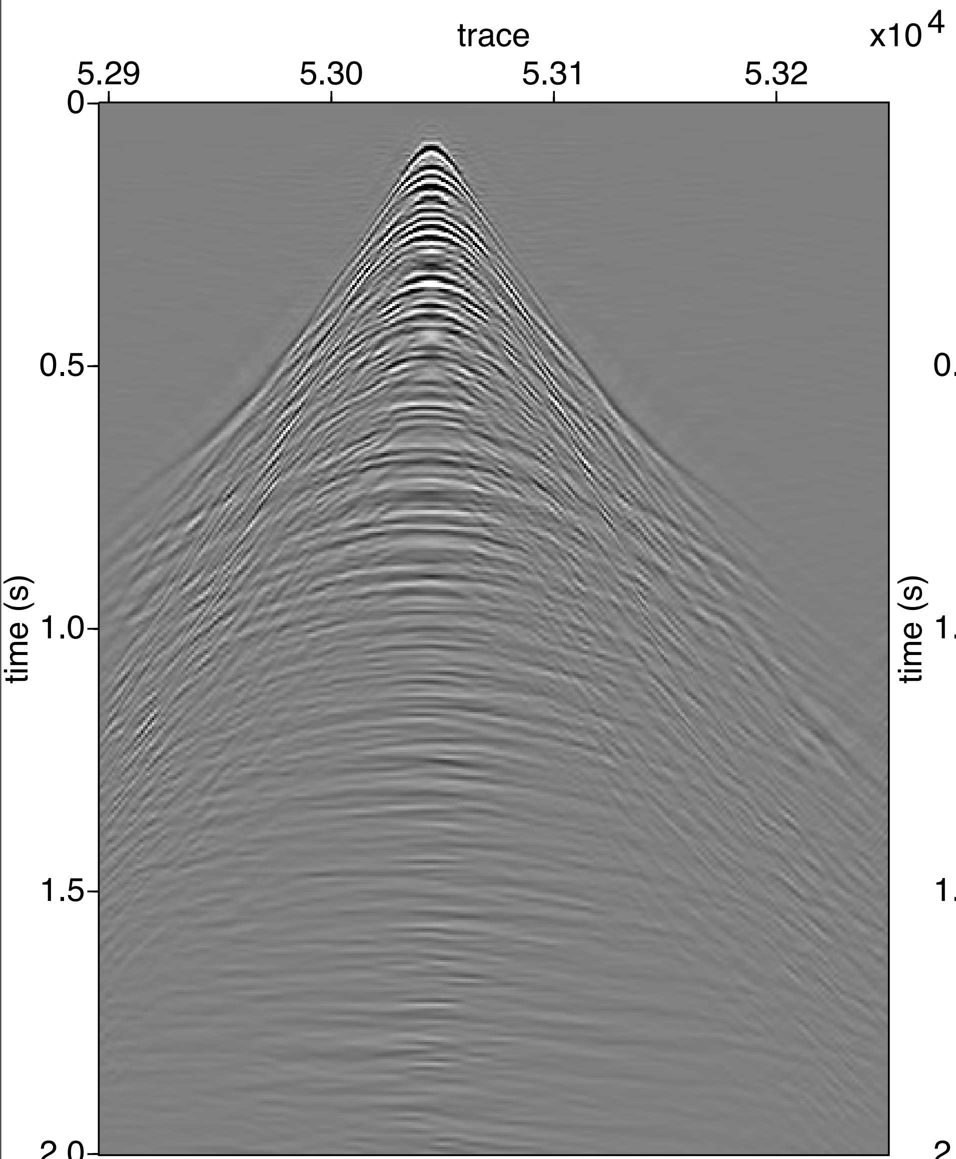
shot gather

2D Curvelet (Src-Rcv)

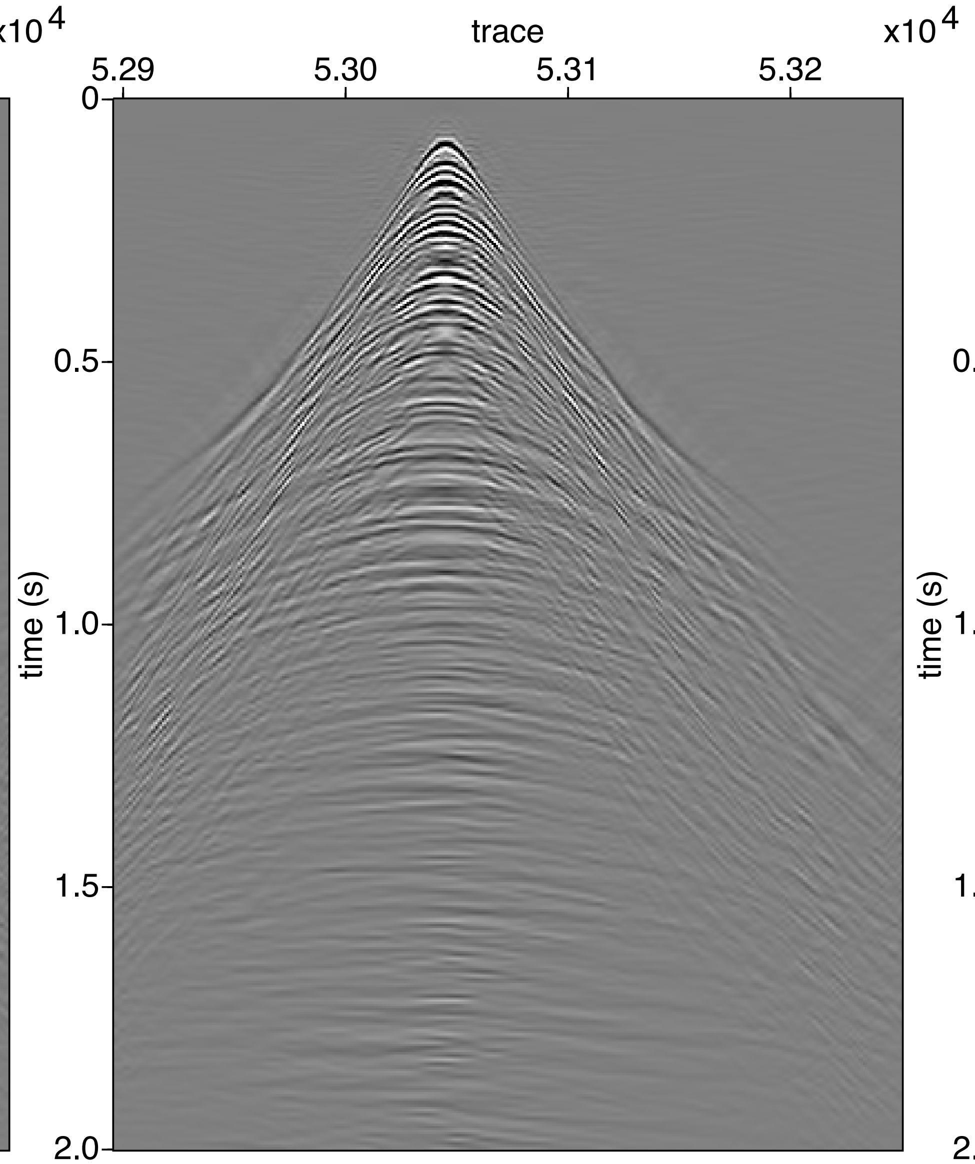
150 iterations

Difference in EPSI Result

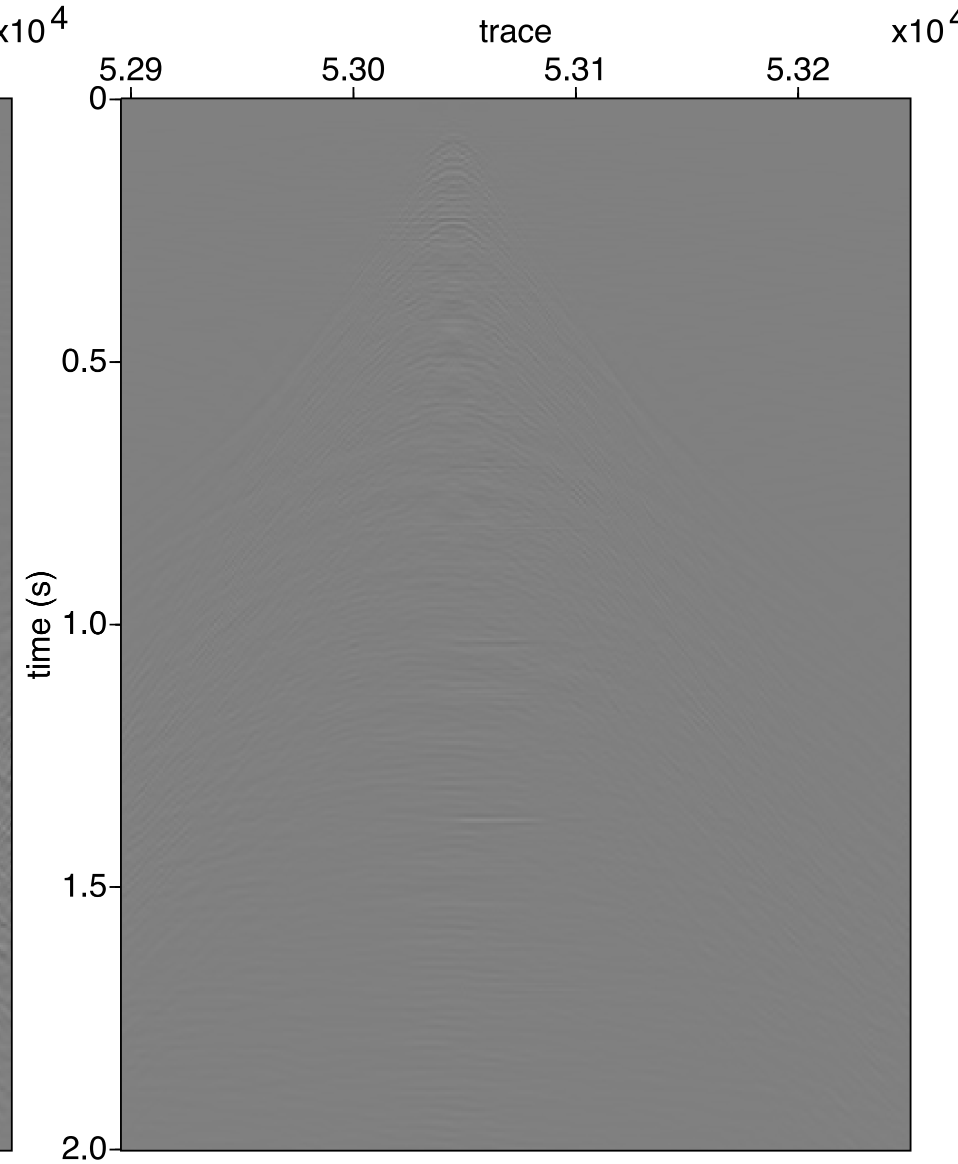
20% rank budget



Primary IR
full data



Primary IR
approximated Data



Difference

Key contributions

Robust implementation of estimation of primaries by sparse inversion

- leverages bi-convex optimization
- output deconvolved surface-free Green's function

Combination with *imaging & dimensionality reduction* look promising

Challenge: upscale to *full 3D*

What can SLIM do help bring FWI into production...

Make FWI more

- *efficient* via dimensionality-reduction
- *robust* via (student t) misfit penalty functionals
- *effective* via transform-domain sparsity promotion
- *versatile* via combination with
  - modeling of surface-related multiples
  - sophisticated *randomized* acquisition front end
  - elastic modeling & parameter identification

Short term
[FWI]

Incorporation of

- dimensionality reduction via *batching*
- robustness via *student t*

Deliverable: *scalable* framework with *manageable* memory *imprint*

Can be done with *existing* code *base*

- delivery more or less '*immediate*'

Mid term [+ imaging]

Regularization & conditioning via inclusion of

- transform-domain (joint) *sparsity* promotion
- approximations to the wave-equation *Hessian*

Deliverable: *effective* formulation that 'scales' to *high* frequencies

Requires

- exposure of Jacobians, their adjoints, and the GN Hessian
- multiple *preconditioned* GN iterations (parallellization)

Long term [+ multiples & acquisition]

Combinations with

- efficient *randomized* acquisition schemes
- (dimensionality-reduced) SRME operators

Deliverable: *parsimonious* formulation that leverages CS & SRME

Requires

- implementation & integration of SRME in FWI/imaging
- careful integration with *incomplete* acquisition including *new* sampling criteria

What’s needed

Research team

Sustained support for

- research faculty at competitive salaries
- faculty

Matched funding via

- NSERC collaborative R&D grants
- NSERC industrial chair grants
- Provincial grants + UBC support

Research team

BG commitment addresses current needs for warm bodies

- to supervise students
- to coordinate R & D
- to maintain & develop IT infrastructure

Situation without support is unsustainable

- SLIM team > 20 people = max handled by single faculty
- miss experienced long-term team members

IT infrastructure

- Sustained support for a "small" local compute solutions to prototype new 3D algorithms
- Access to a "large" compute solution to test developed algorithms on industry-size problems

Matched funding via

- NSERC collaborative R&D grants
- Provincial grants + UBC support

NO funding for HP-IT in Canada *without* industry *matching*!

SLIM

Organization

Matching will be organized in a “data mining satellite institute” at UBC involving faculty from CS, Math, EOS, etc.

“Brazilian parent institute” modeled after math institutes

- short/long-term thematic programs for visitors
- assistance with large-scale implementations
- help with industrialization where faculty act as consultants

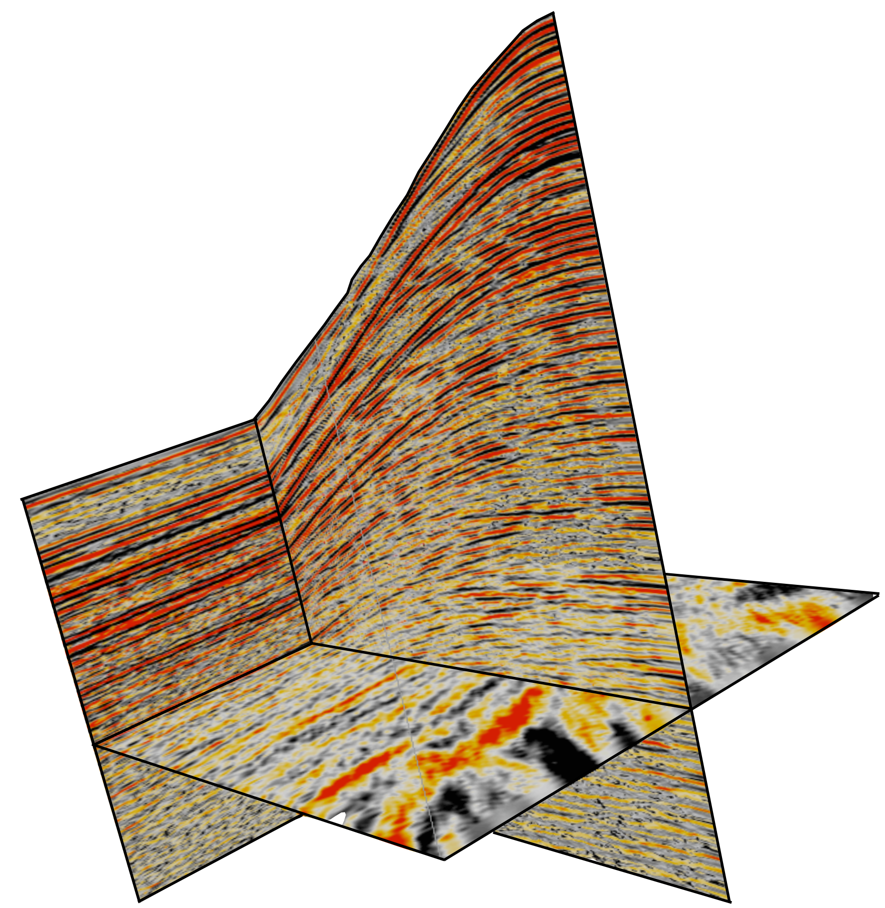
Guarantees influx of innovations in the area of “data mining”.

Requires a long-term sustainable approach, which differs from recent initiatives in Singapore, Kaust, and Brazil in the 80’s

Phases

1. Build parent and satellite institutes
   - access to IP from satellites
2. Have Brazil develop their own IP
   - generate ‘own’ capability
3. Help Brazil export IP
   - develop world-class services

HPC considerations



SLIM

Overview

Part 1

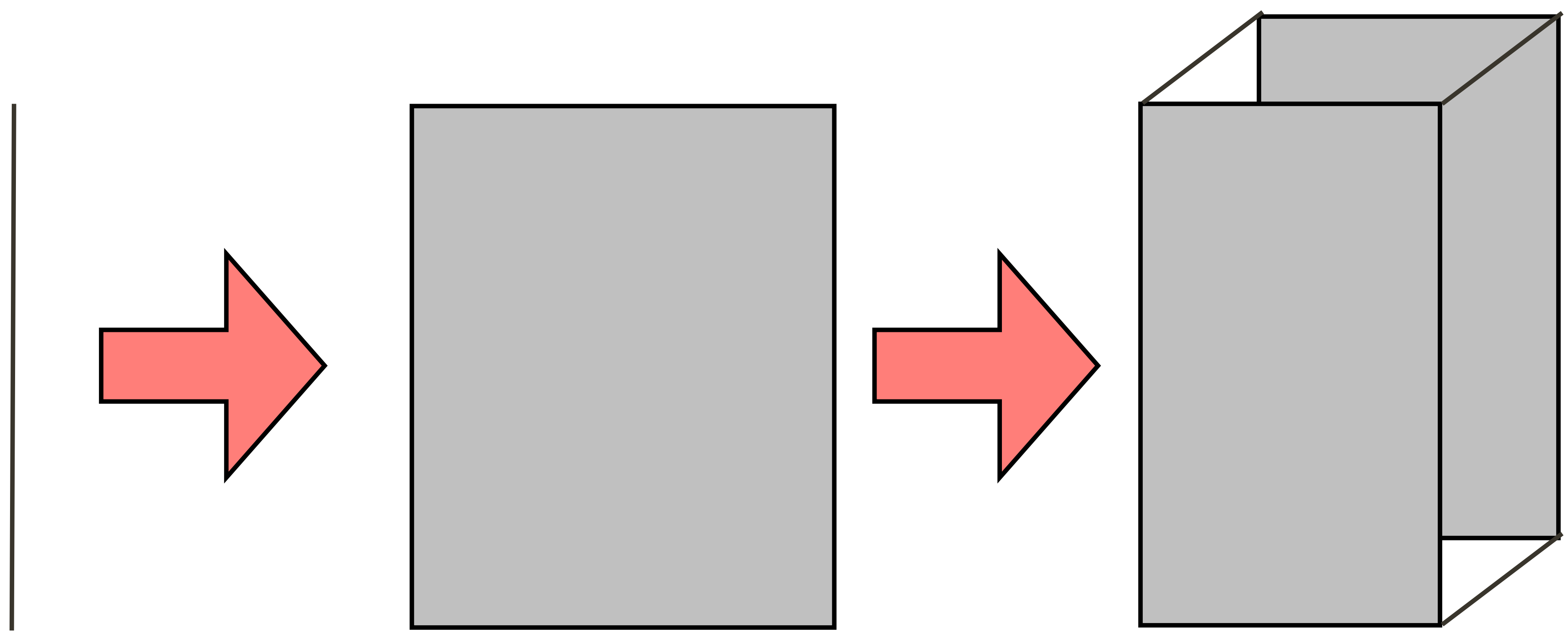
- Data organization
- Algorithm design
- example: Matlab & Javaseis

Part 2

- Fast FWI without source encoding
- FWI with robust misfit

Part 1

Data organization



trace-based:
standard
processing

shot-based:
RTM, FWI

cube-based:
FWI+

Trace-based

- operations that work on single trace
- `embarrassingly parallel’
- lot of disk i/o

shot-based

- RTM & FWI ...
- `embarrassingly parallel', given that one shot fits in the memory of a single node
- avoid communication & storage of data by reducing each shot to end result (misfit, gradient)

shot-based

misfit and gradient for FWI:

$$f = \sum_i f_i, \quad f_i = ||\mathbf{d}_i - P\mathbf{u}_i||_2^2$$

$$\mathbf{g} = \sum_i \mathbf{g}_i, \quad \mathbf{g}_i = \mathbf{u}_i \otimes \mathbf{v}_i$$

only communicate $\{f_i, \mathbf{g}_i\}$,
no need to store wavefields $\{\mathbf{u}_i, \mathbf{v}_i\}$

cube-based

- least-squares migration, FWI+
- needs `massively parallel' approach
- store *or* recompute wavefields

cube-based

LS-migration: $\min_{\mathbf{x}} \sum_i ||A_i\mathbf{x} - \mathbf{b}_i||_2^2$

$$\mathbf{b}_i = \mathbf{d}_i - P\mathbf{u}_i$$

at each iteration we need the residual for all i, and the action of A_i and A_i^*, all of which require the wavefields $\{\mathbf{u}_i, \mathbf{v}_i\}$

Algorithm design

- object-oriented programming
- algorithms work at high level and manipulate objects
- divide code into testable units

Algorithm design

- data cube and model `objects'

$$\mathcal{D} \quad \mathcal{M}$$

- modelling operator

$$\mathcal{D} = F(\mathcal{M})$$

- jacobian

$$\mathcal{D}' = J\mathcal{M}', \quad \mathcal{M}' = J^*\mathcal{D}'$$

Algorithm design

$$\color{red}{\mathcal{D} = F(\mathcal{M})}$$
$$\mathcal{R} = \mathcal{D} - \mathcal{D}_{\text{obs}}$$
$$f = \text{norm}(\mathcal{R})$$
$$\mathcal{G} = J^*(\mathcal{R})$$
$$\mathcal{M} = \mathcal{M} - \alpha\mathcal{G}$$

compute $\mathbf{u}_1$	$\mathbf{m}$ $\mathbf{d}_{1,\text{obs}}$
compute $\mathbf{u}_2$	$\mathbf{m}$ $\mathbf{d}_{2,\text{obs}}$
compute $\mathbf{u}_3$	$\mathbf{m}$ $\mathbf{d}_{3,\text{obs}}$
compute $\mathbf{u}_4$	$\mathbf{m}$ $\mathbf{d}_{4,\text{obs}}$

SLIM

Algorithm design

$$\mathcal{D} = F(\mathcal{M})$$
$$\mathcal{R} = \mathcal{D} - \mathcal{D}_{\text{obs}}$$
$$f = \text{norm}(\mathcal{R})$$
$$\mathcal{G} = J^*(\mathcal{R})$$
$$\mathcal{M} = \mathcal{M} - \alpha\mathcal{G}$$

$\mathbf{r}_1 = P\mathbf{u}_1 - \mathbf{d}_{1,\text{obs}}$	$\mathbf{u}_1\,\mathbf{m}$ $\mathbf{d}_{1,\text{obs}}$
$\mathbf{r}_2 = P\mathbf{u}_2 - \mathbf{d}_{2,\text{obs}}$	$\mathbf{u}_2\,\mathbf{m}$ $\mathbf{d}_{2,\text{obs}}$
$\mathbf{r}_3 = P\mathbf{u}_3 - \mathbf{d}_{3,\text{obs}}$	$\mathbf{u}_3\,\mathbf{m}$ $\mathbf{d}_{3,\text{obs}}$
$\mathbf{r}_4 = P\mathbf{u}_4 - \mathbf{d}_{4,\text{obs}}$	$\mathbf{u}_4\,\mathbf{m}$ $\mathbf{d}_{4,\text{obs}}$

Algorithm design

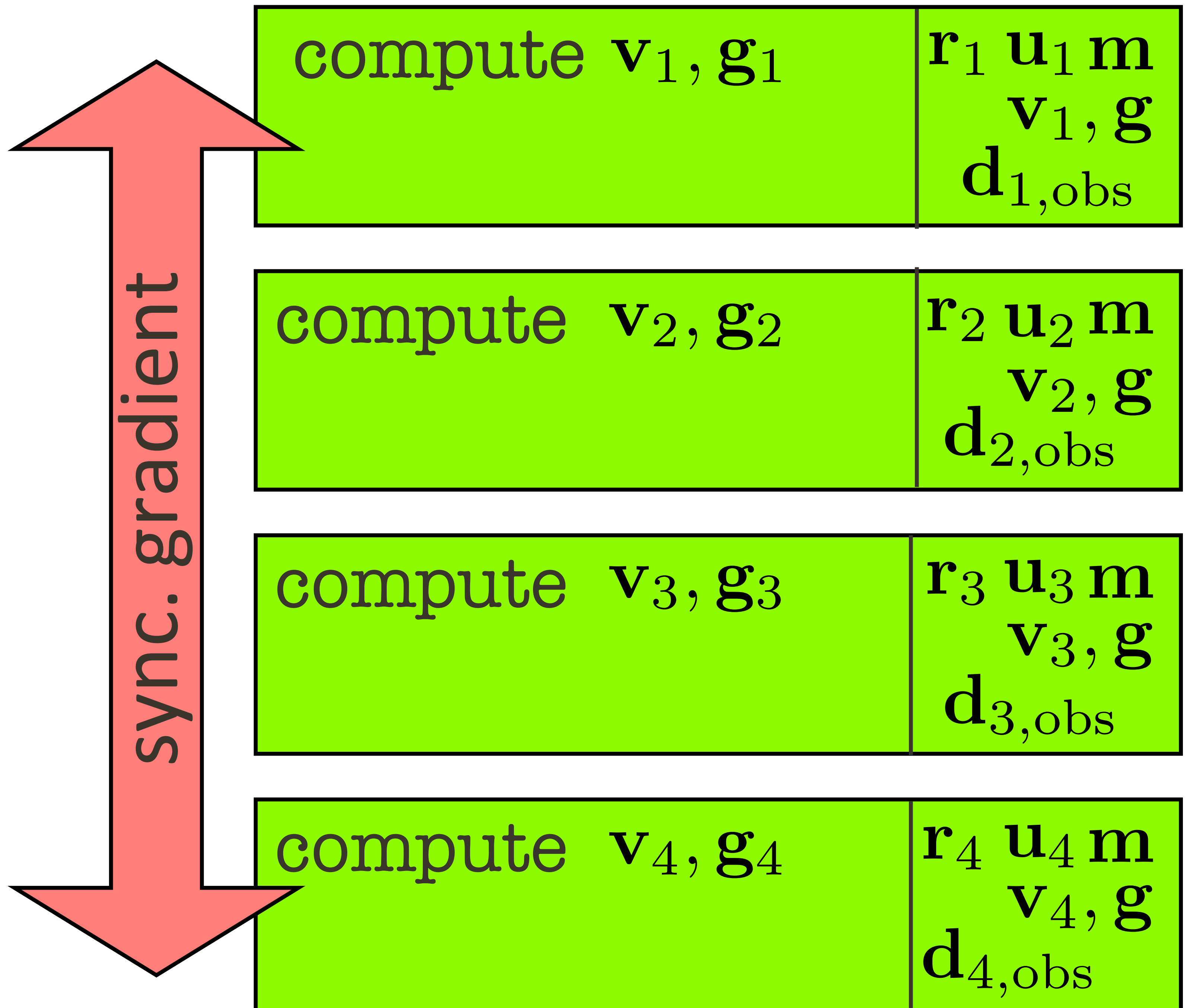
$$\mathcal{D} = F(\mathcal{M})$$
$$\mathcal{R} = \mathcal{D} - \mathcal{D}_{\text{obs}}$$
$$f = \text{norm}(\mathcal{R})$$
$$\mathcal{G} = J^*(\mathcal{R})$$
$$\mathcal{M} = \mathcal{M} - \alpha\mathcal{G}$$

$$f = \sum_i f_i$$

$f_1 = \|\|\mathbf{r}_1\|\|_2^2$	$\mathbf{r}_1\ \mathbf{u}_1\ \mathbf{m}$ $\mathbf{d}_{1,\text{obs}}$
$f_2 = \|\|\mathbf{r}_2\|\|_2^2$	$\mathbf{r}_2\ \mathbf{u}_2\ \mathbf{m}$ $\mathbf{d}_{2,\text{obs}}$
$f_3 = \|\|\mathbf{r}_3\|\|_2^2$	$\mathbf{r}_3\ \mathbf{u}_3\ \mathbf{m}$ $\mathbf{d}_{3,\text{obs}}$
$f_4 = \|\|\mathbf{r}_4\|\|_2^2$	$\mathbf{r}_4\ \mathbf{u}_4\ \mathbf{m}$ $\mathbf{d}_{4,\text{obs}}$

Algorithm design

$$\mathcal{D} = F(\mathcal{M})$$
$$\mathcal{R} = \mathcal{D} - \mathcal{D}_{\text{obs}}$$
$$f = \text{norm}(\mathcal{R})$$
$$\mathcal{G} = J^*(\mathcal{R})$$
$$\mathcal{M} = \mathcal{M} - \alpha\mathcal{G}$$



Algorithm design

$$\mathcal{D} = F(\mathcal{M})$$
$$\mathcal{R} = \mathcal{D} - \mathcal{D}_{\text{obs}}$$
$$f = \text{norm}(\mathcal{R})$$
$$\mathcal{G} = J^*(\mathcal{R})$$
$$\mathcal{M} = \mathcal{M} - \alpha\mathcal{G}$$

$\mathbf{m} = \mathbf{m} - \alpha\mathbf{g}$	$\mathbf{r}_1\ \mathbf{u}_1\ \mathbf{m}$ $\mathbf{v}_1, \mathbf{g}$ $\mathbf{d}_{1,\text{obs}}$
$\mathbf{m} = \mathbf{m} - \alpha\mathbf{g}$	$\mathbf{r}_2\ \mathbf{u}_2\ \mathbf{m}$ $\mathbf{v}_2, \mathbf{g}$ $\mathbf{d}_{2,\text{obs}}$
$\mathbf{m} = \mathbf{m} - \alpha\mathbf{g}$	$\mathbf{r}_3\ \mathbf{u}_3\ \mathbf{m}$ $\mathbf{v}_3, \mathbf{g}$ $\mathbf{d}_{3,\text{obs}}$
$\mathbf{m} = \mathbf{m} - \alpha\mathbf{g}$	$\mathbf{r}_4\ \mathbf{u}_4\ \mathbf{m}$ $\mathbf{v}_4, \mathbf{g}$ $\mathbf{d}_{4,\text{obs}}$

Matlab example

- use matlab as `scripting' language
- call external modelling code
- store wavefields on disk
- keep model and update in memory

Matlab example

```
% read data
Dobs = DataMap('datafile');

% set parameters
model = ...
Q = ...

% initial model
m0  = readfile('modelfile');

% FWI
fh = @(x) Jls(x,Q,model);

mn = minimize(fh,m0);
```

Matlab example

```
[D] = function F(m,Q,model)
...
...
D = DataMap(nrec,nsrc,nfreq)
for i=1:nfreq
    for j=1:nsrc
        u = simulate(m,Q(:,i),...);
        D(:,j,i) = P*u;
    end
end
```

Matlab example

```
[f,g] = function Jls(m,Q,D,model)

% modeling
[Dt,Jt] = F(m,Q,model);
% residual
R   = Dt - D;
% misfit
f   = norm(R)^2;
% gradient
g   = Jt*R;
```

Matlab example

```
function [m] = minimize(fh,m0)
mk = m0;

for k = 1:maxiter
  [fk,gk] = fh(mk);
  l = 1;
  fn = fh(mk - l*gk);
  while fh(mk-l*gk) > fk + l*norm(gk);
    l = l/2;
  end
  mk = mk - l*g;
end
```

Matlab example

```
class DataMap

properties
  size
  filename

methods
  function M = DataMap(filename,size)
  % create empty datamap

  function M = plus(A,B)
  % create new datamap A+B

  function a   =  norm(A)
  % calculate norm of datamap
```

JavaSeis

- use existing capability for handling seismic data in distributed (memory and disk)environment
- optimized operations such as transpose and FFT’s
- can be bridged to matlab

Conclusions

- Use matlab as `scripting language'
- allows us to quickly propotype and benefit from algorithms developed by `experts'
-

SLIM

Conclusions

- Overloading allows us to call external modelling code, and access data from external sources (disk, memory)
- no need to explicitly import data into matlab
-