Time-domain least-squares RTM with sparsity promotion on field data

Philipp A. Witte and Felix J. Herrmann

SLIM
University of British Columbia

Motivation

Our first steps with sparsity promoting LSRTM in the time-domain:

- develop robust workflow with little user interaction
- experiments with increasing difficulty

Synthetic data set: linearized data, (non-) inversion crime

↓

Synthetic data set: non-linearized data

↓

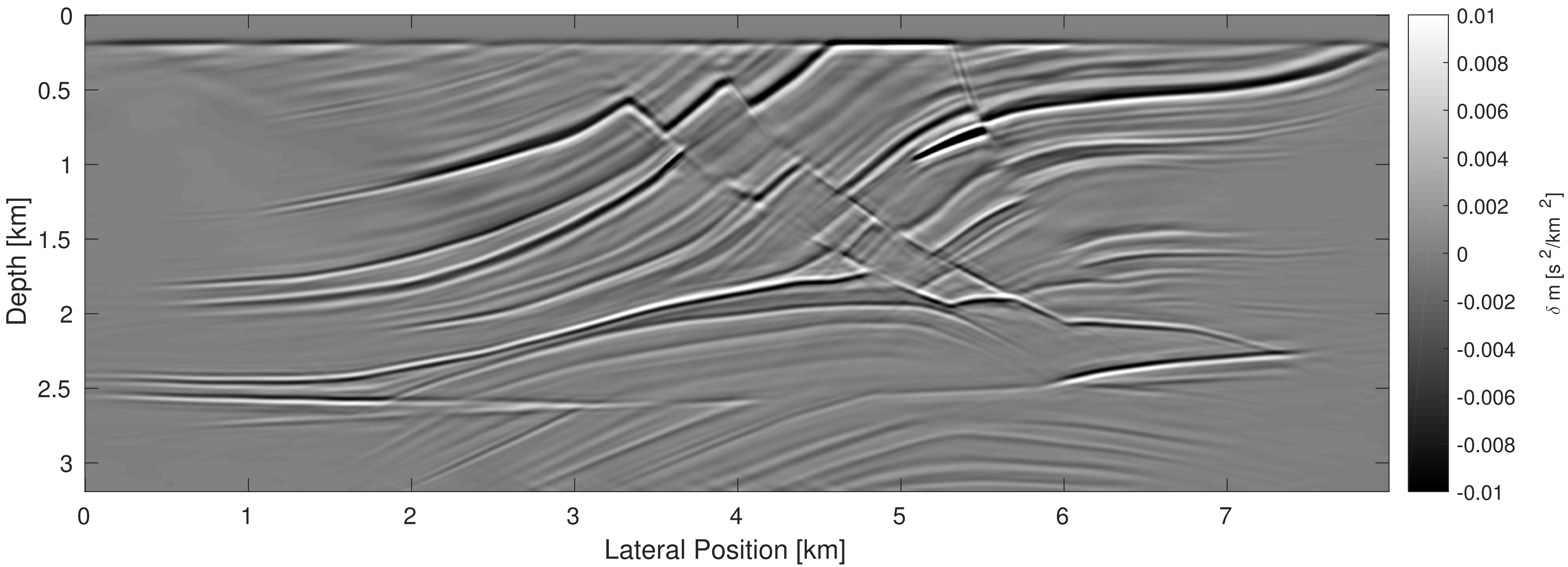
Field data set: Machar 2D

Motivation

Solve sparsity promoting LSRTM w/ linearized Bregman

$$\text{minimize} \quad \lambda||\mathbf{C}\delta\mathbf{m}||_1 + \frac{1}{2}||\mathbf{C}\delta\mathbf{m}||_2^2$$

$$\text{subject to } ||\, \mathbf{J}\delta\mathbf{m} - \delta\mathbf{d}||_2 \leq \sigma$$

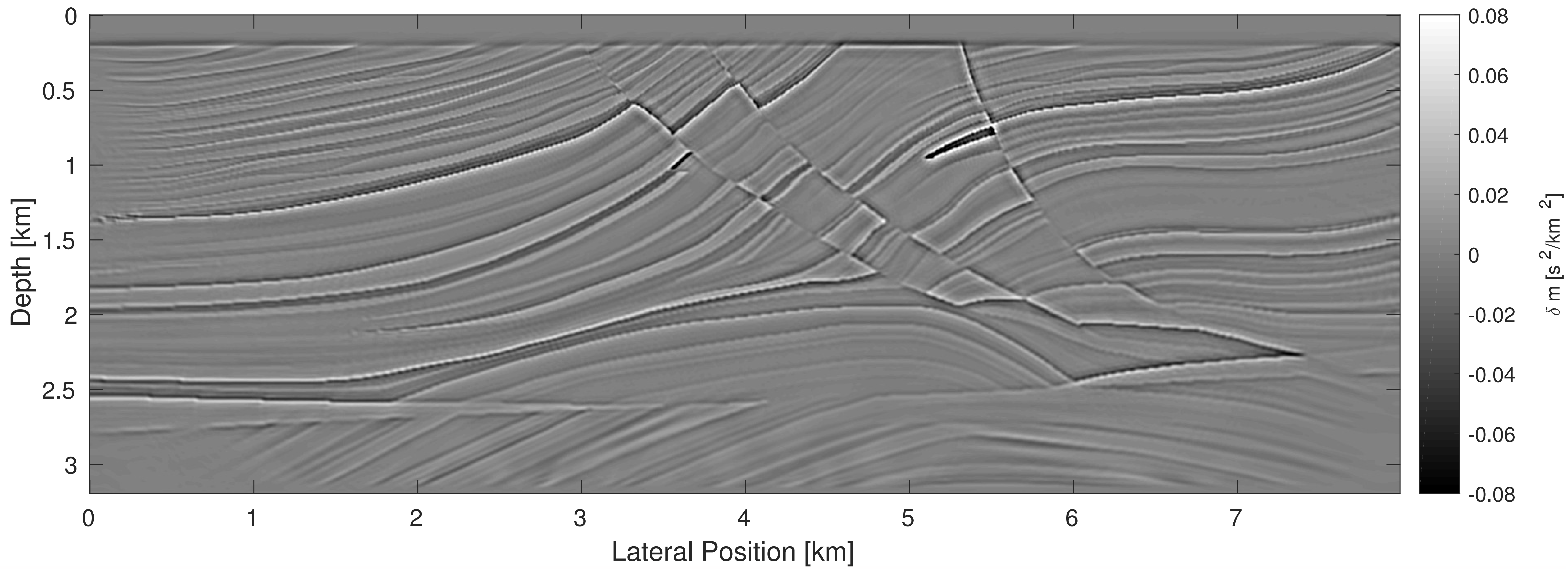


Basic implementation of SP-LSRTM

Motivation

Solve sparsity promoting LSRTM w/ linearized Bregman

$$\begin{aligned}&\text{minimize} \quad \lambda||\mathbf{C}\delta\mathbf{m}||_1 + \frac{1}{2}||\mathbf{C}\delta\mathbf{m}||_2^2 \\ &\text{subject to } ||\,\mathbf{J}\delta\mathbf{m} - \delta\mathbf{d}||_2 \leq \sigma\end{aligned}$$



Pre-conditioned SP-LSRTM w/ λ auto-tuning

Sparsity promoting LSRTM

Problem formulation:

$$\begin{aligned}&\text{minimize} \quad \lambda||\mathbf{C}\delta\mathbf{m}||_1 + \frac{1}{2}||\mathbf{C}\delta\mathbf{m}||_2^2 \\ &\text{subject to } ||\,\mathbf{J}\delta\mathbf{m} - \delta\mathbf{d}||_2 \leq \sigma\end{aligned}$$

$\delta\mathbf{m}$: model perturbation/image

$\delta\mathbf{d}$: linearized data (single scattered data)

$\mathbf{J}$: linearized forward modeling operator (Jacobian)

$\mathbf{C}$: curvelet transform

Sparsity promoting LSRTM

Problem formulation:

$$\text{minimize} \quad \lambda||\mathbf{C}\delta\mathbf{m}||_1 + \frac{1}{2}||\mathbf{C}\delta\mathbf{m}||_2^2$$
$$\text{subject to } ||\ \mathbf{J}\delta\mathbf{m} - \delta\mathbf{d}||_2 \leq \sigma$$

Left- and right-hand preconditioning:

$$\delta\mathbf{m} = \mathbf{M}_R^{-1}\mathbf{x}$$
$$\mathbf{M}_L^{-1}\mathbf{J}\mathbf{M}_R^{-1}\mathbf{x} = \mathbf{M}_L^{-1}\delta\mathbf{d}$$

SLIM

Preconditioning

Left-hand preconditioning (data space)

$$\mathbf{M}_L^{-1} = \mathbf{T}_d \mathbf{F}$$

$\mathbf{T}_d$: Topmute

$\mathbf{F}$: Fractional integration $\partial_{|t|}^{-1/2}$

Right-hand preconditioning (model space)

$$\mathbf{M}_R^{-1} = \mathbf{T}_m \mathbf{A}$$

$\mathbf{T}_m$: Topmute

$\mathbf{A}$: Depth scaling

Preconditioned SP-LSRTM

$$\begin{aligned} &\text{minimize} \quad \lambda||\mathbf{Cx}||_1 + \frac{1}{2}||\mathbf{Cx}||_2^2 \\ &\text{subject to } || \underbrace{\mathbf{M}_L^{-1}\mathbf{J}\mathbf{M}_R^{-1}}_{\hat{\mathbf{J}}}\mathbf{x} - \underbrace{\mathbf{M}_L^{-1}\delta\mathbf{d}}_{\mathbf{b}}||_2 \leq \sigma \end{aligned}$$

Algorithm:

$$\begin{array}{ll} 1. & \textbf{for } k = 0, 1, \cdots \\ 2. & \quad \mathbf{z}_{k+1} = \mathbf{z}_k - t_k \hat{\mathbf{J}}^*_{r(k)} (\hat{\mathbf{J}}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)}) \cdot \max(0, 1 - \frac{\sigma}{||\hat{\mathbf{J}}_{r(k)}\mathbf{x}_k - \mathbf{b}_{r(k)}||_2}) \\ 3. & \quad \mathbf{x}_{k+1} = \mathbf{C}^* S_\lambda(\mathbf{C}\mathbf{z}_{k+1}) \\ 4. & \textbf{end for} \end{array}$$

SP-LSRTM examples w/ linearized data

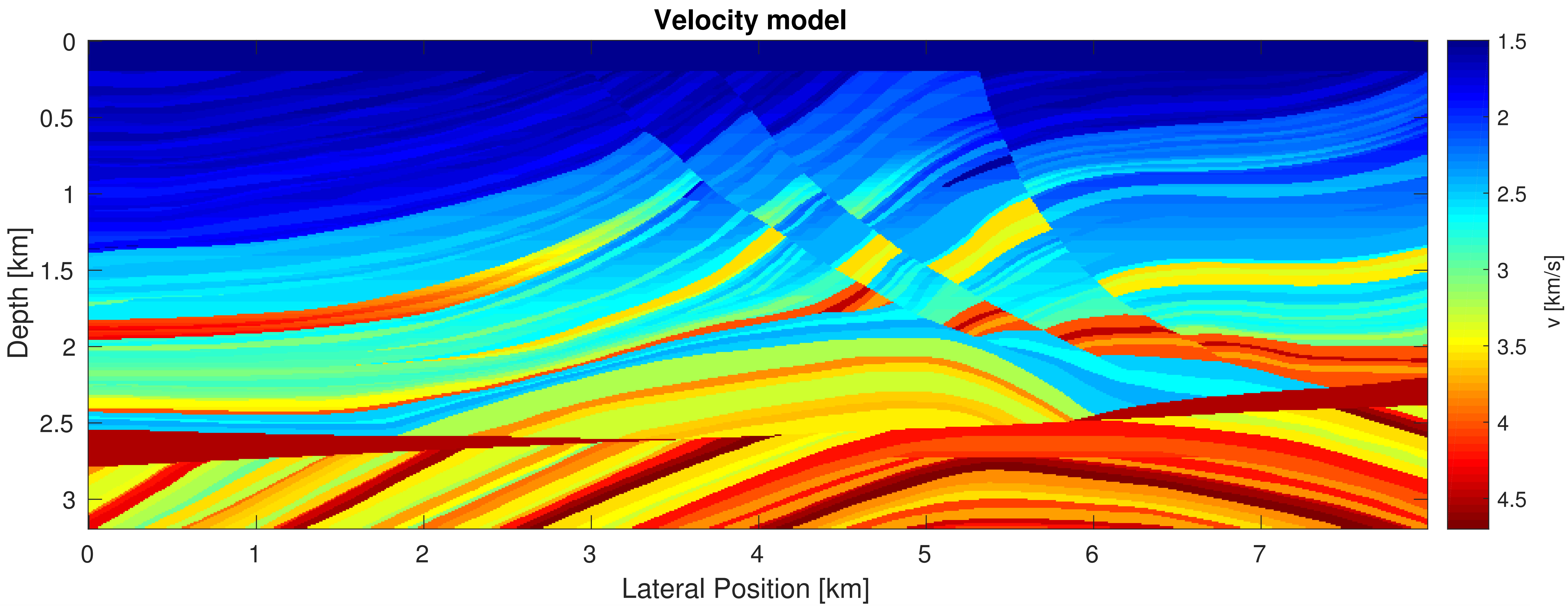
Marmousi model

- 320 shots, 4 seconds recording time
- 30 Hz Ricker wavelet
- 25 m source spacing
- OBNs with 10 m receiver spacing

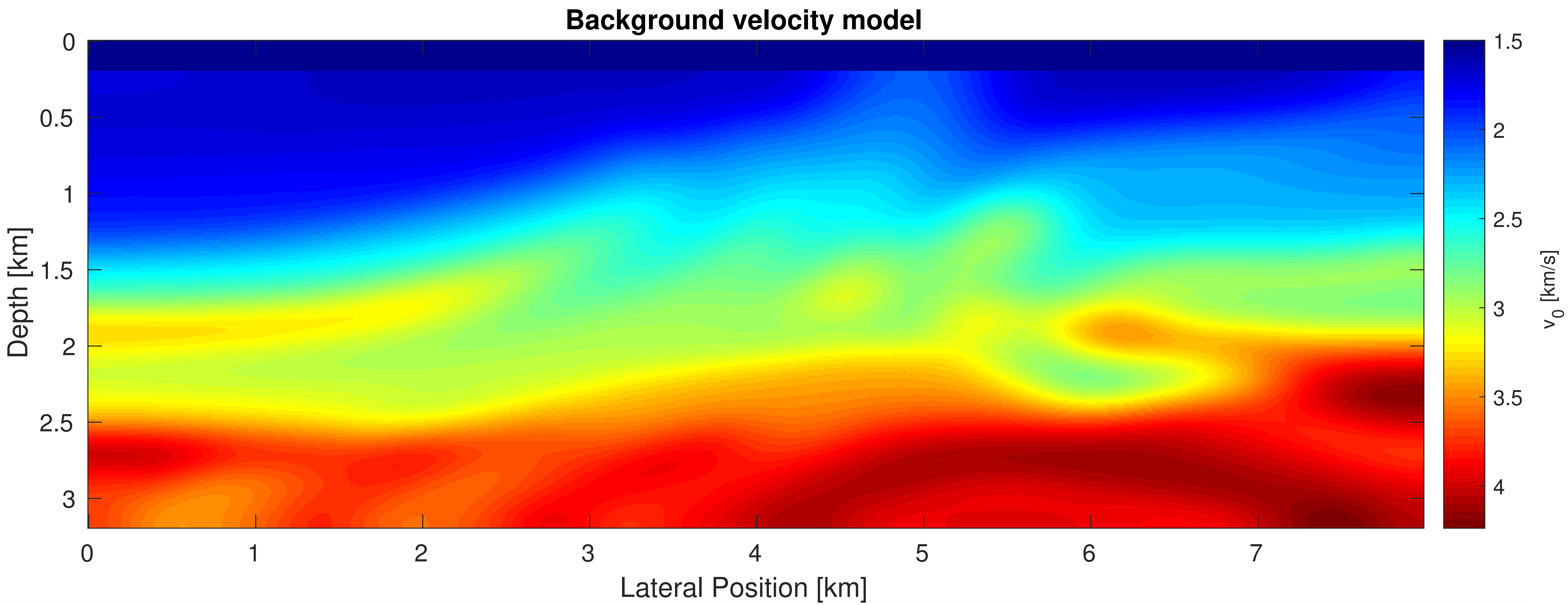
Inversion parameters

- 40 iterations
- 8 shots per iteration (1 data pass)
- linearized observed data (inversion crime)
- no data preprocessing

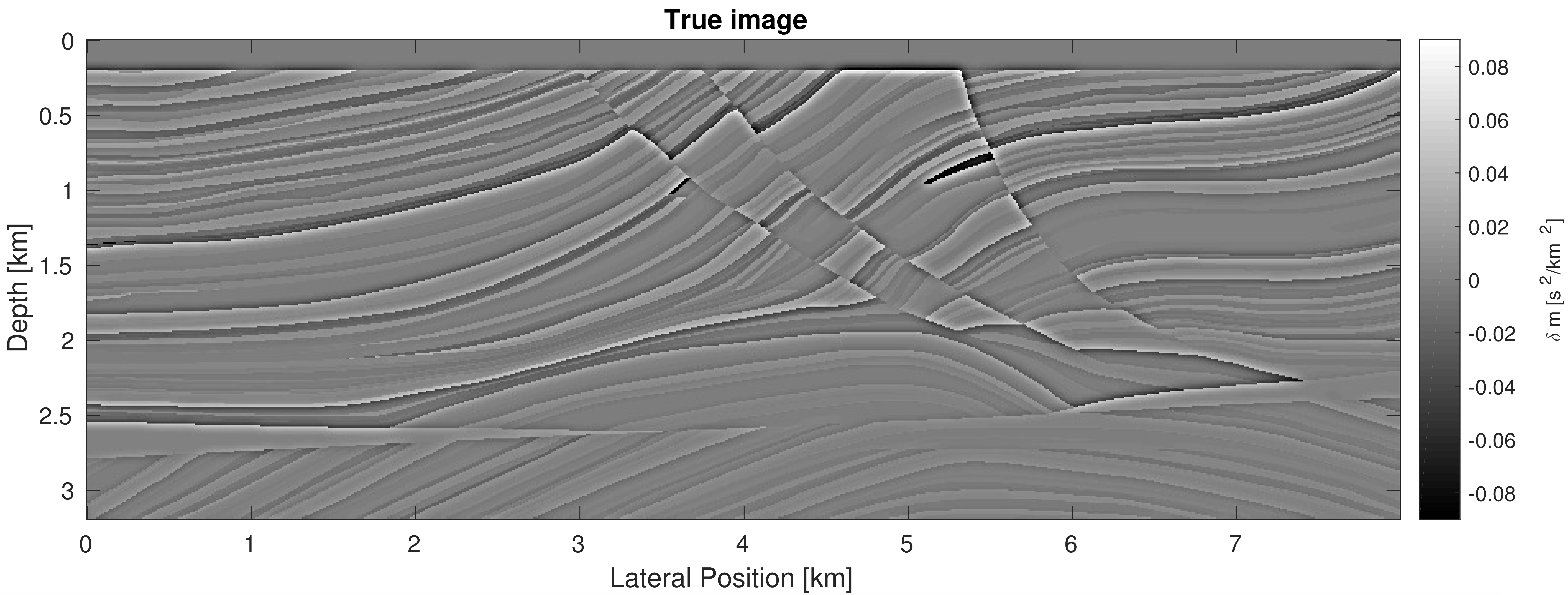
SP-LSRTM: Marmousi



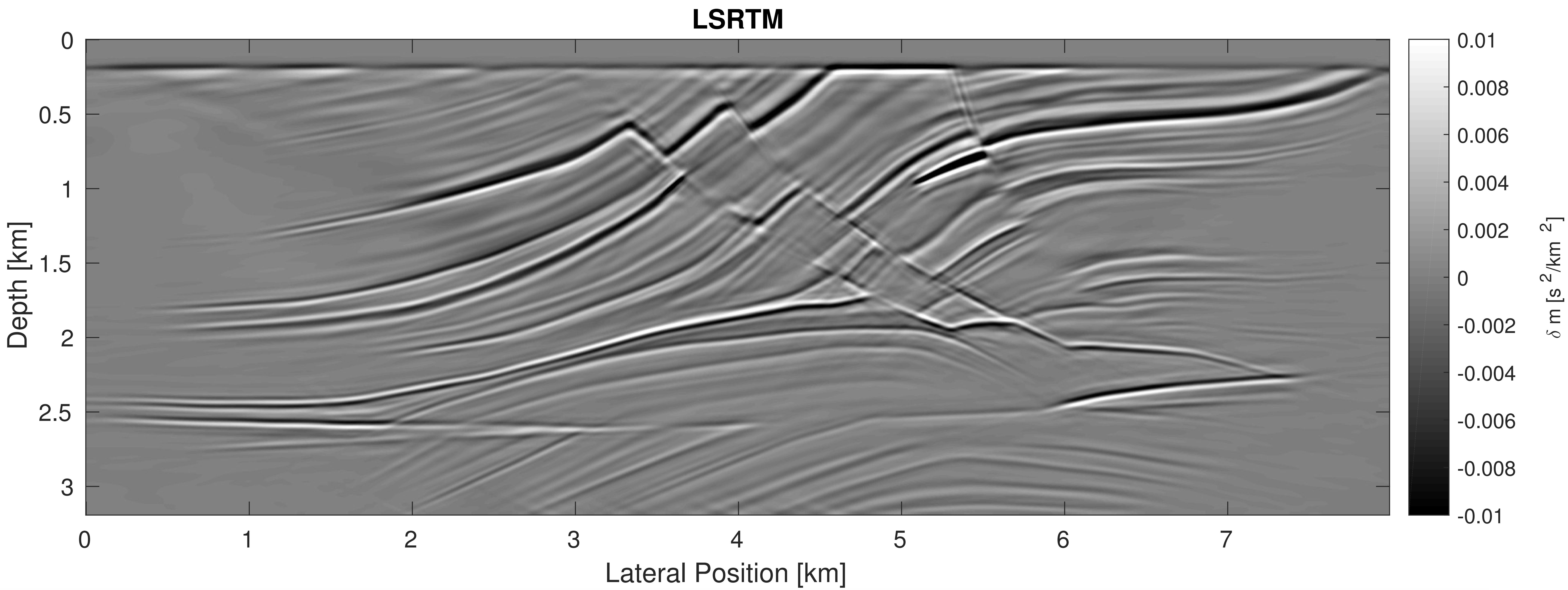
SP-LSRTM: Marmousi



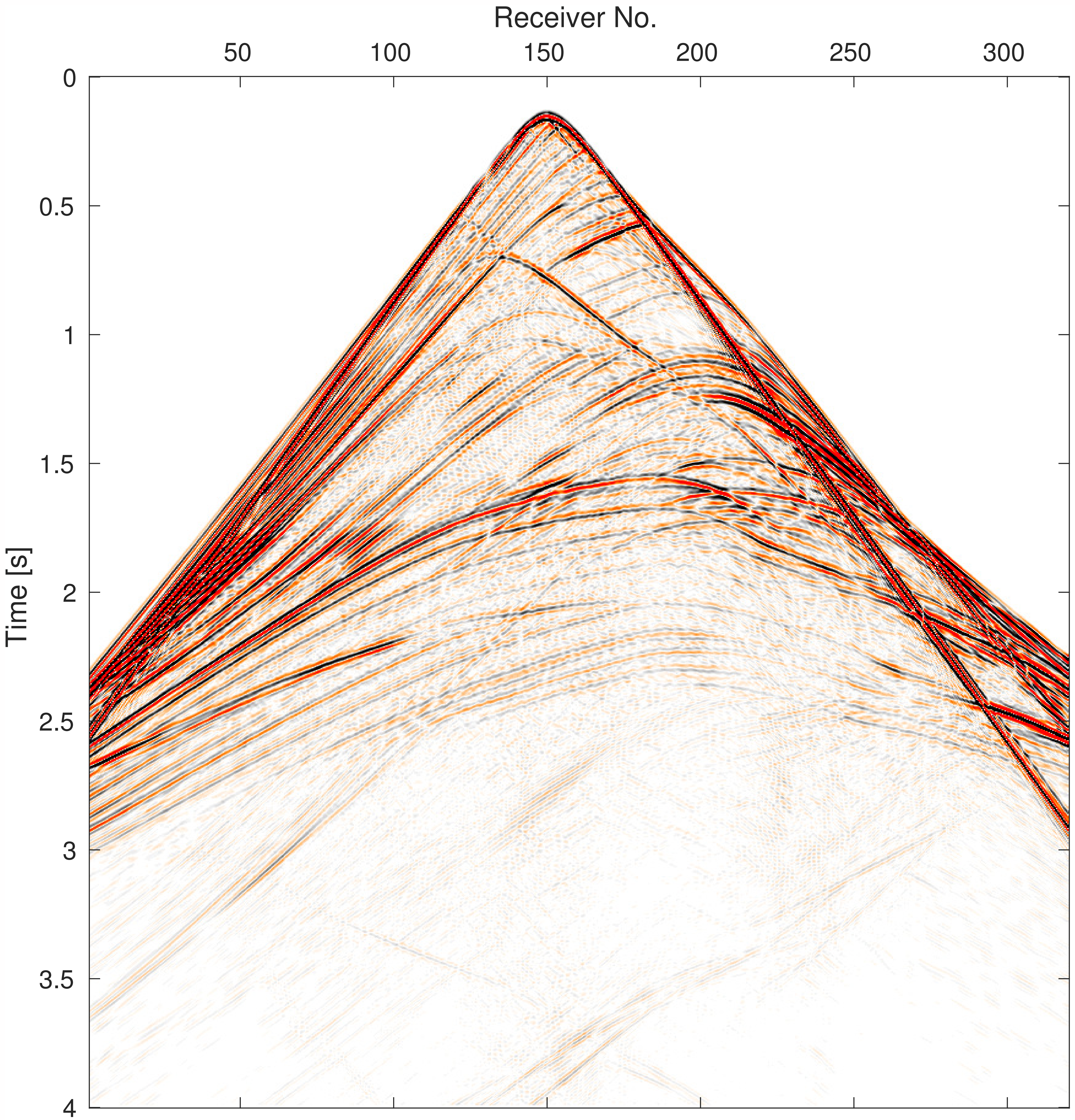
SP-LSRTM: Marmousi



SP-LSRTM: Marmousi



SP-LSRTM: Marmousi



Linearized data

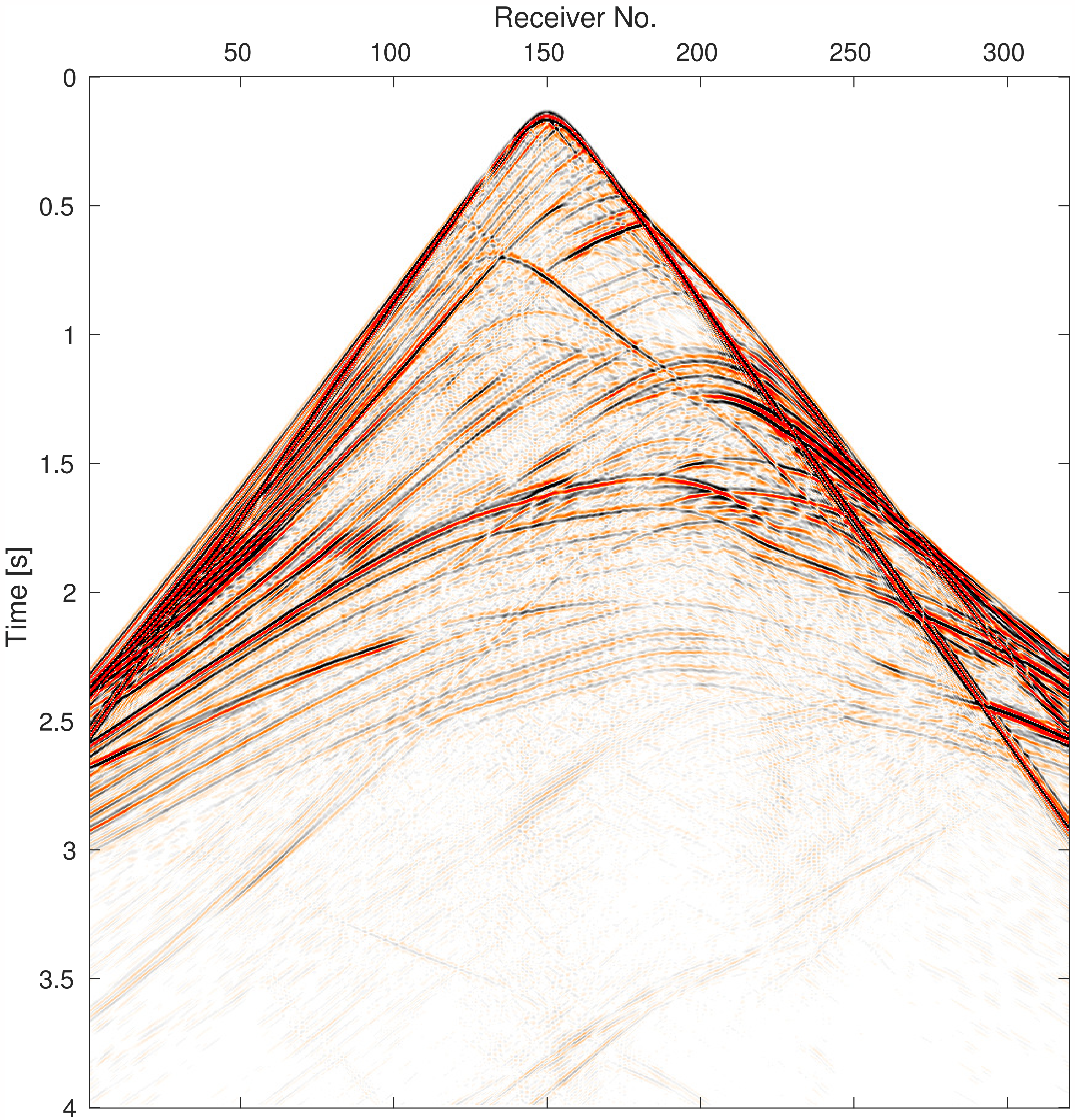
Problem:

- ocean bottom reflection
- backscattered energy in background model ⟶ low frequency updates

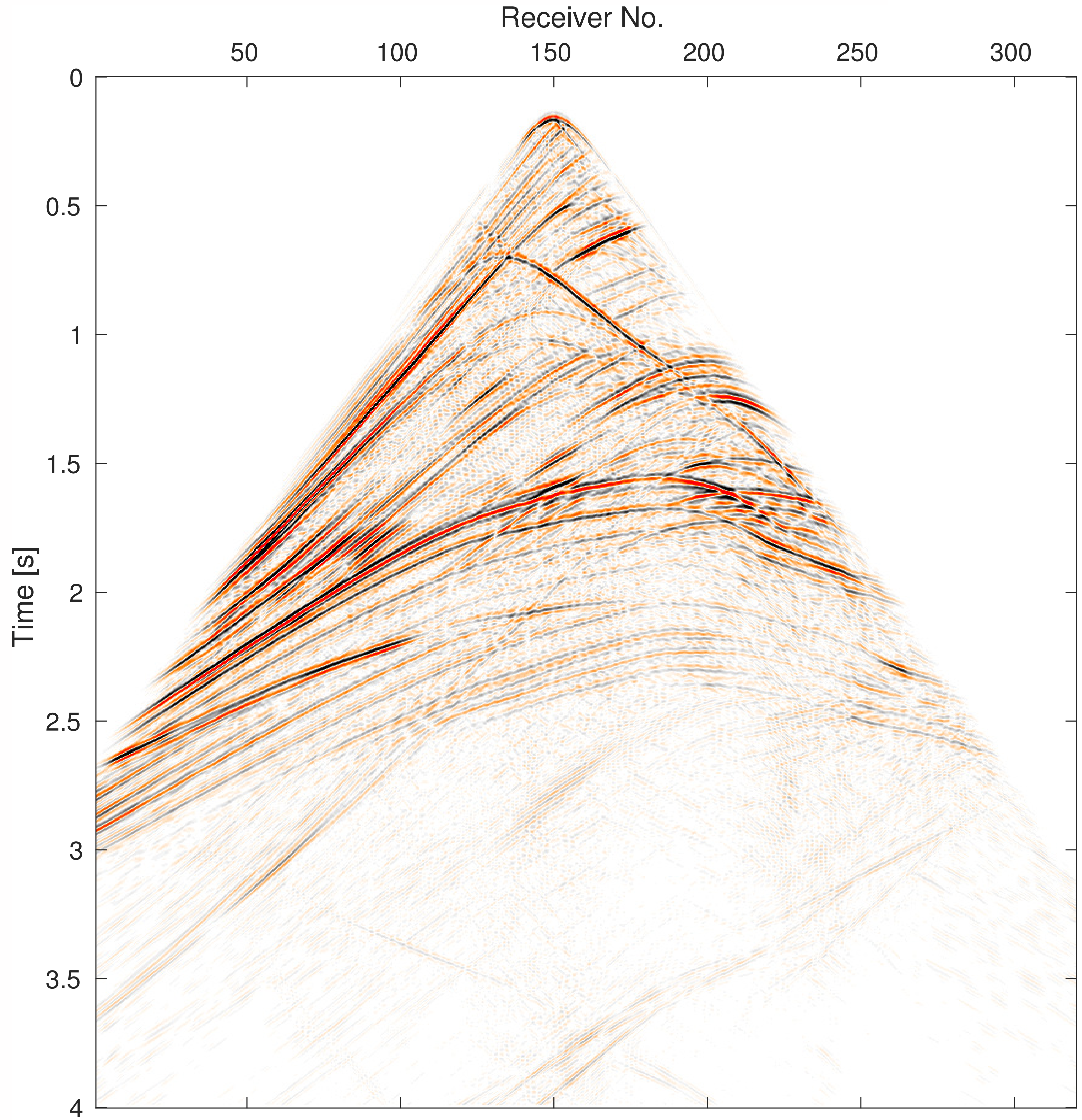
Data-topmute

- mute data outside window of OB reflection
- apply mute at each iteration to observed + synthetic data

SP-LSRTM: Marmousi

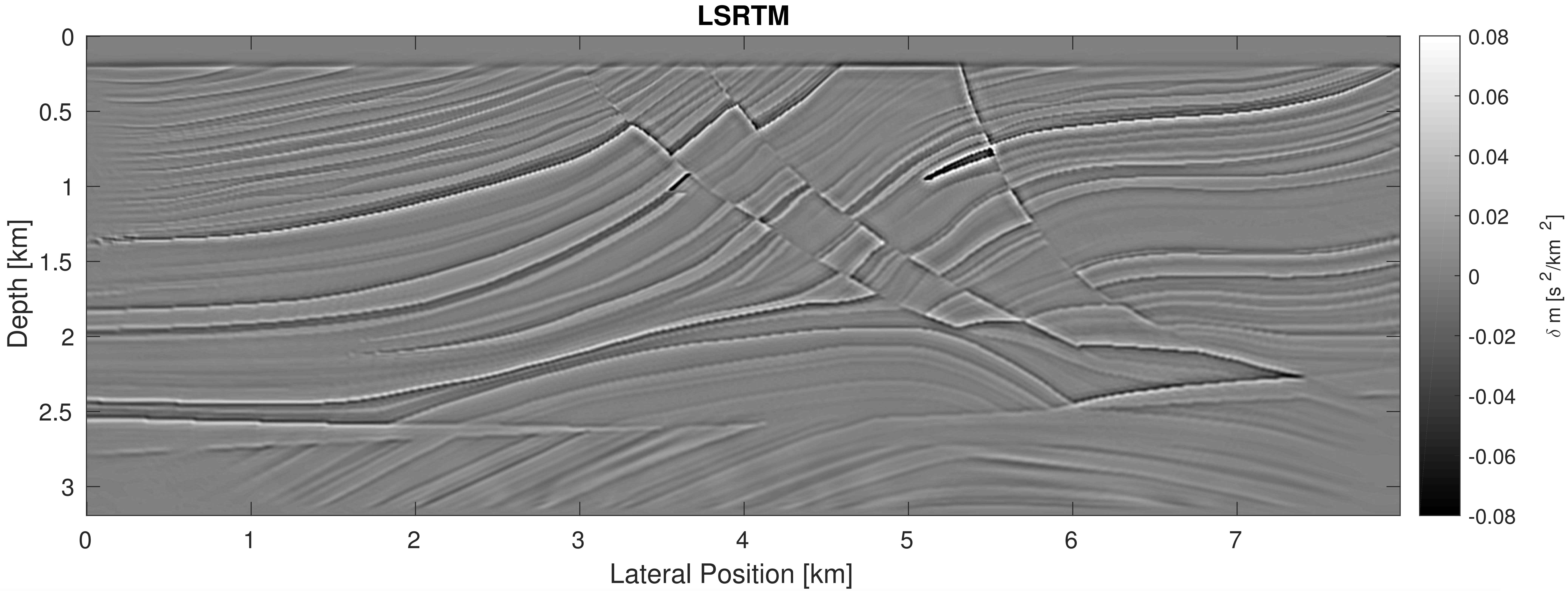


Linearized data



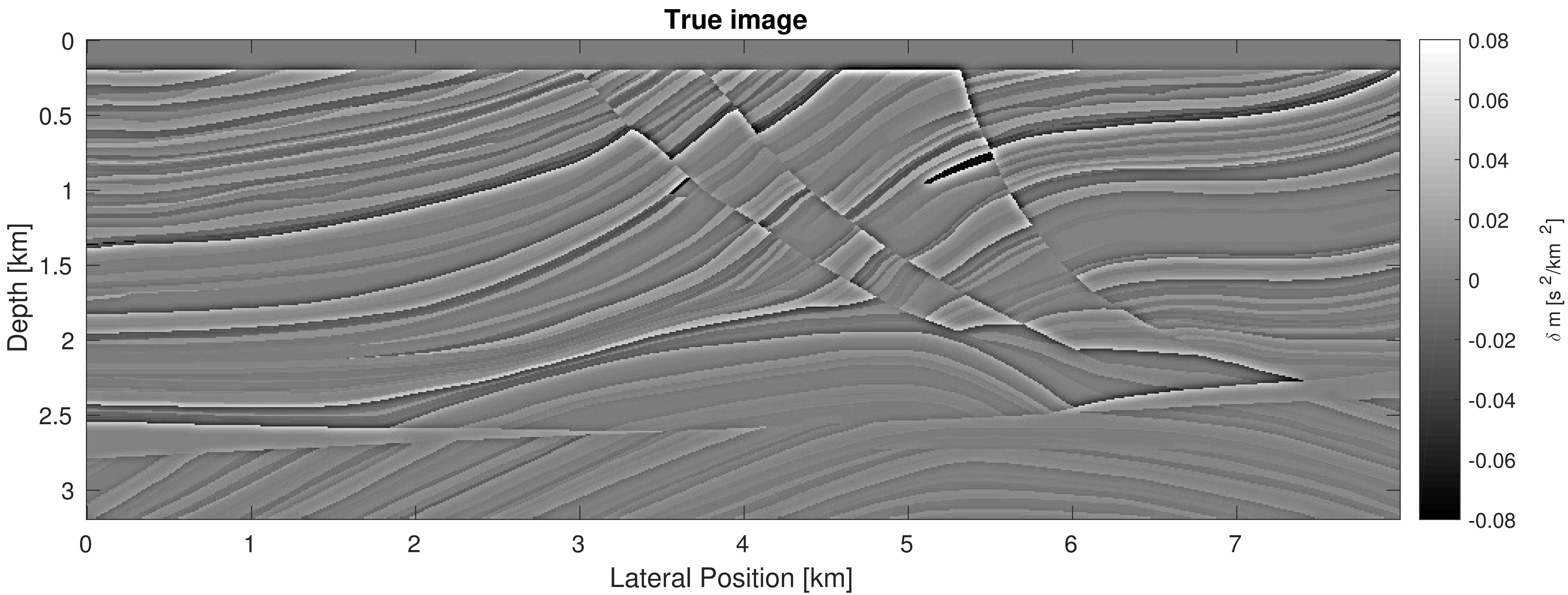
Muted linearized data

SP-LSRTM: Marmousi

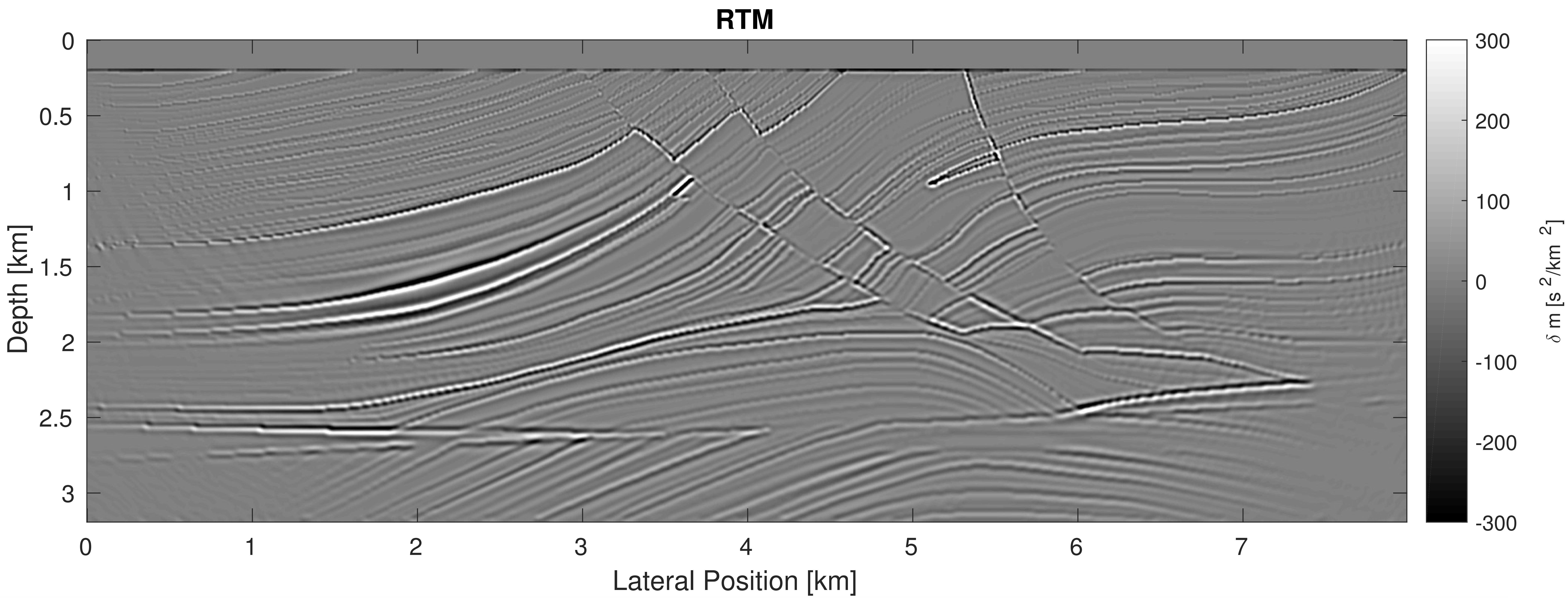


Result after 40 iterations with data topmute

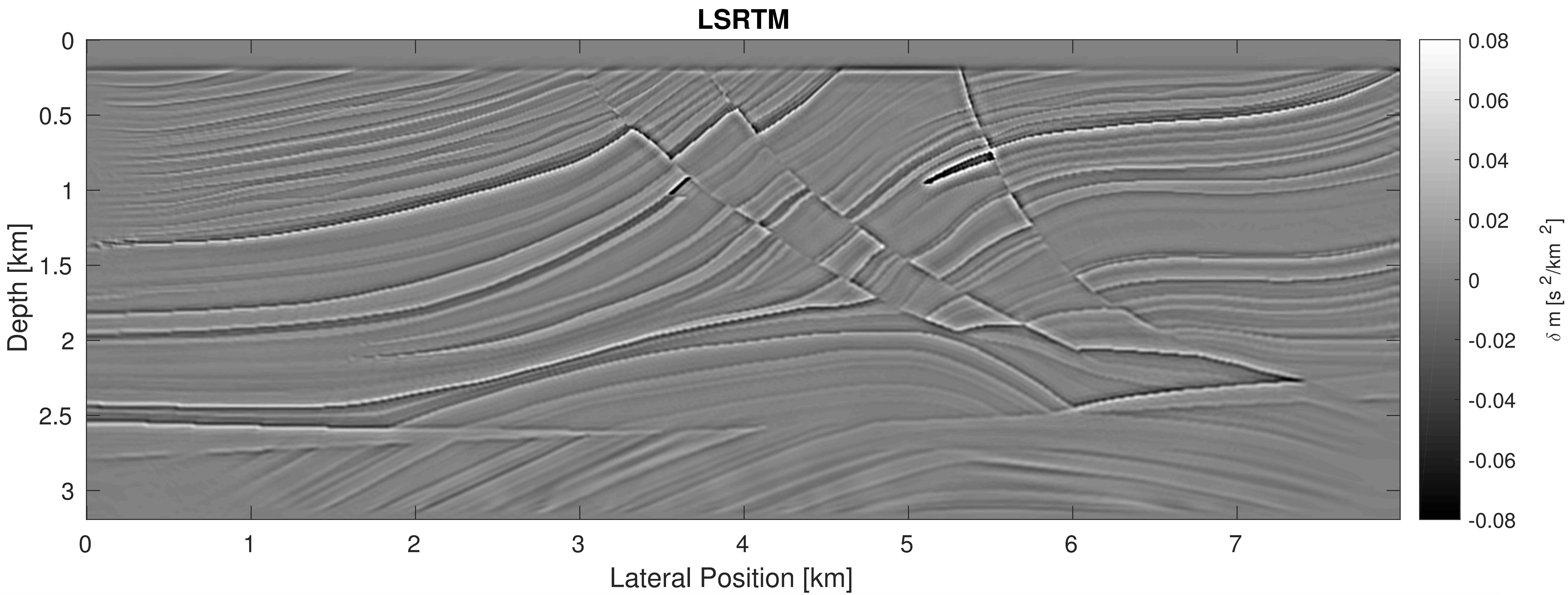
SP-LSRTM: Marmousi



SP-LSRTM: Marmousi

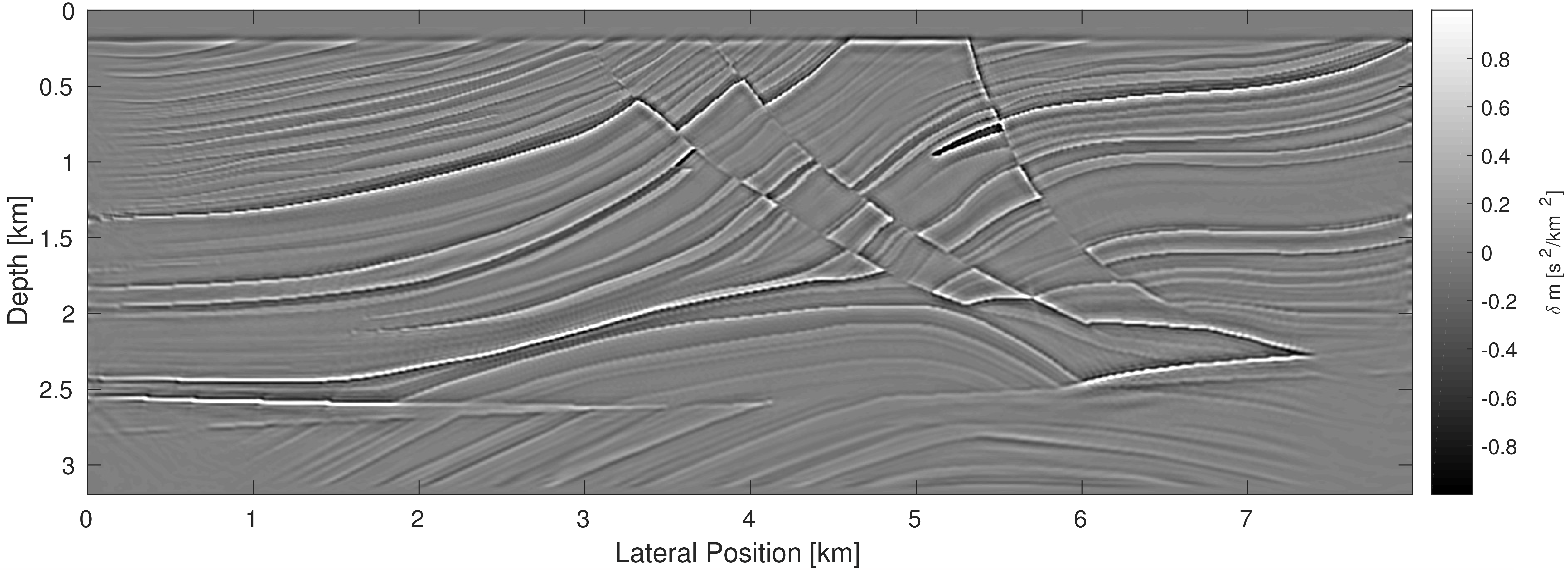


SP-LSRTM: Marmousi



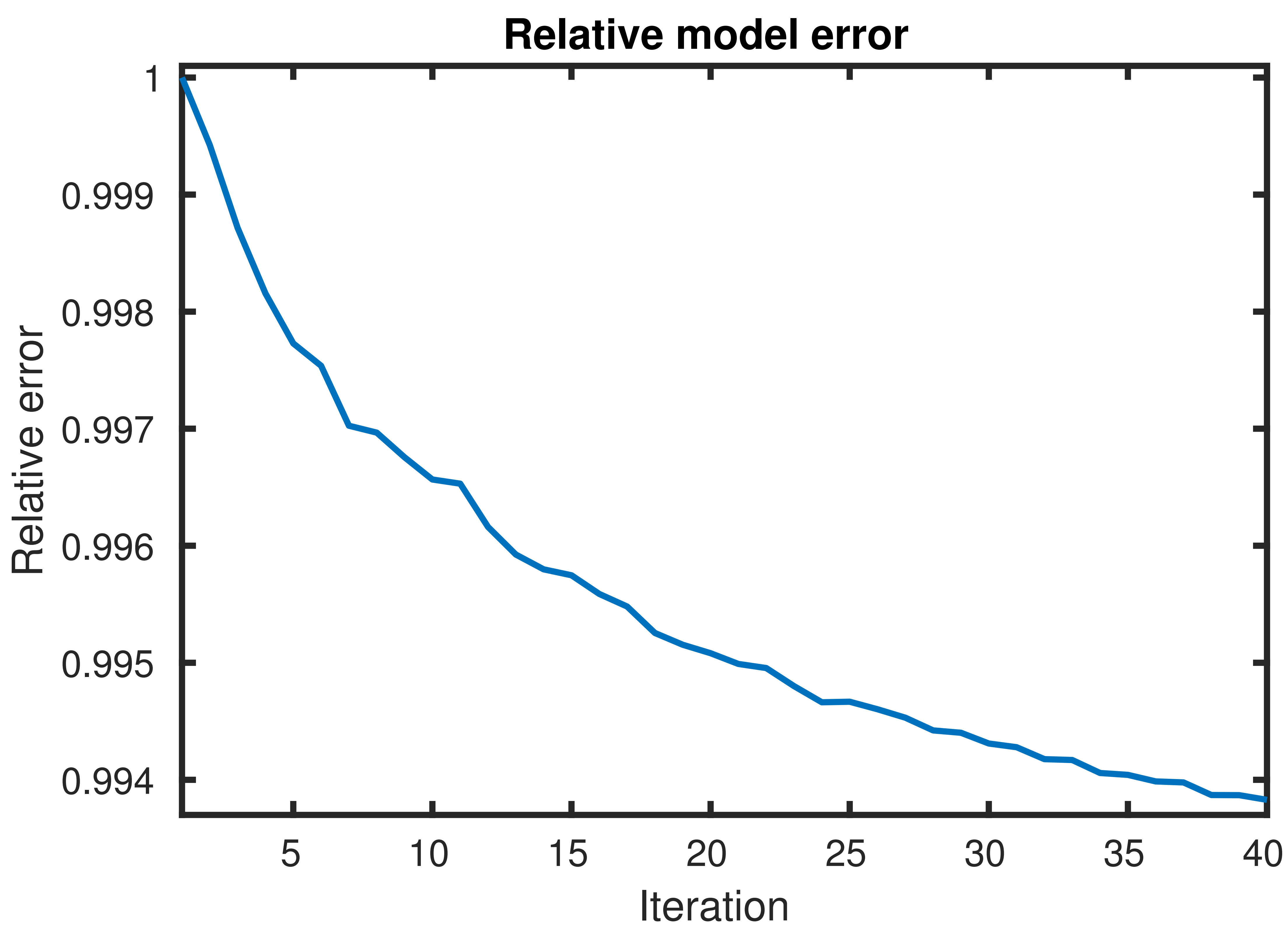
Inverse crime

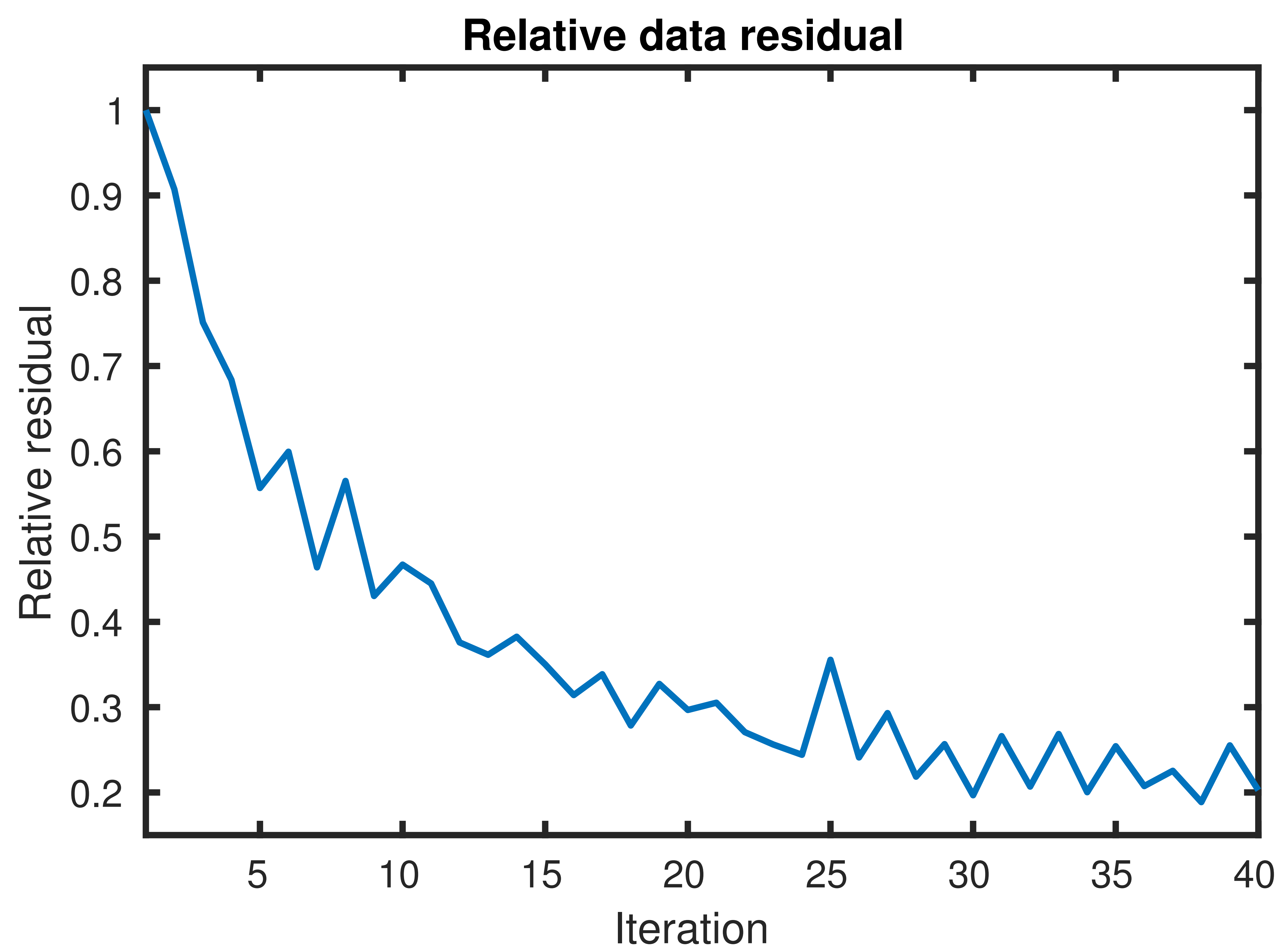
SP-LSRTM: Marmousi



Non-inverse crime (observed data modeled with i-wave)

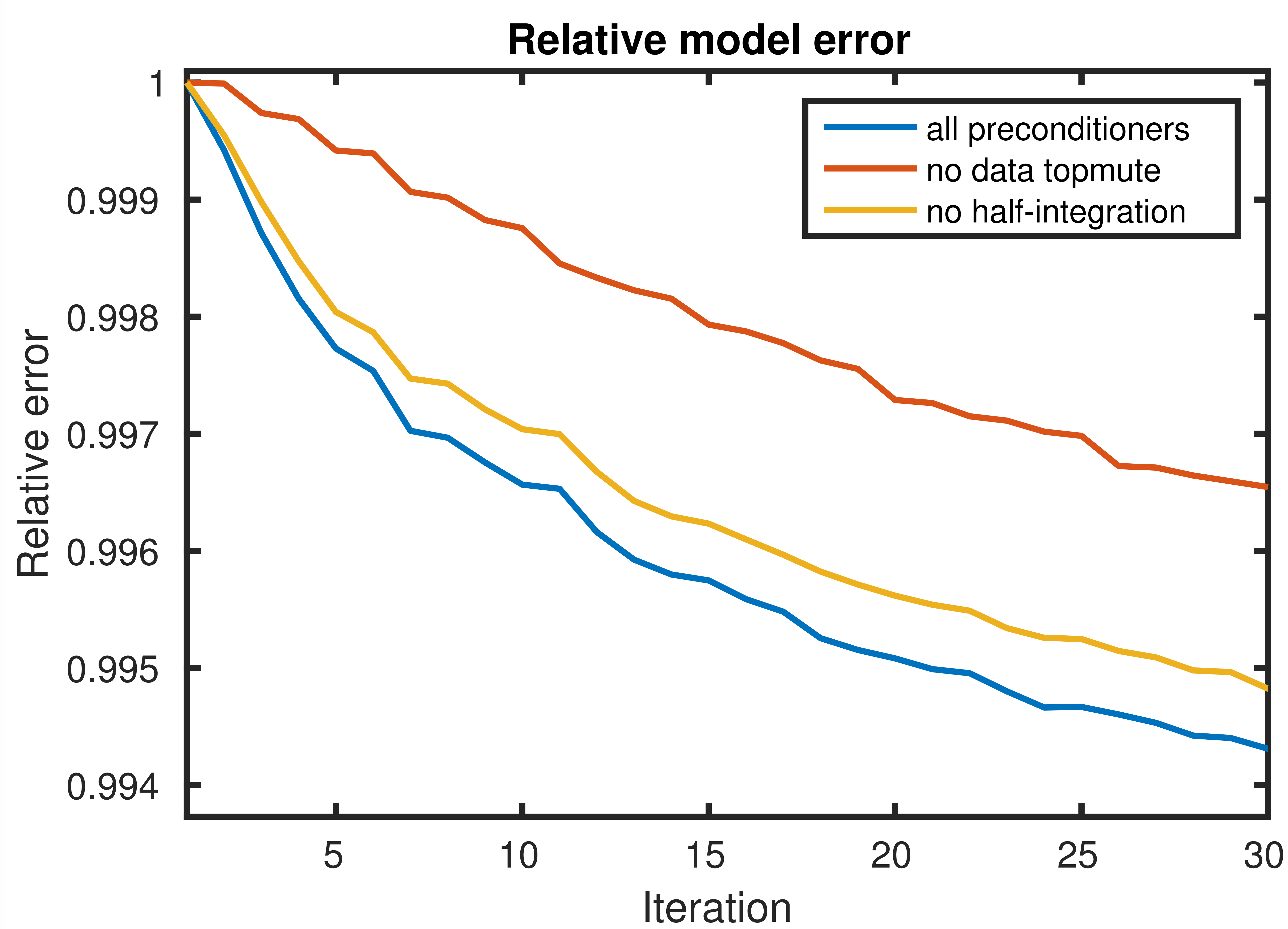
SP-LSRTM: Marmousi





SP-LSRTM: Marmousi

Influence of pre-conditioners on model error:



SP-LSRTM examples w/ linearized data

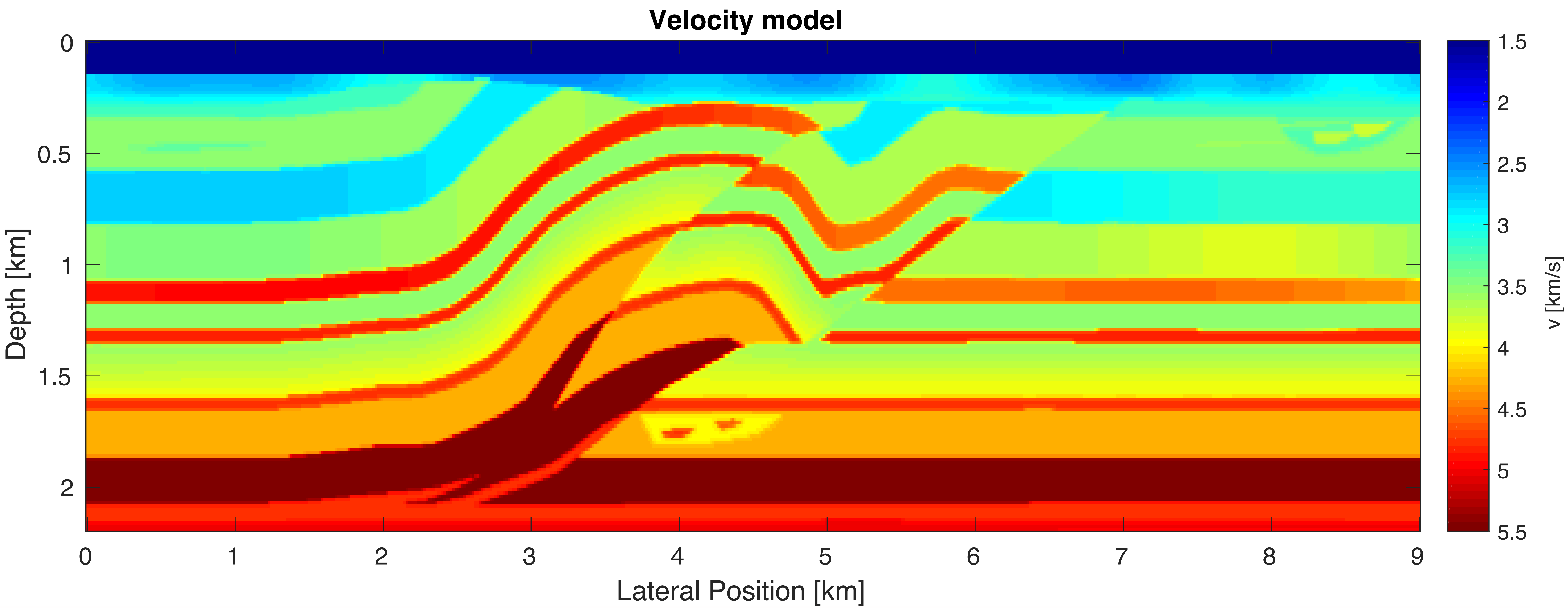
Overthrust model

- 360 shots, 4 seconds recording time
- 30 Hz Ricker wavelet
- 25 m source spacing
- OBNs with 10 m receiver spacing

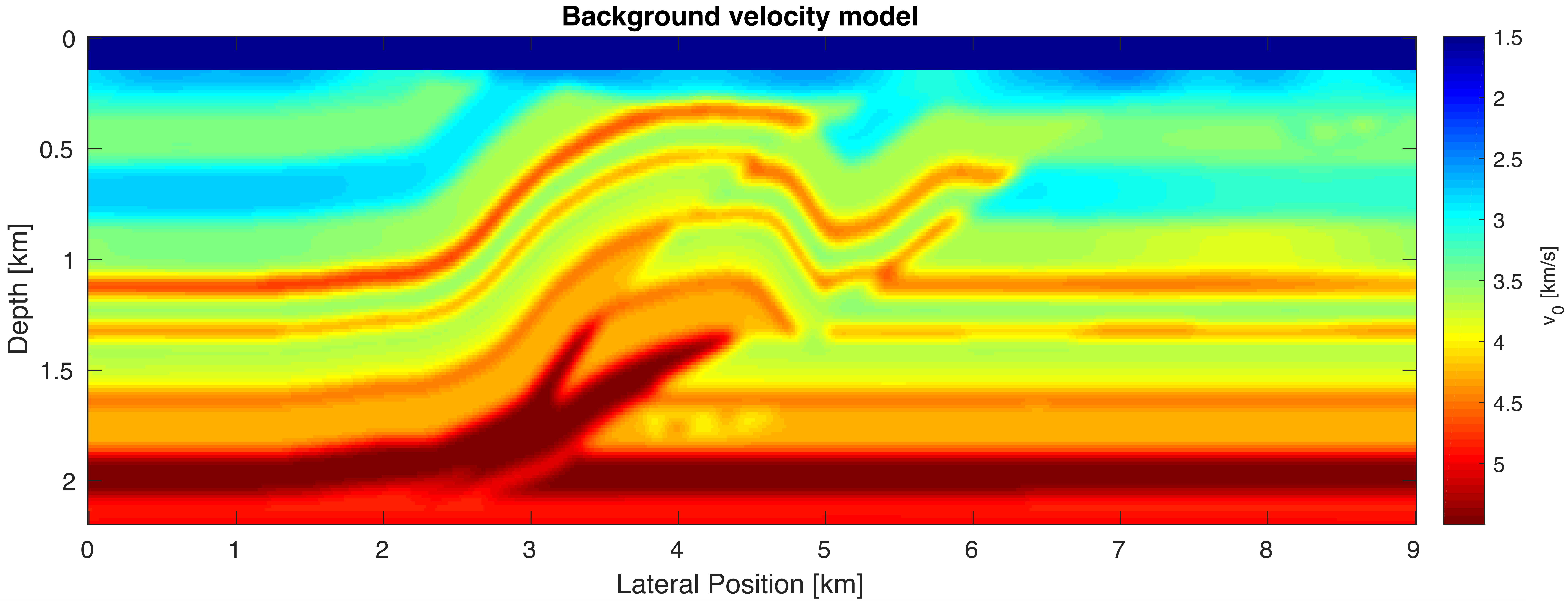
Inversion parameters

- 30 iterations
- 16 shots per iteration (1.3 data passes)
- linearized observed data (inversion crime)

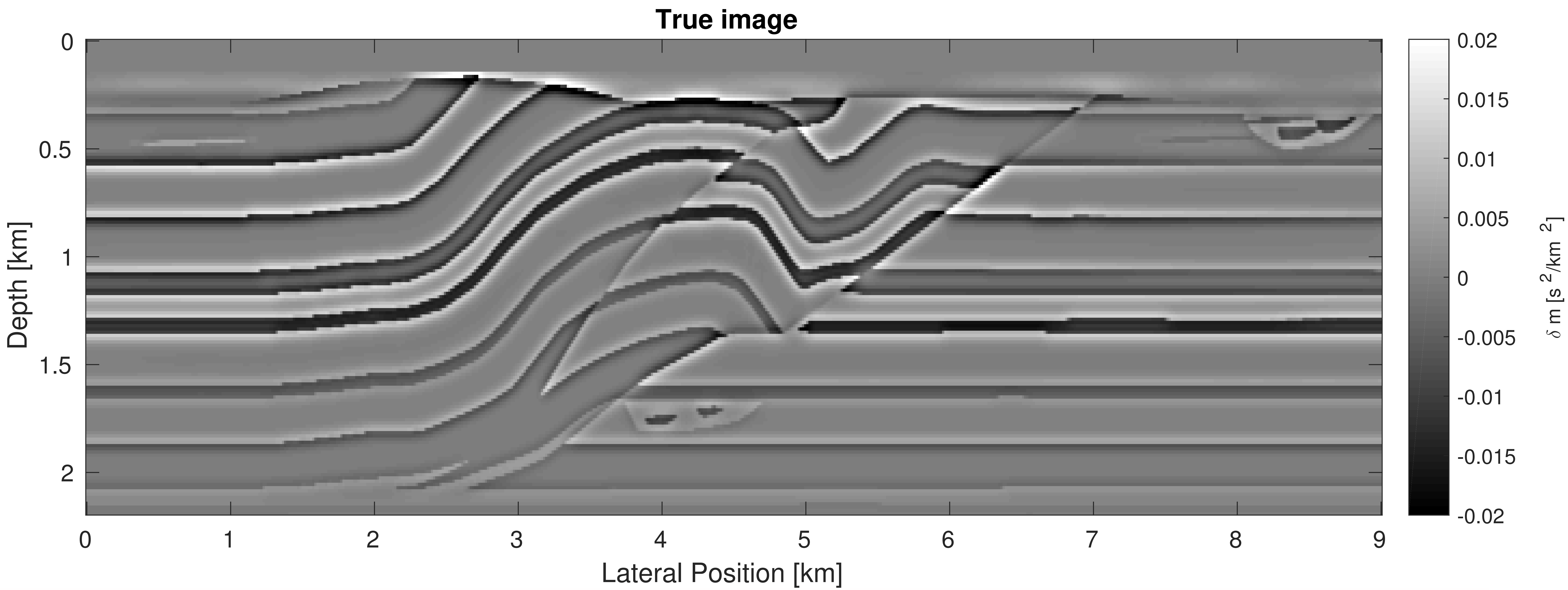
SP-LSRTM: Overthrust



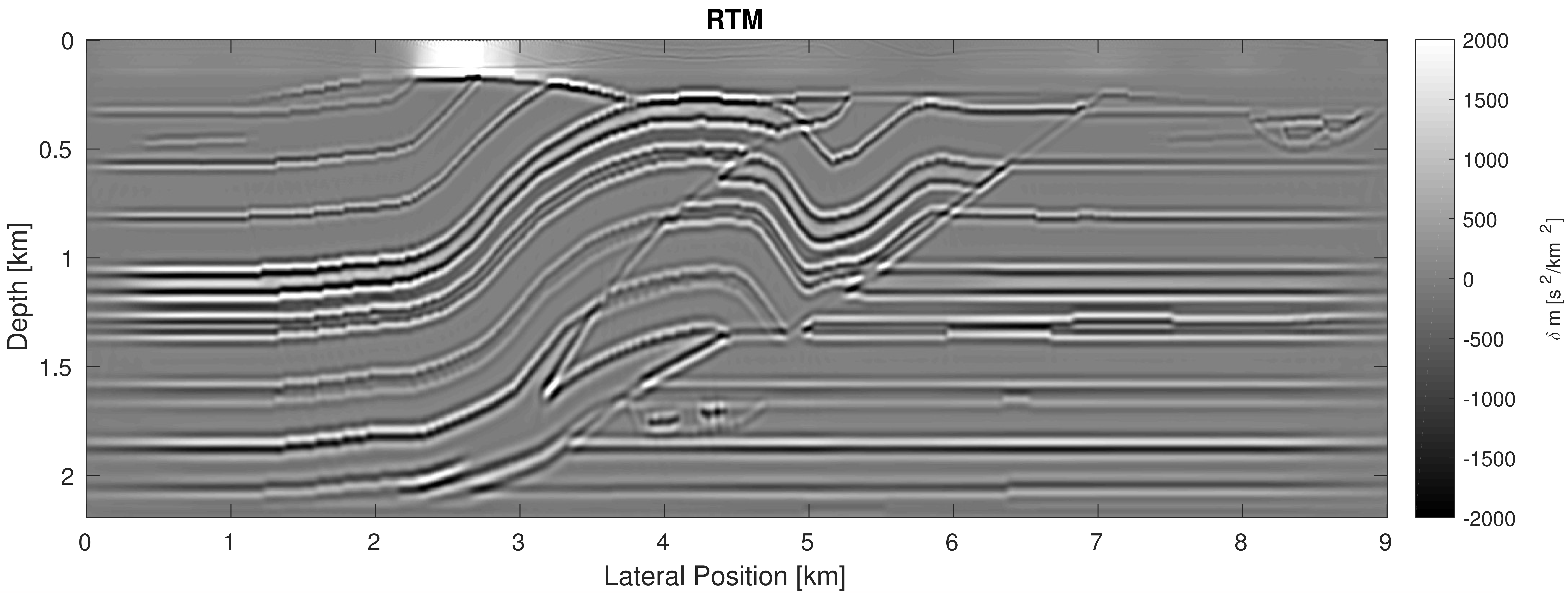
SP-LSRTM: Overthrust



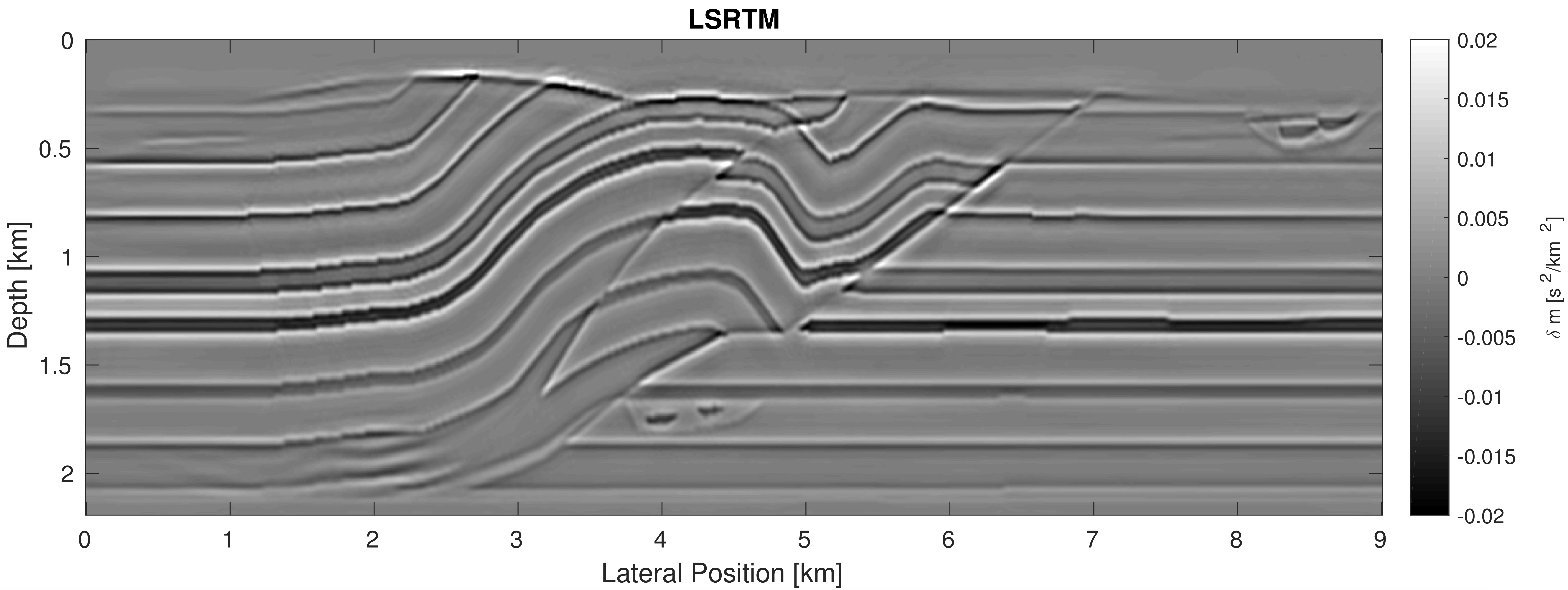
SP-LSRTM: Overthrust



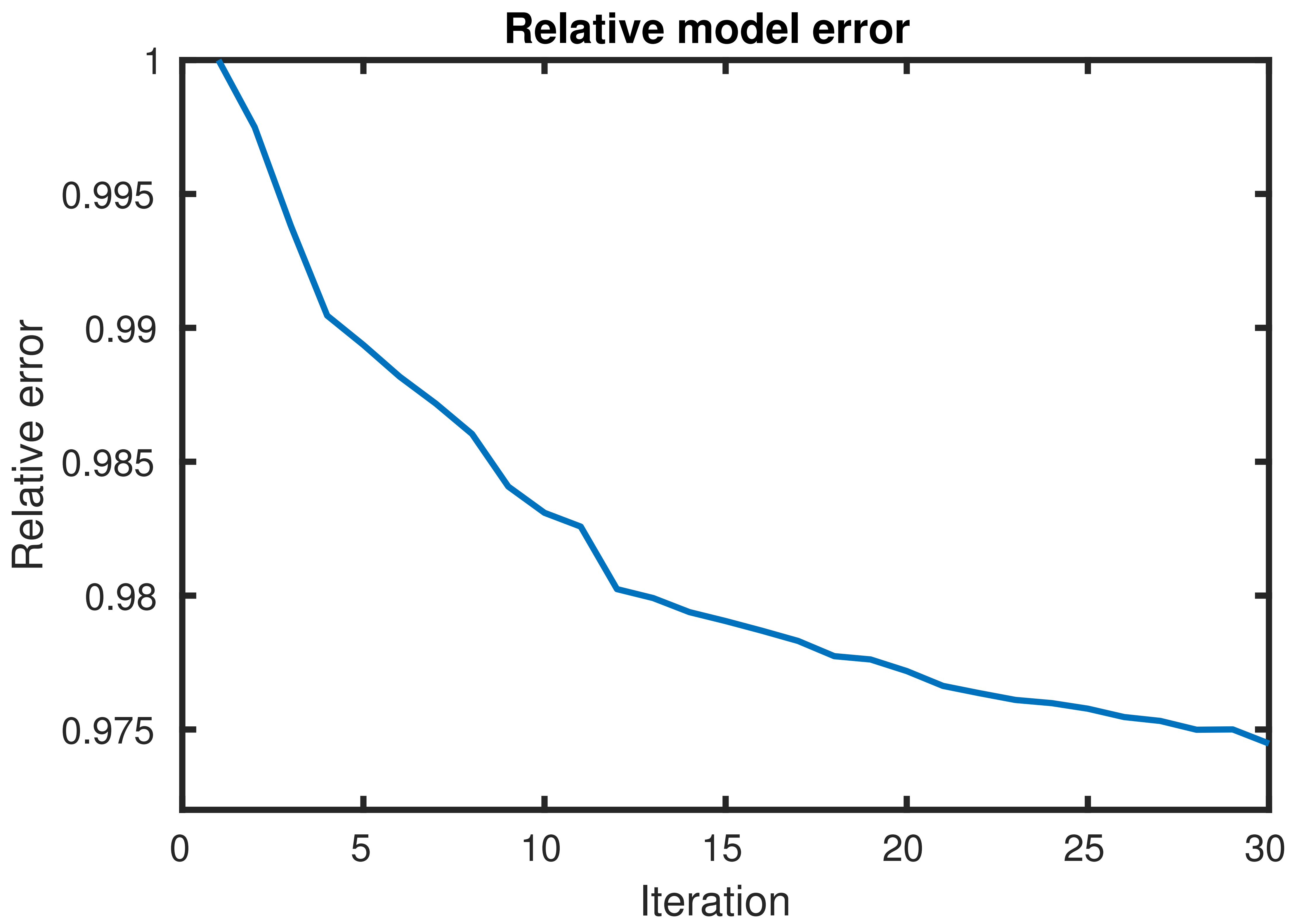
SP-LSRTM: Overthrust

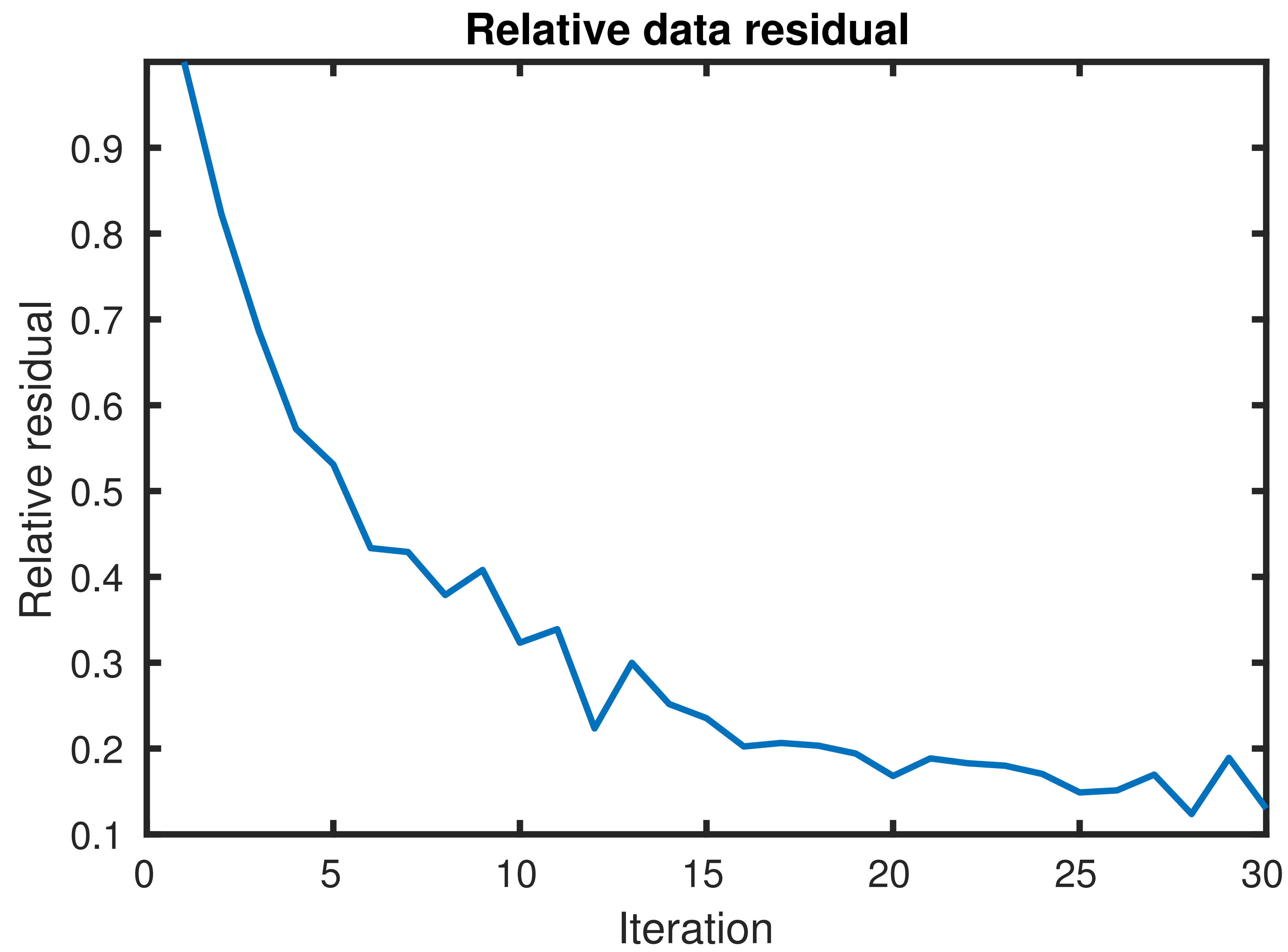


SP-LSRTM: Overthrust



SP-LSRTM: Overthrust





SP-LSRTM with non-linearized data

Sparsity promoting LSRTM with linearized data:

- inverted image close to true image
- noticeable improvement compared to RTM

How does our algorithm behave for non-linearized data?

- Amplitudes of observed data and modeled linearized data can never match

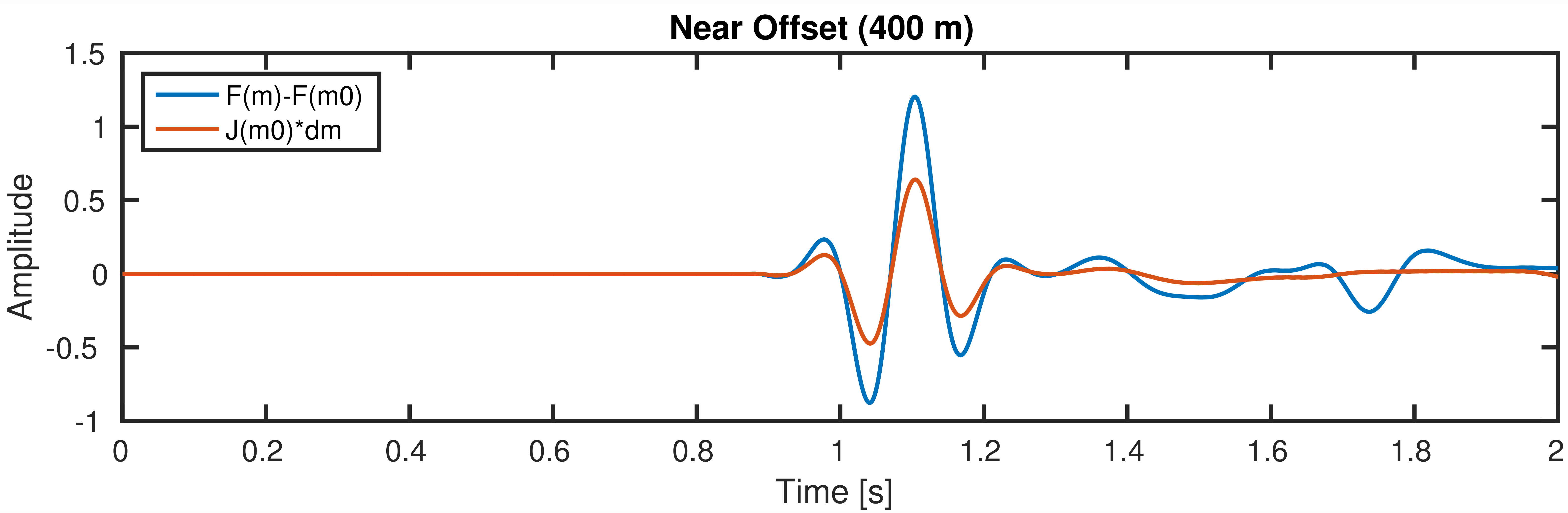
SP-LSRTM with non-linearized data

Sparsity promoting LSRTM with linearized data:

- inverted image close to true image
- noticeable improvement compared to RTM

How does our algorithm behave for non-linearized data?

- Amplitudes of observed data and modeled linearized data can never match



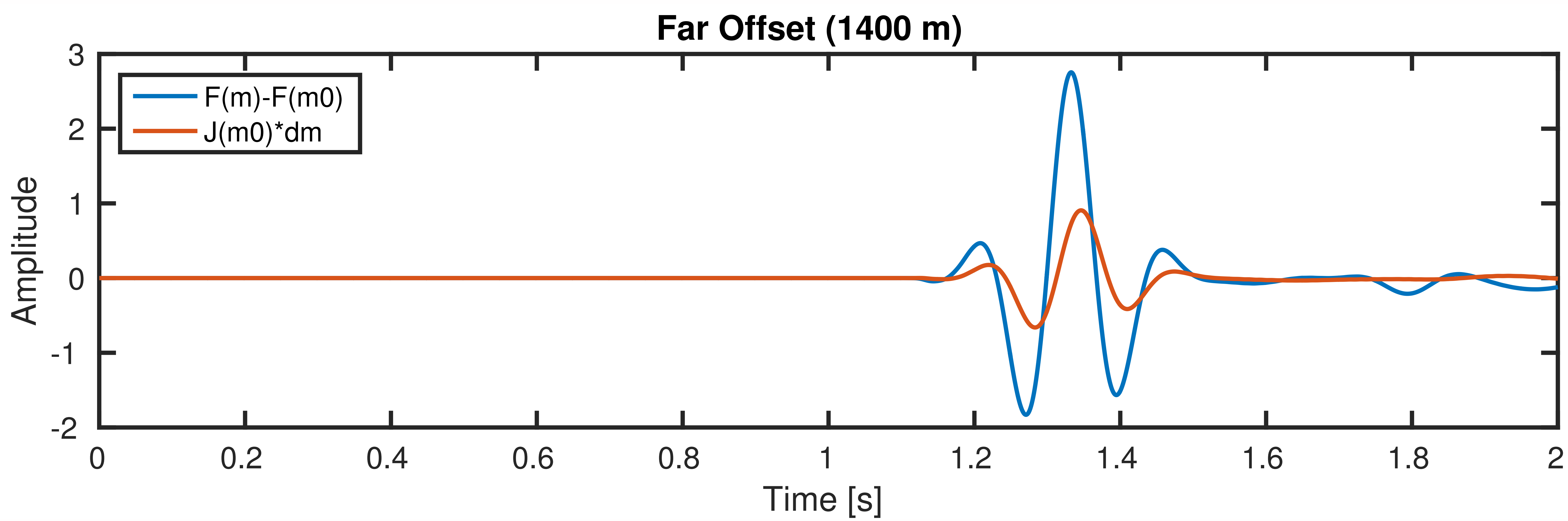
SP-LSRTM with non-linearized data

Sparsity promoting LSRTM with linearized data:

- inverted image close to true image
- noticeable improvement compared to RTM

How does our algorithm behave for non-linearized data?

- Amplitudes of observed data and modeled linearized data can never match



SP-LSRTM with non-linearized data

Sparsity promoting LSRTM with linearized data:

- inverted image close to true image
- noticeable improvement compared to RTM

How does our algorithm behave for non-linearized data?

- Amplitudes of observed data and modeled linearized data can never match

What is the influence of having a correct Jacobian pair?

```
for j=1:maxiter

    # predicted data
    d_pred = J*dm;

    # gradient
    g = J'*(d_pred - d_obs)

    # update image
    dm = dm - t*g

end
```

versus

```
for j=1:maxiter

    # predicted data
    d_pred = model(m0 + dm);

    # gradient
    g = rtm(d_pred - d_obs)

    # update image
    dm = dm - t*g

end
```

SP-LSRTM algorithm w/o Jacobians:

1. **for** $k = 0, 1, \cdots$
2. Demigration:
   $$\delta \mathbf{d}_k = \mathbf{M}_L^{-1} \mathbf{F}_{r(k)} (\mathbf{m}_0 + \mathbf{M}_R^{-1} \mathbf{x}_k) - \mathbf{M}_L^{-1} \mathbf{b}_{r(k)}$$
3. Migration of data residual:
   $$\delta \mathbf{m}_k = (\mathbf{M}_R^{-1})^T \mathbf{F}_{r(k)}^T \Big((\mathbf{M}_L^{-1})^T \delta \mathbf{d}_k \Big)$$
4. Vertical derivative:
   $$\delta \mathbf{m}_k = \mathbf{D}_z \delta \mathbf{m}_k$$
5. Update z:
   $$\mathbf{z}_{k+1} = \mathbf{z}_k - t_k \cdot \delta \mathbf{m}_k$$
6. Update x:
   $$\mathbf{x}_{k+1} = \mathbf{C}^* S_\lambda (\mathbf{C} \mathbf{z}_{k+1})$$
7. Bound projections:
8. $(\mathbf{z}_{k+1}, \mathbf{x}_{k+1}) = \mathcal{P}_{Breg} (\mathbf{z}_{k+1}, \mathbf{x}_{k+1})$
9. **end for**

Projection operator

Projection operator for bound constraints w/ linearized Bregman:

$$(\tilde{\mathbf{z}}, \tilde{\mathbf{x}}) = \mathcal{P}_{Breg}(\mathbf{z}, \mathbf{x})$$

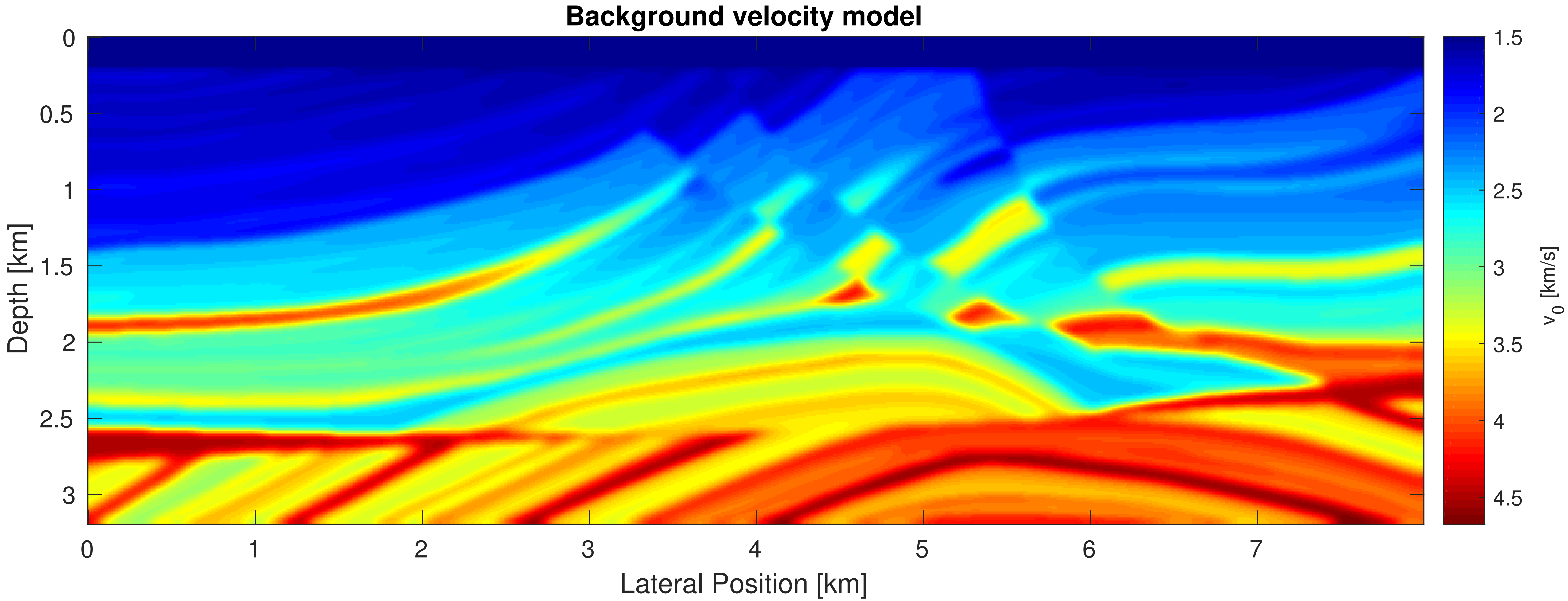
$$\tilde{\mathbf{x}} = \mathcal{P}_B\Big(S_\lambda(\mathbf{x})\Big) = \text{median}\Big(\mathbf{a}, S_\lambda(\mathbf{x}), \mathbf{b}\Big)$$

$\mathbf{a}$: lower bound

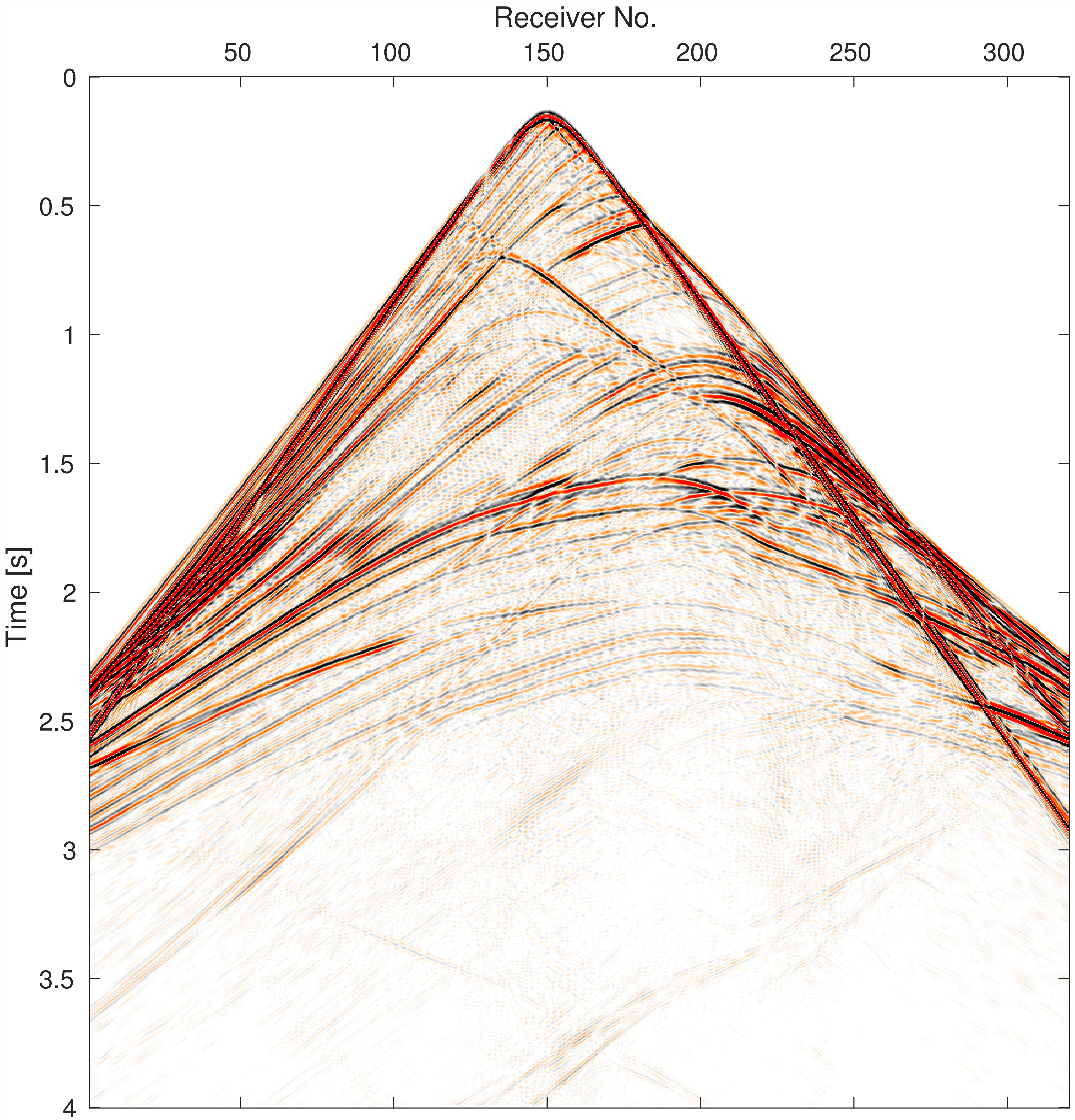
$\mathbf{b}$: upper bound

$$\tilde{\mathbf{z}}_j = \begin{cases} \mathbf{z}_j & \mathbf{a}_j \leq S_\lambda(\tilde{\mathbf{x}})_j \leq \mathbf{b}_j \\ \mathbf{b}_j + \lambda & S_\lambda(\tilde{\mathbf{x}})_j > \mathbf{b}_j \\ \mathbf{a}_j - \lambda & S_\lambda(\tilde{\mathbf{x}})_j < \mathbf{a}_j \end{cases}$$

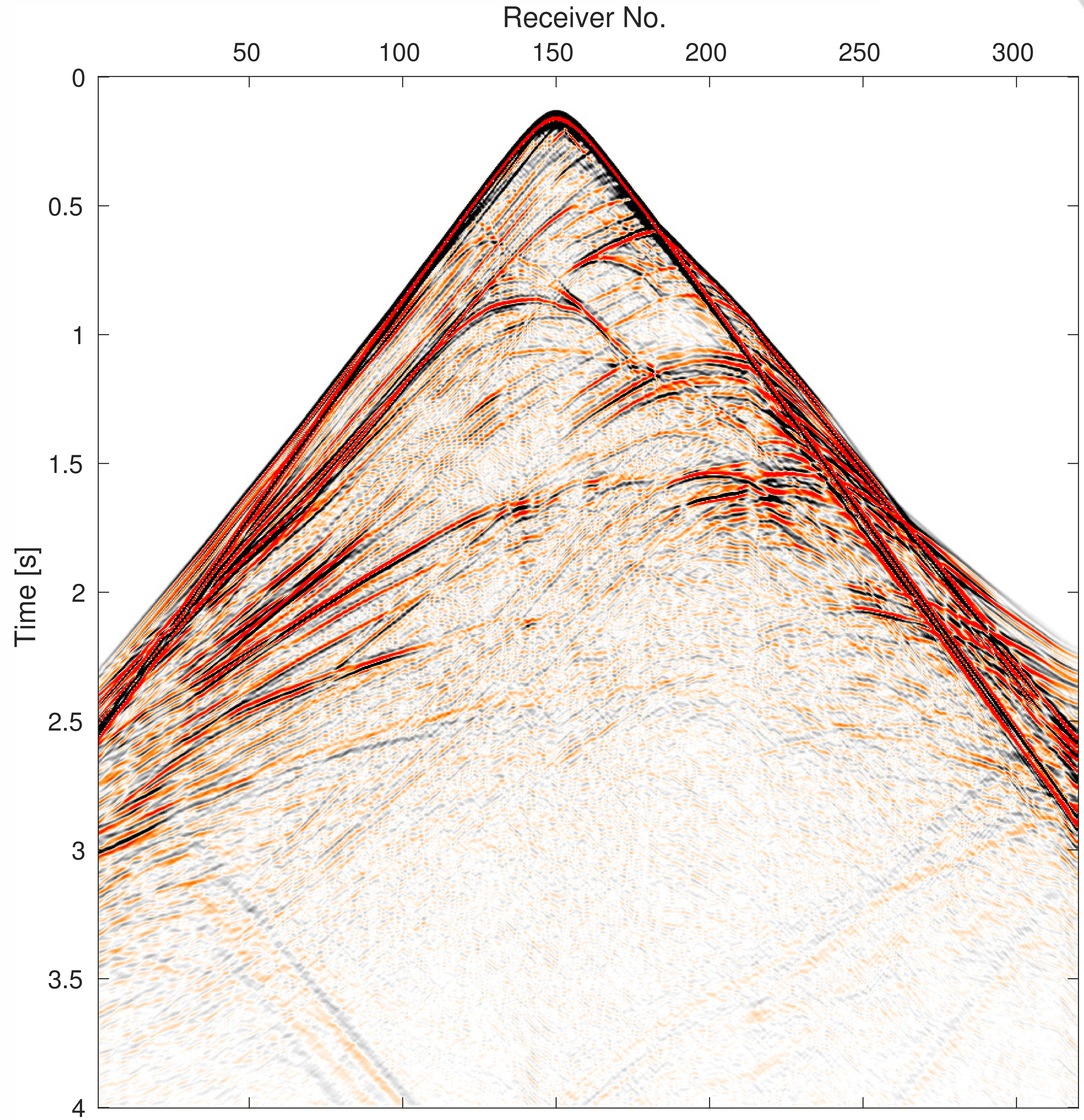
Influence of correct adjoints



Influence of correct adjoints

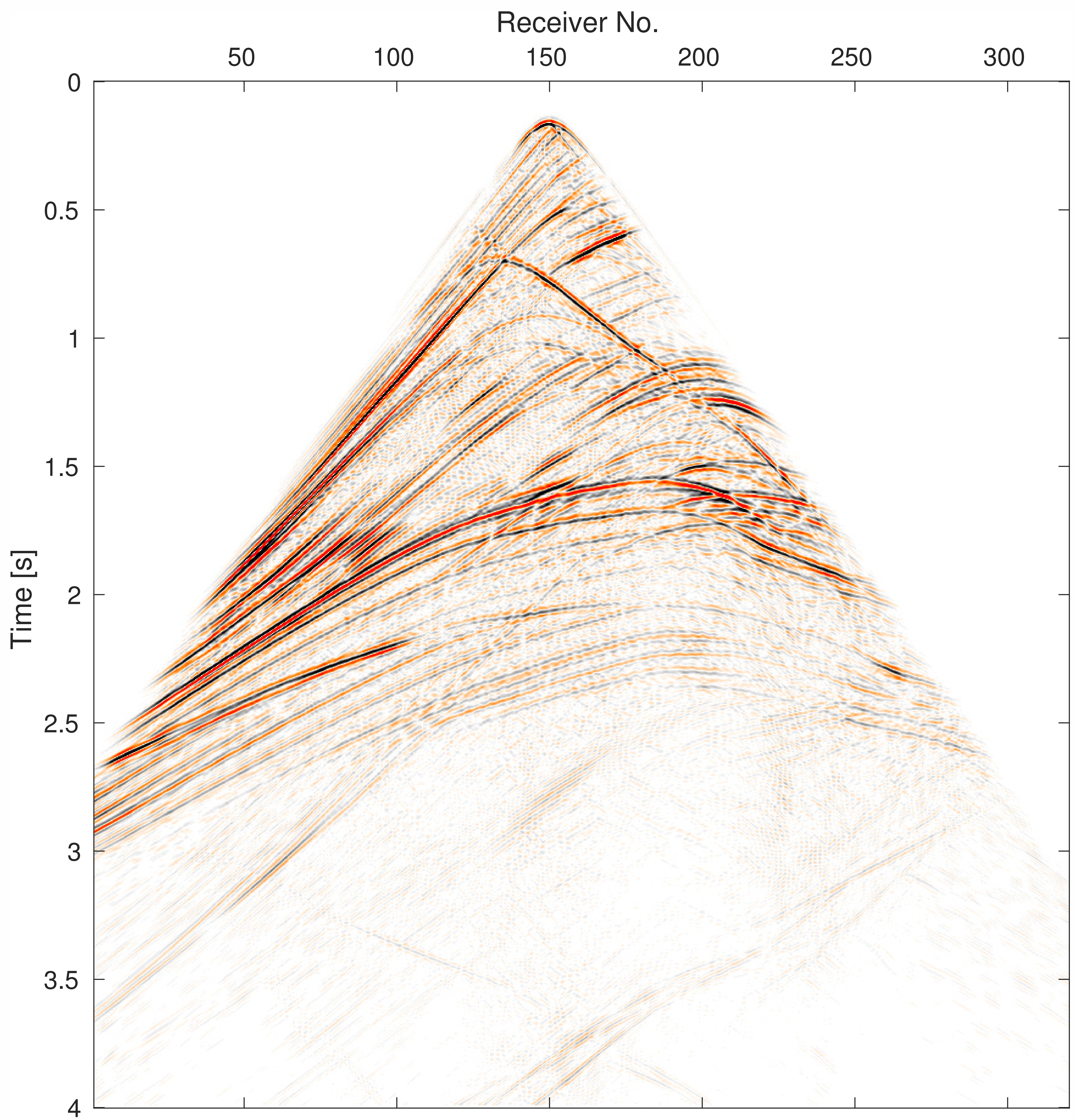


Linearized data

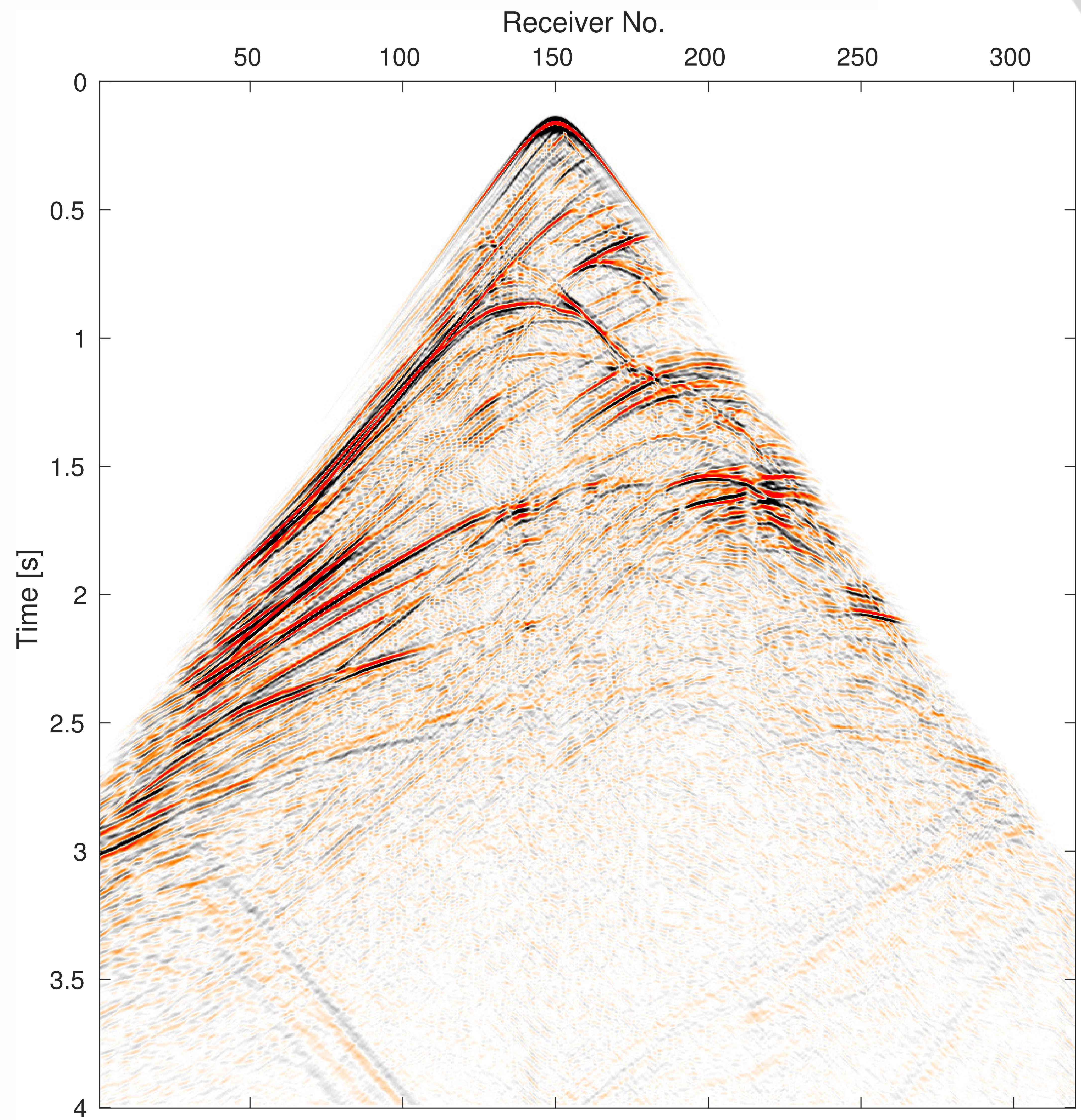


Non-linearized data

Influence of correct adjoints

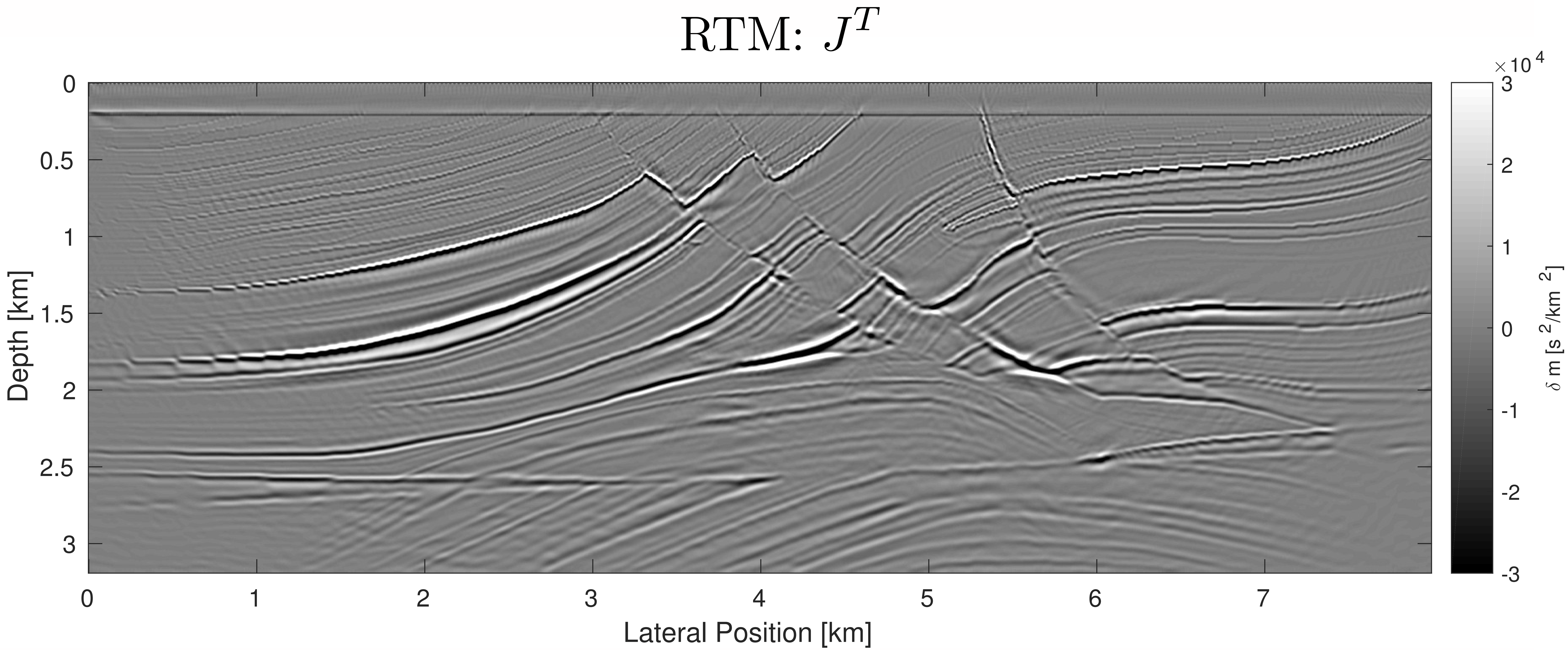


Muted linearized data

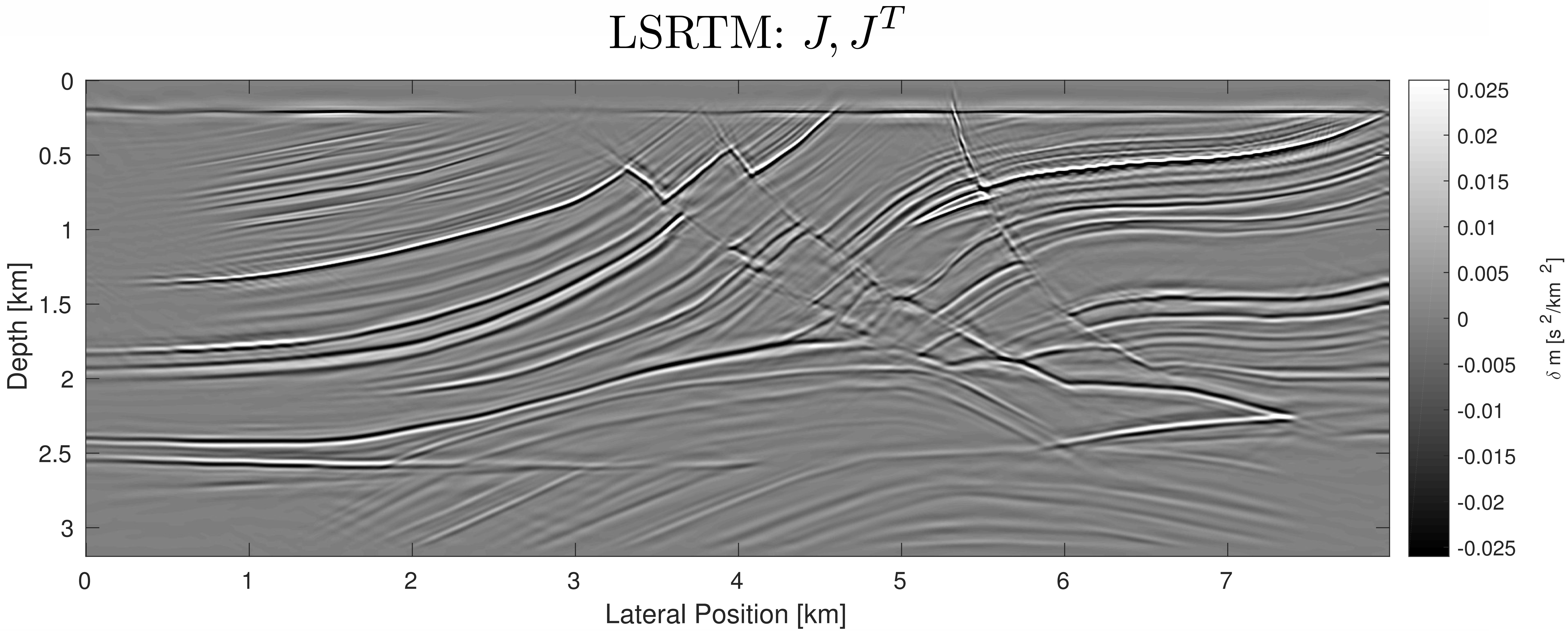


Muted non-linearized data

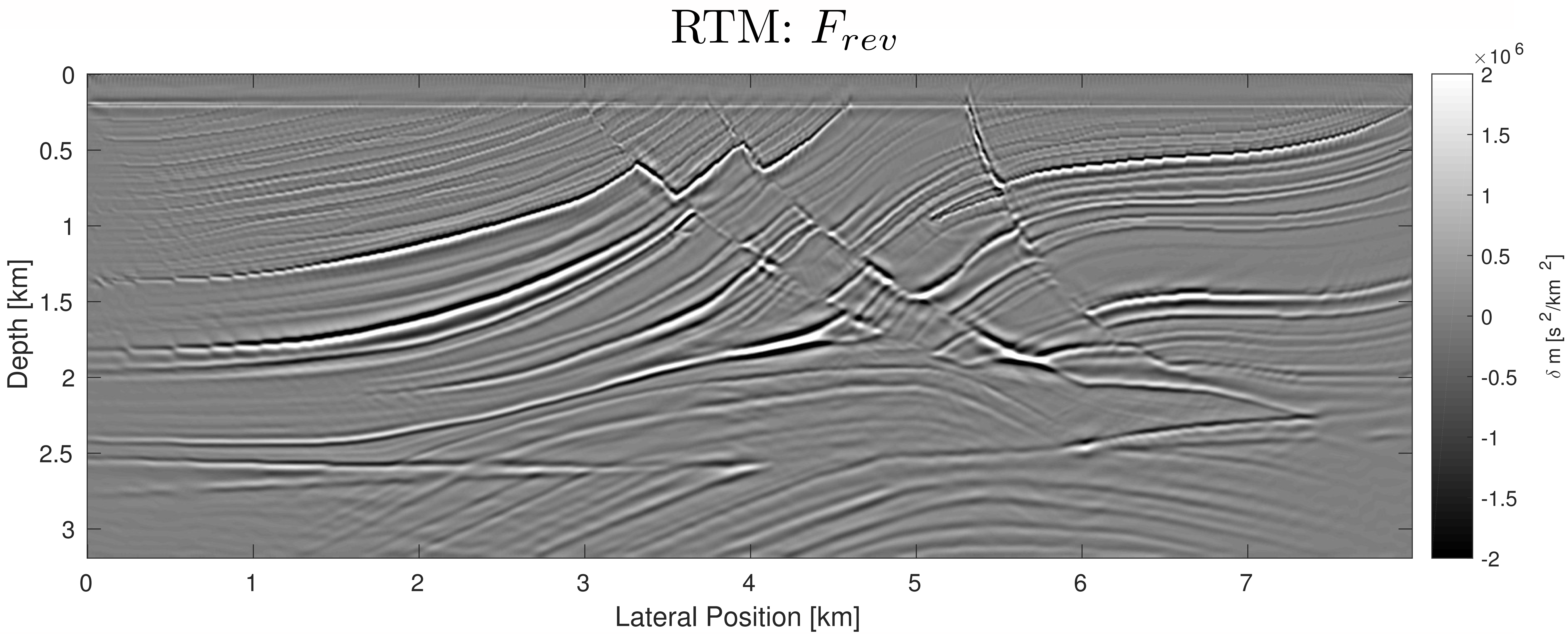
Influence of correct adjoints



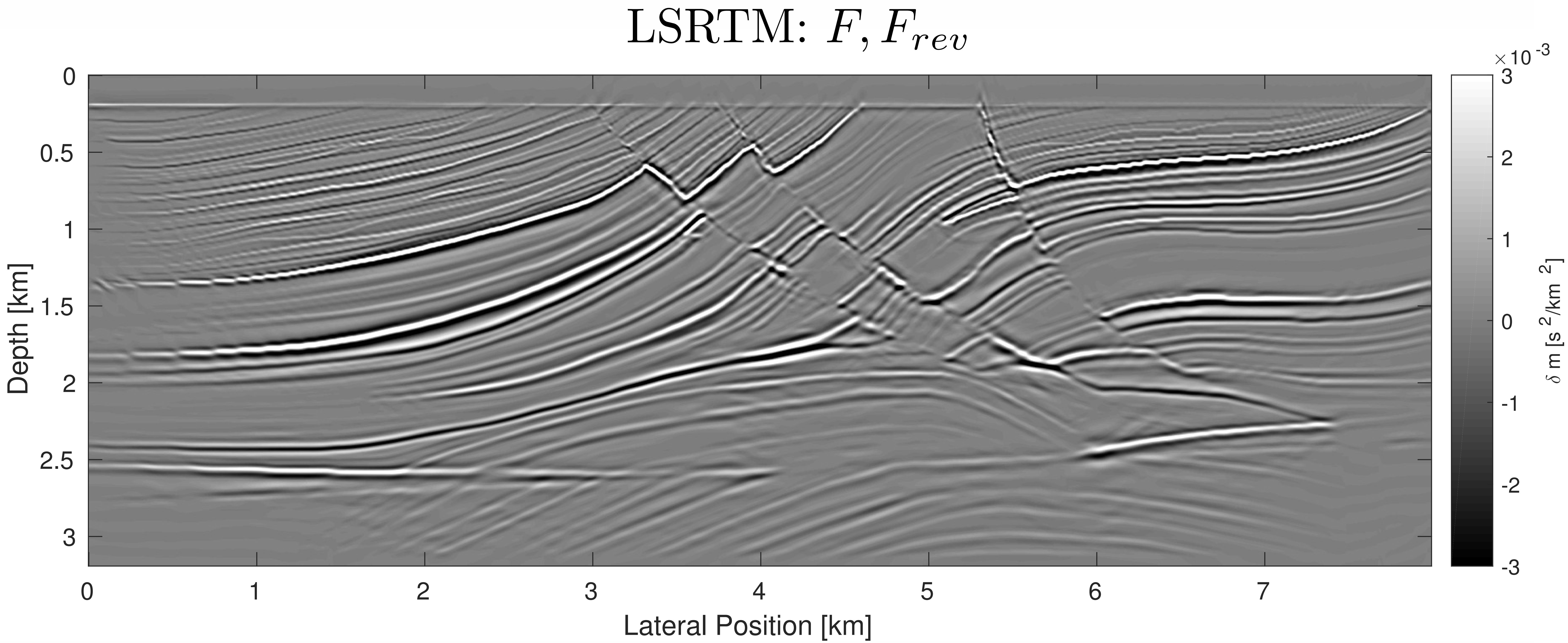
Influence of correct adjoints



Influence of correct adjoints



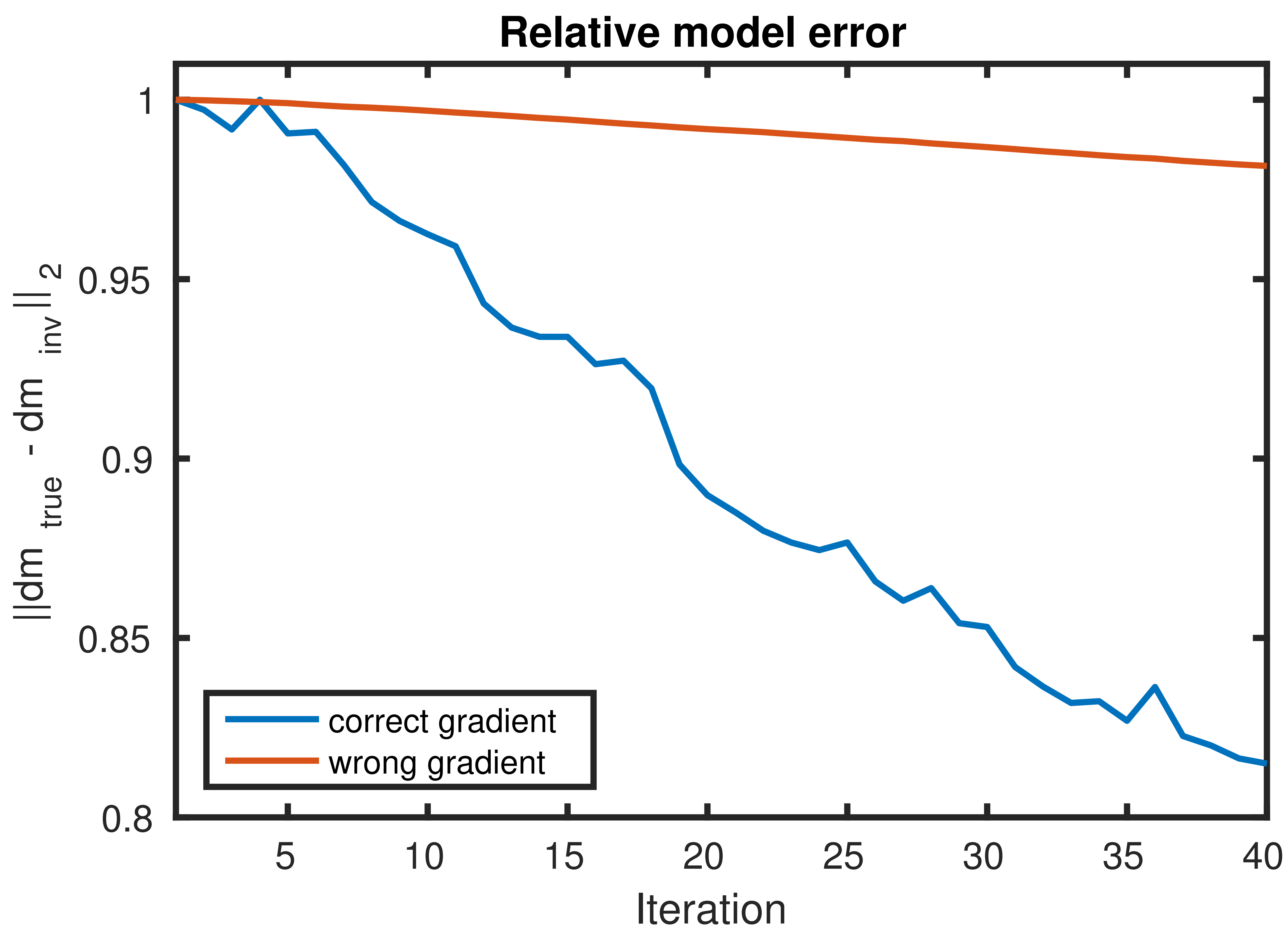
Influence of correct adjoints



Influence of correct adjoints

Sparsity promoting LSRTM w/ correct and incorrect gradient:

- model error decays w/ incorrect gradient
- but: decay w/ correct gradient is much faster



SLIM

Field data example

Sparsity promoting least squares RTM on BP Machar field data set:

- Machar oil field in North sea
- 330 shots w/ 8 seconds recording time
- maximum no. of 505 receivers (OBN)

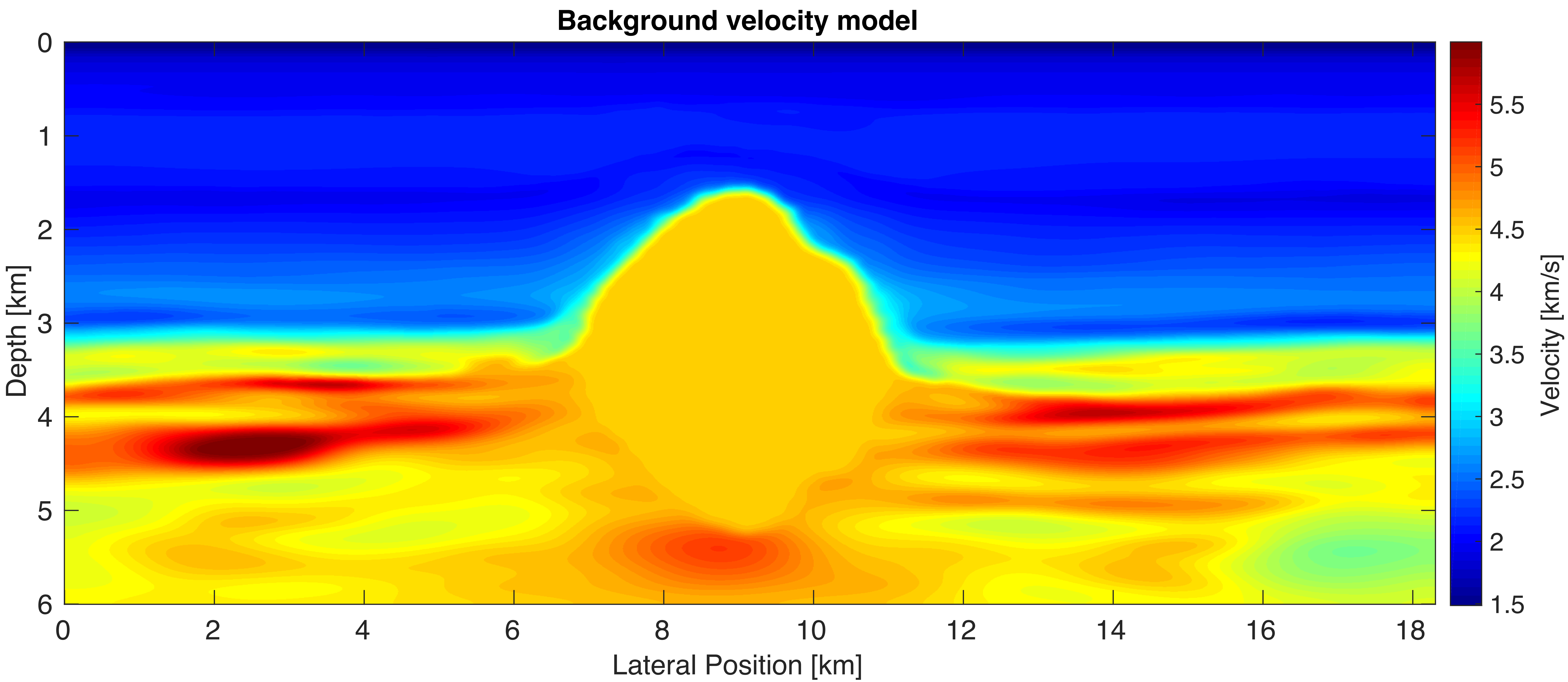
Preprocessing by BP:

- source designature
- mute of direct wave
- multiple removal

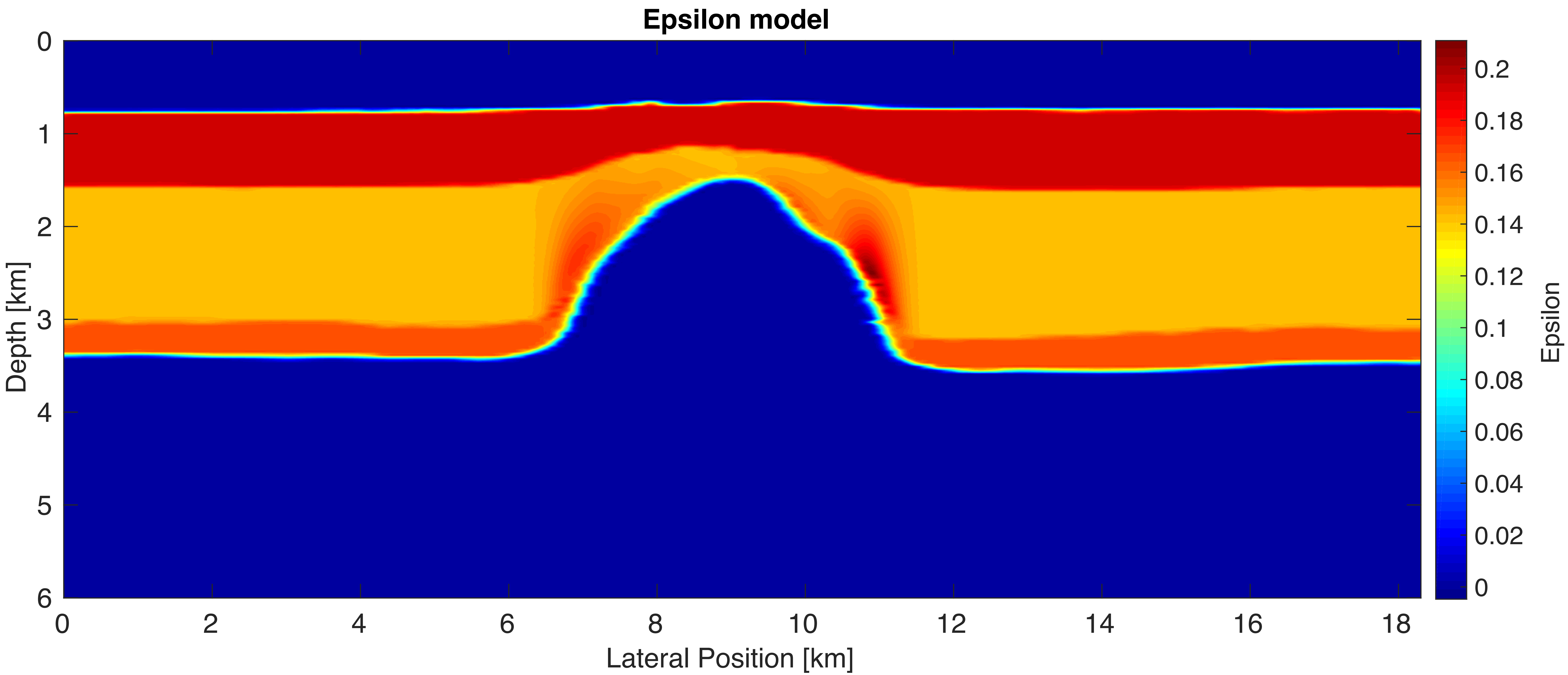
Software:

- Matlab 2D code w/ pseudo-acoustic wave equation
- 10 iterations of linearized Bregman w/ 100 shots per iteration (3 passes through data)
- on the fly source estimation (starting wavelet: 50 Hz Ricker)

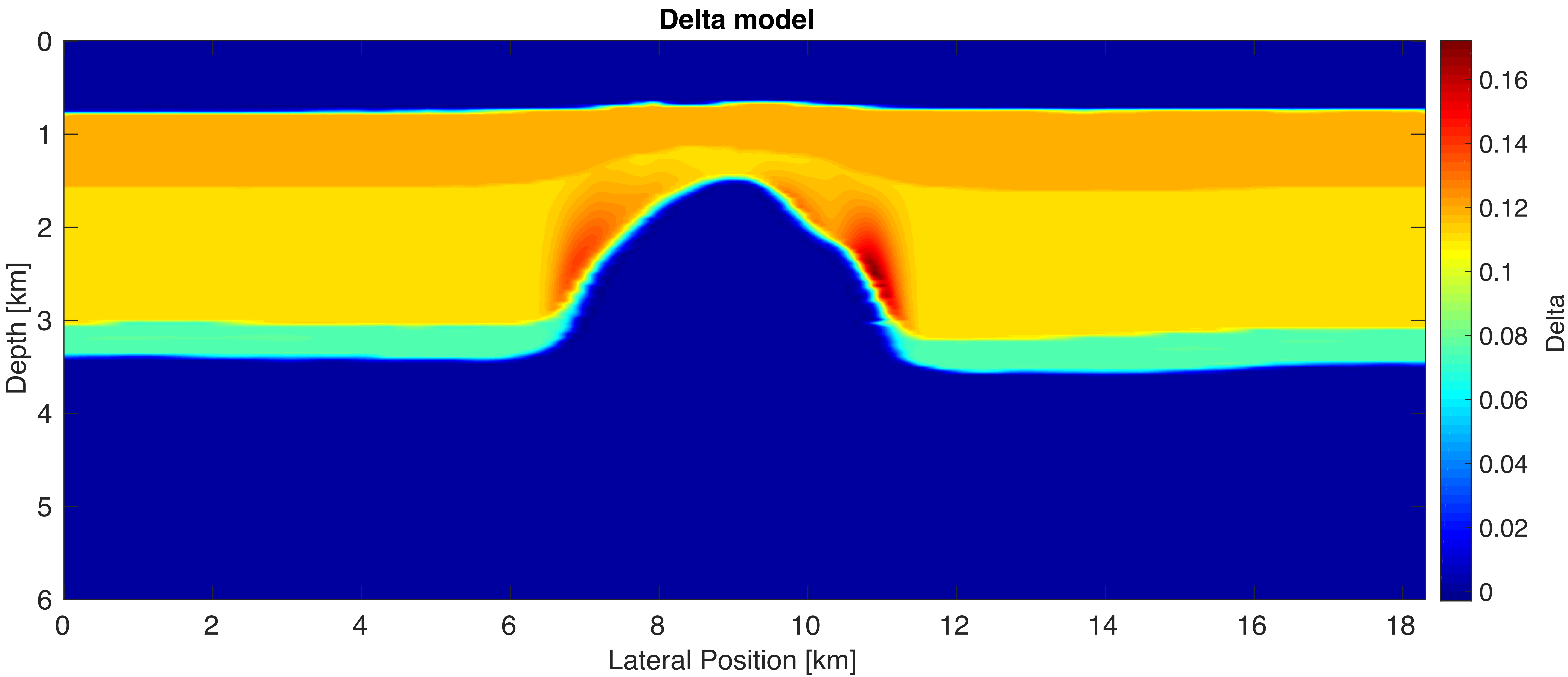
Machar model



Machar model



Machar model



Machar model

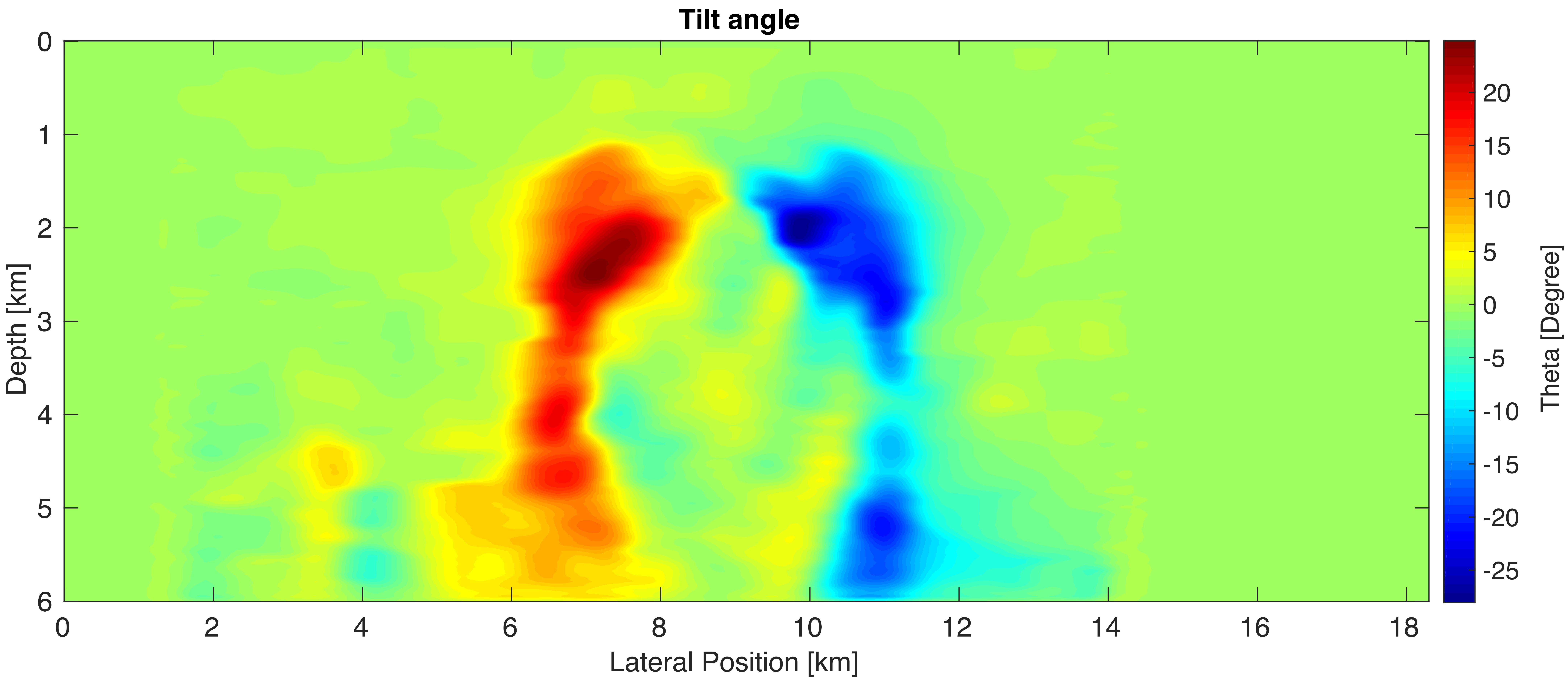


Image updates

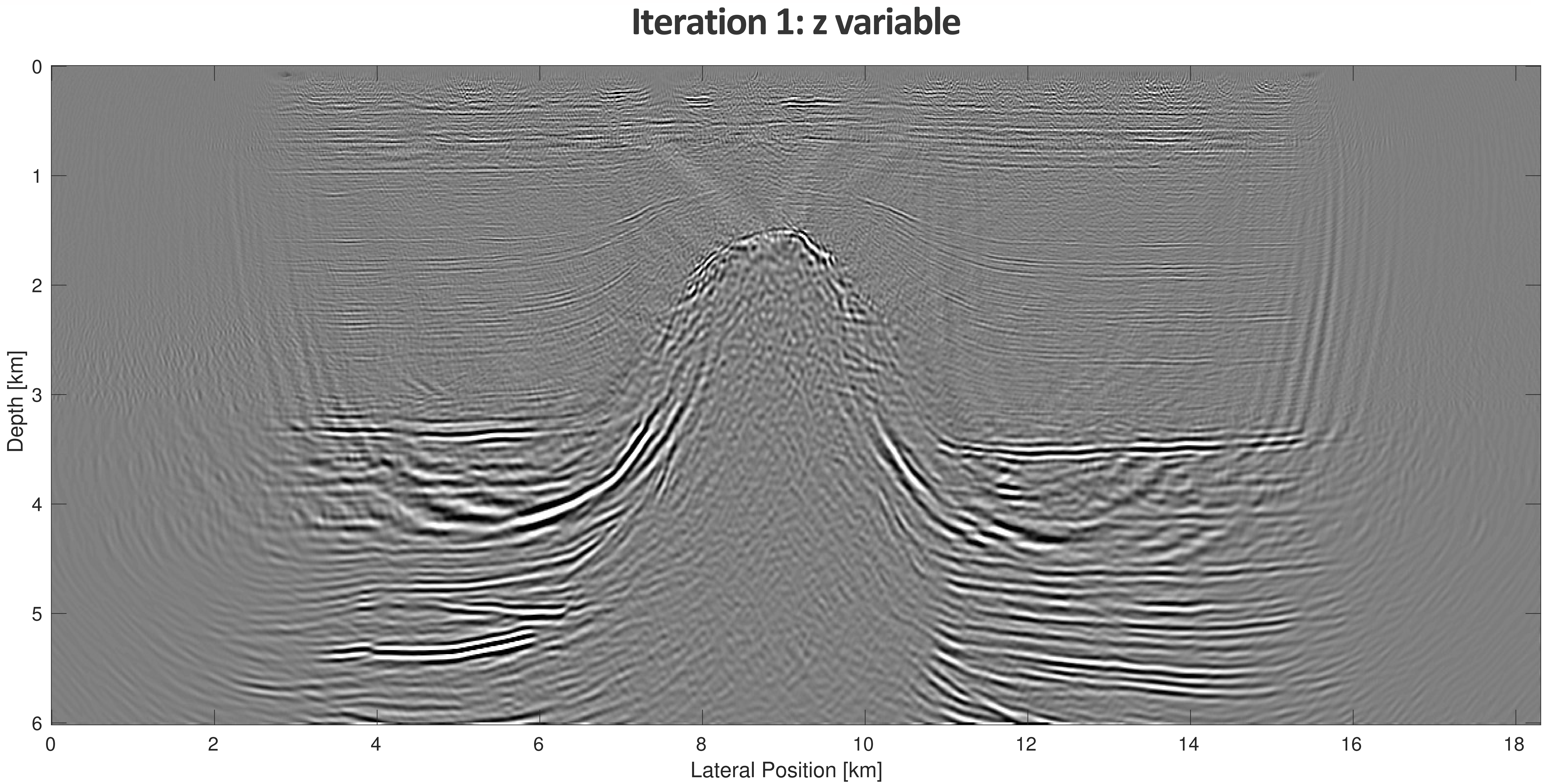
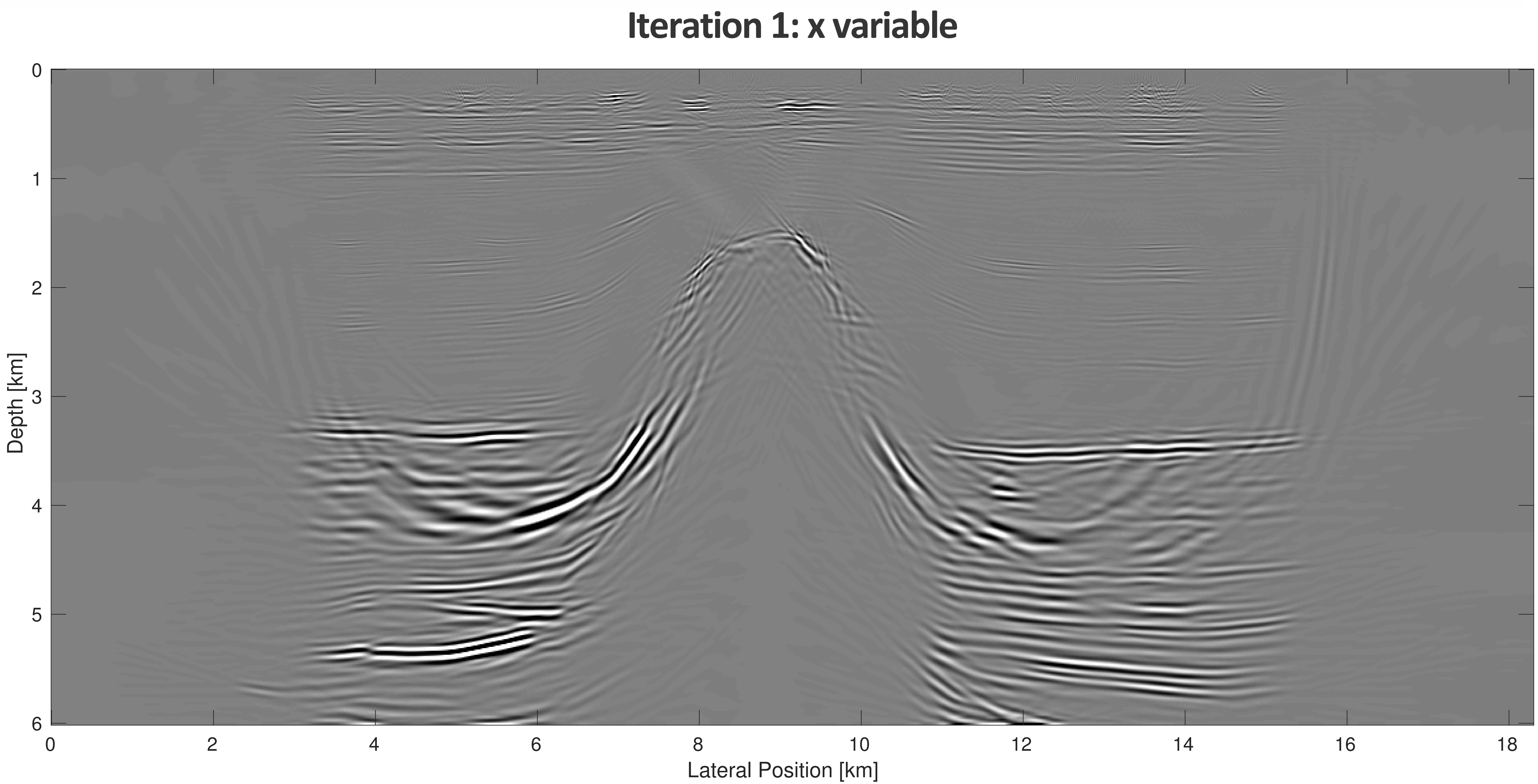
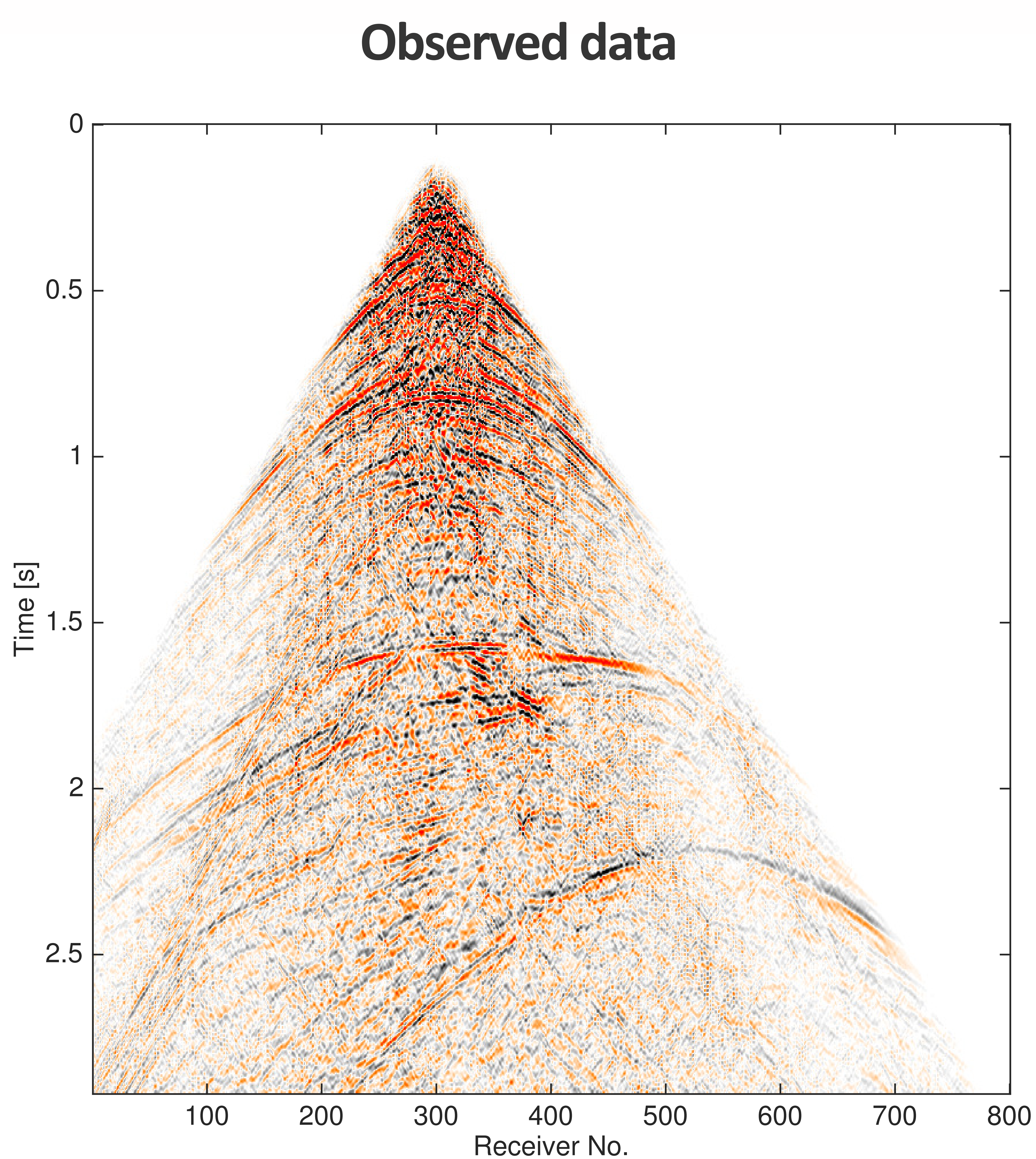
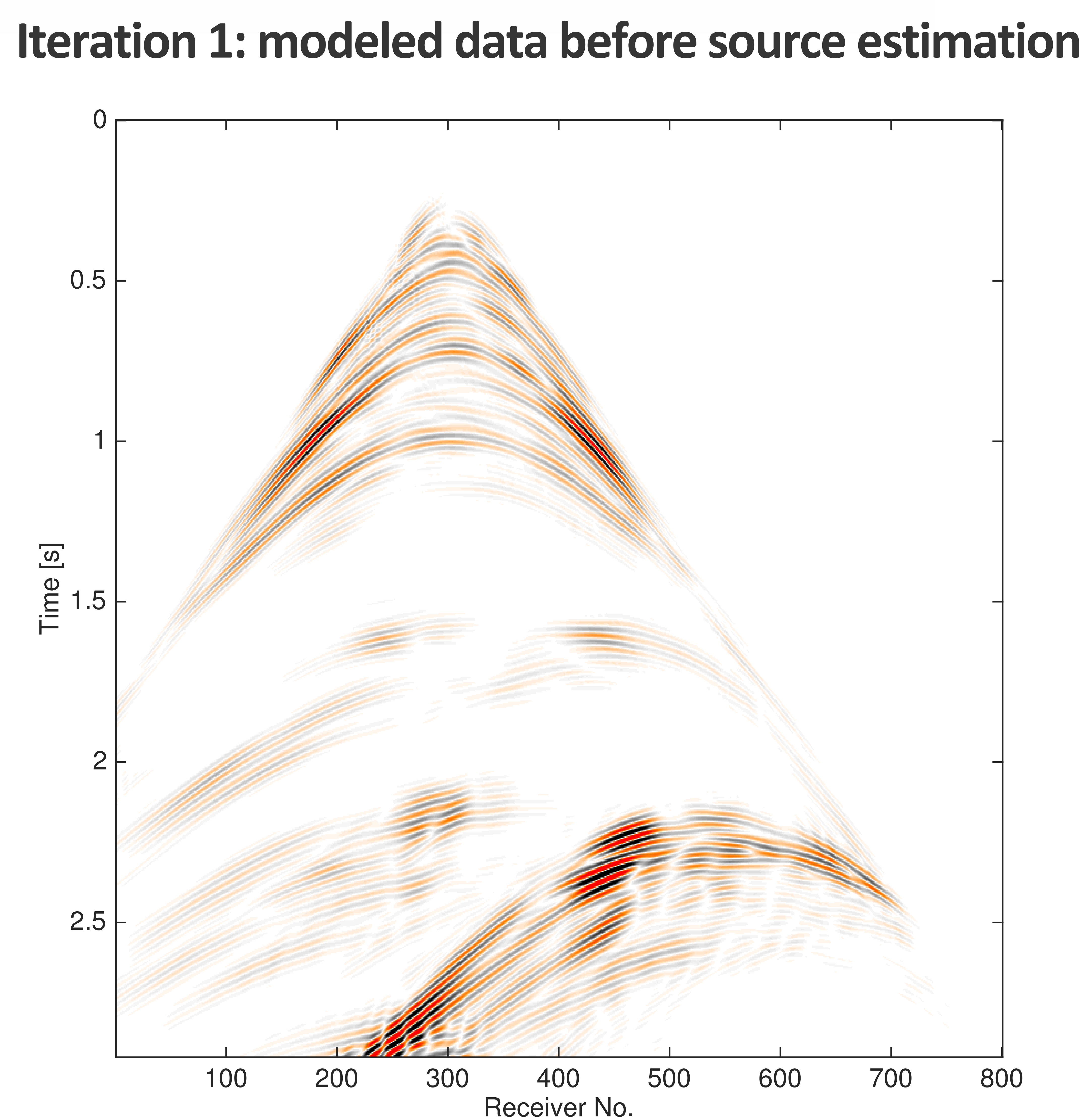


Image updates

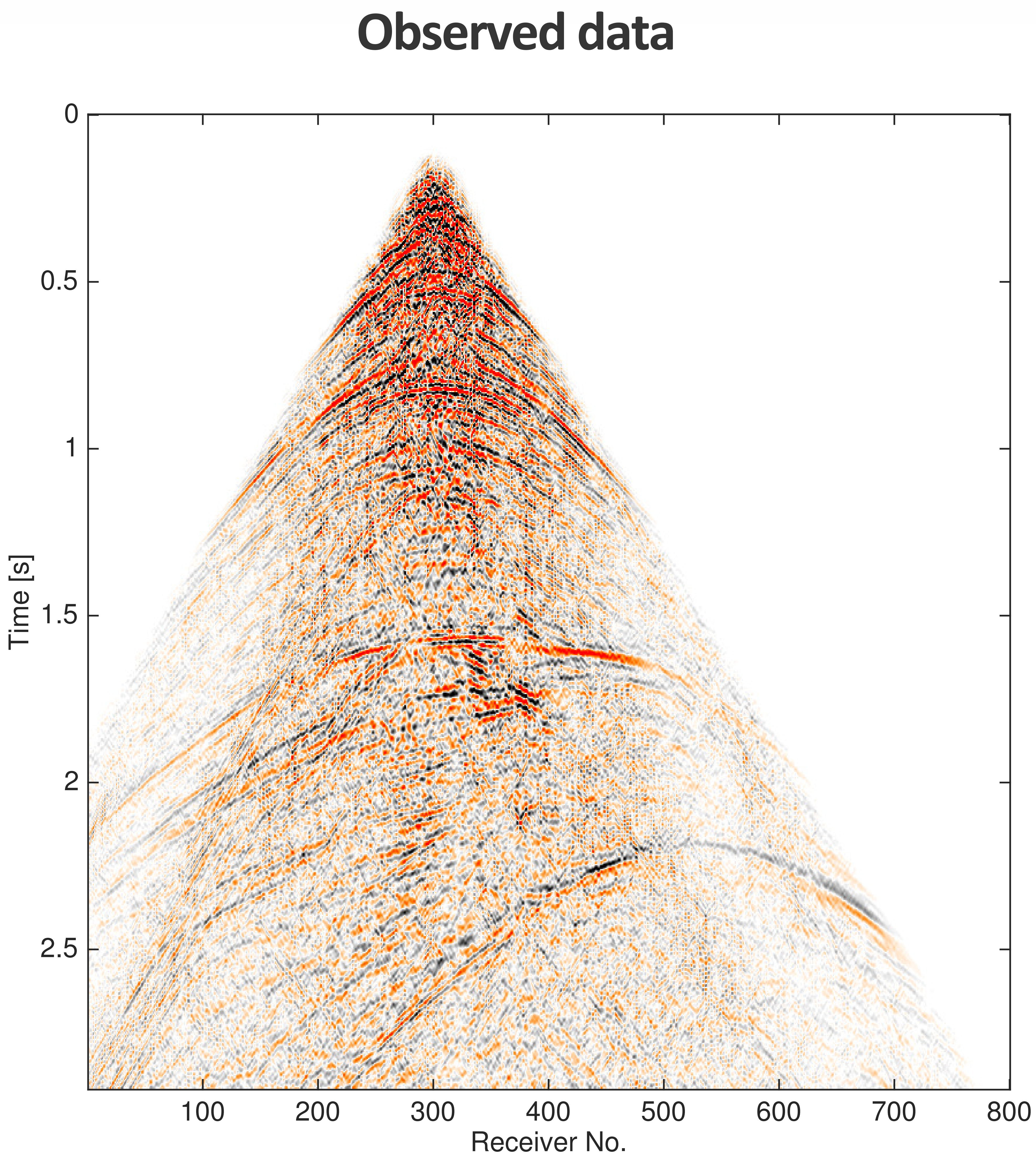


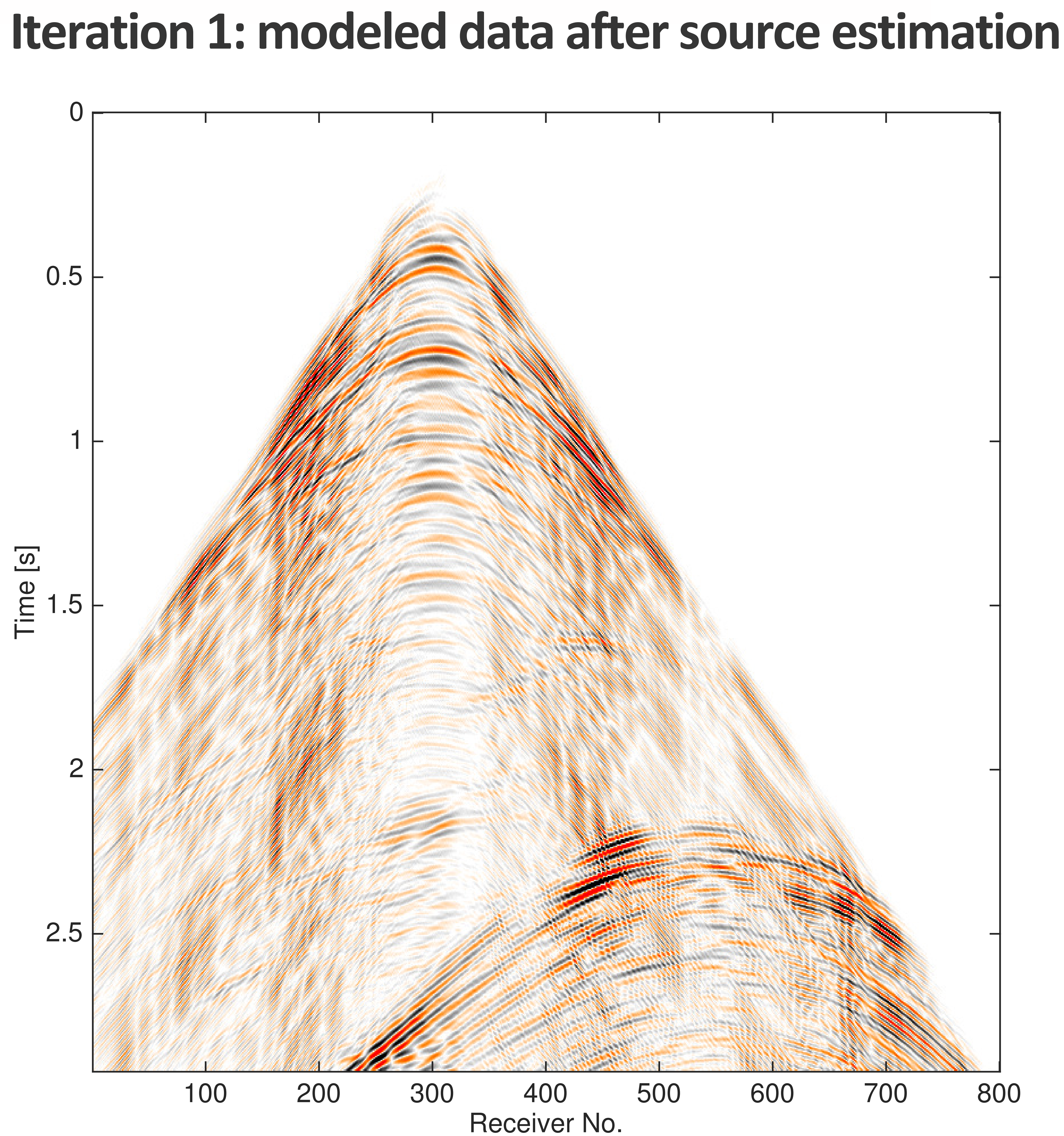
Observed and modeled shot records





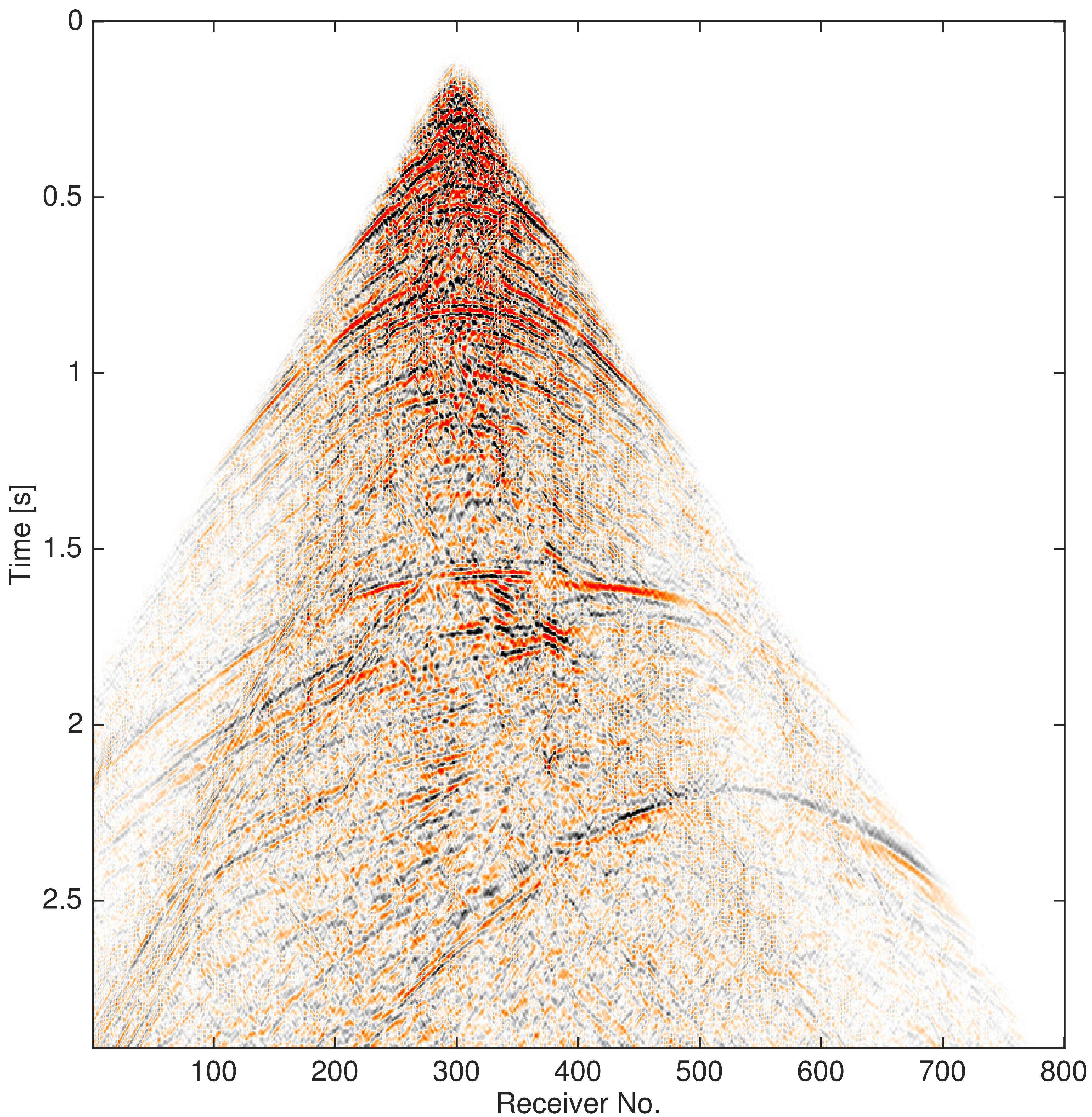
Observed and modeled shot records



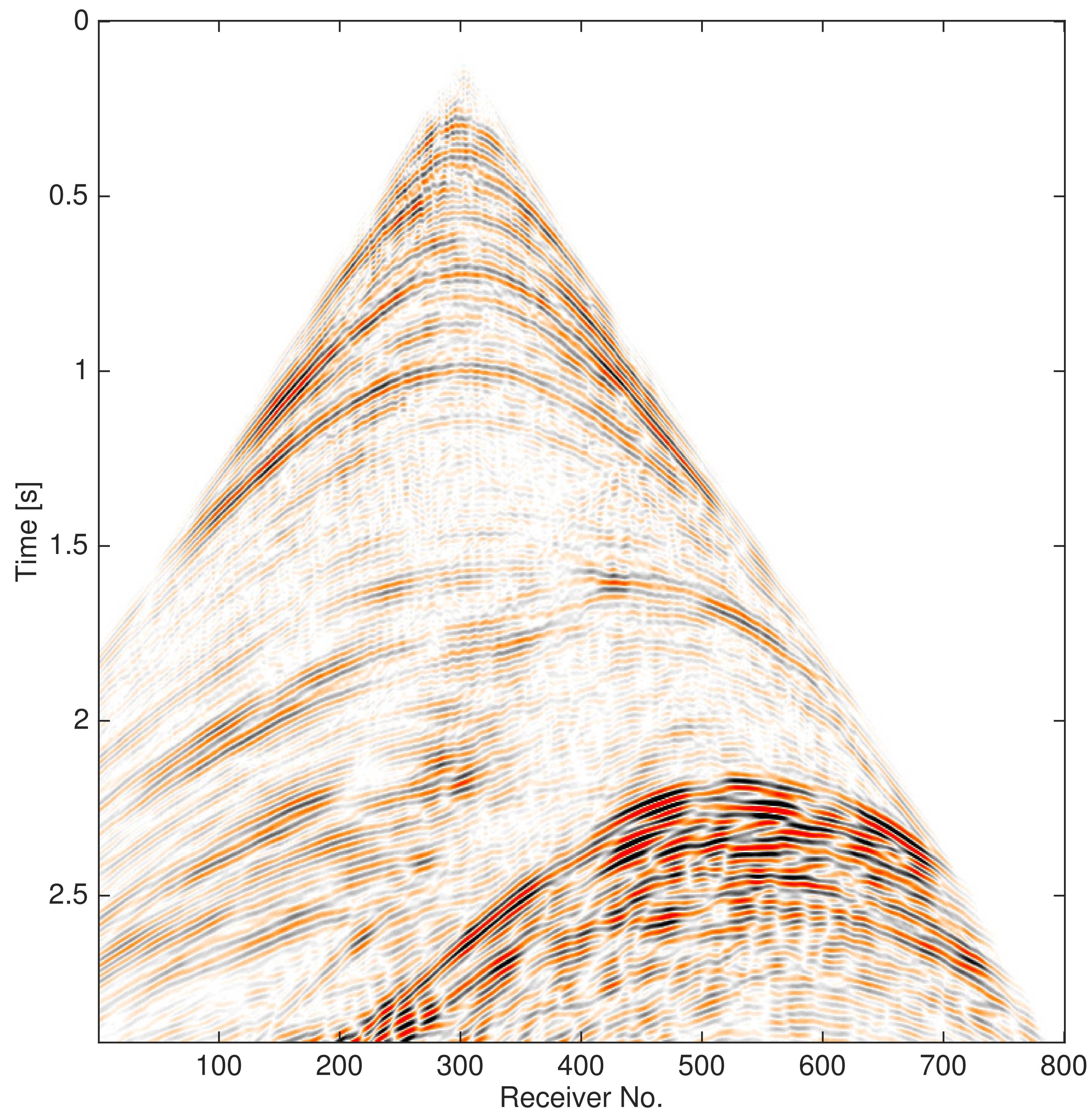


Observed and modeled shot records

Observed data

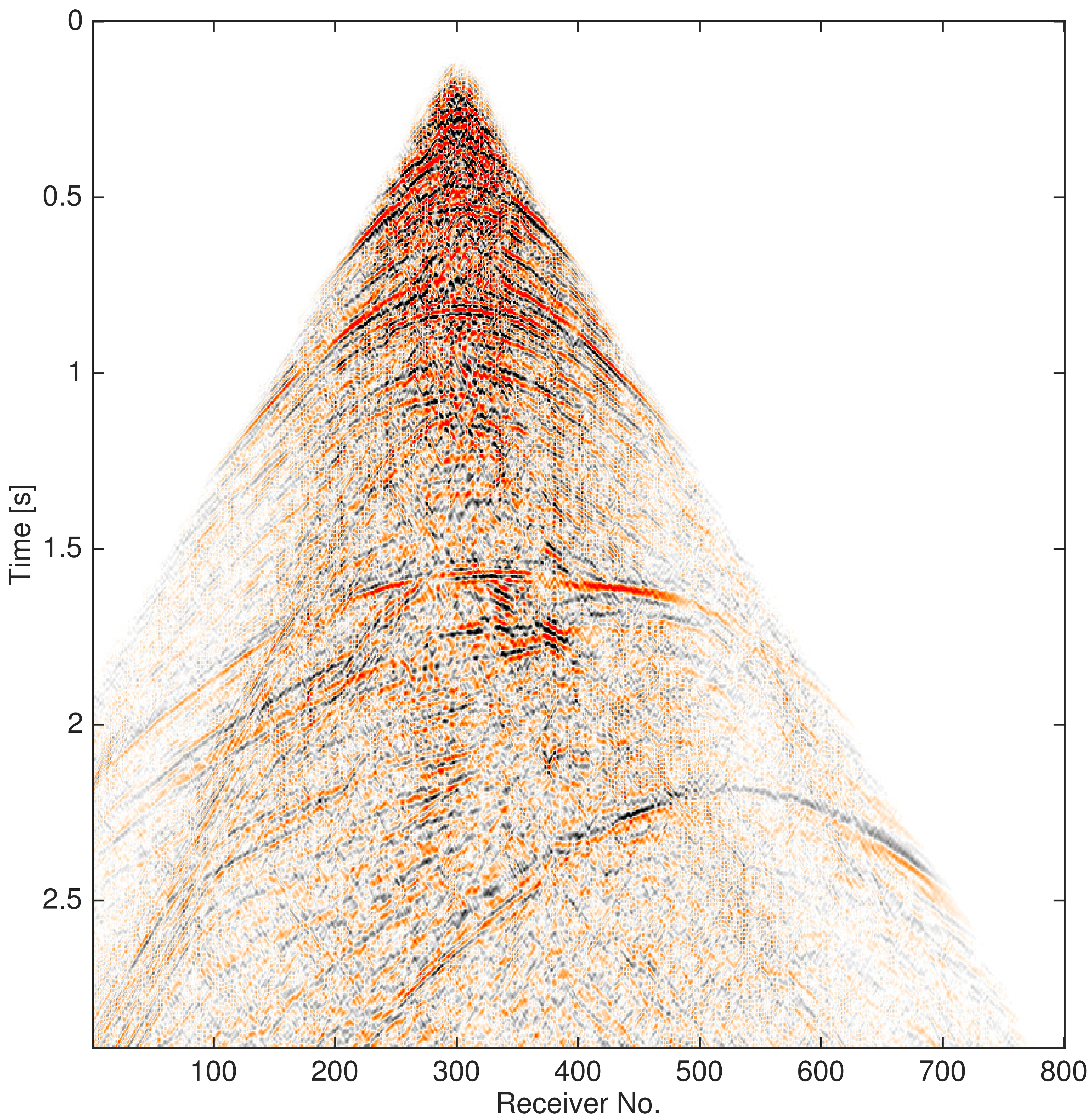


Iteration 10: modeled data before source estimation

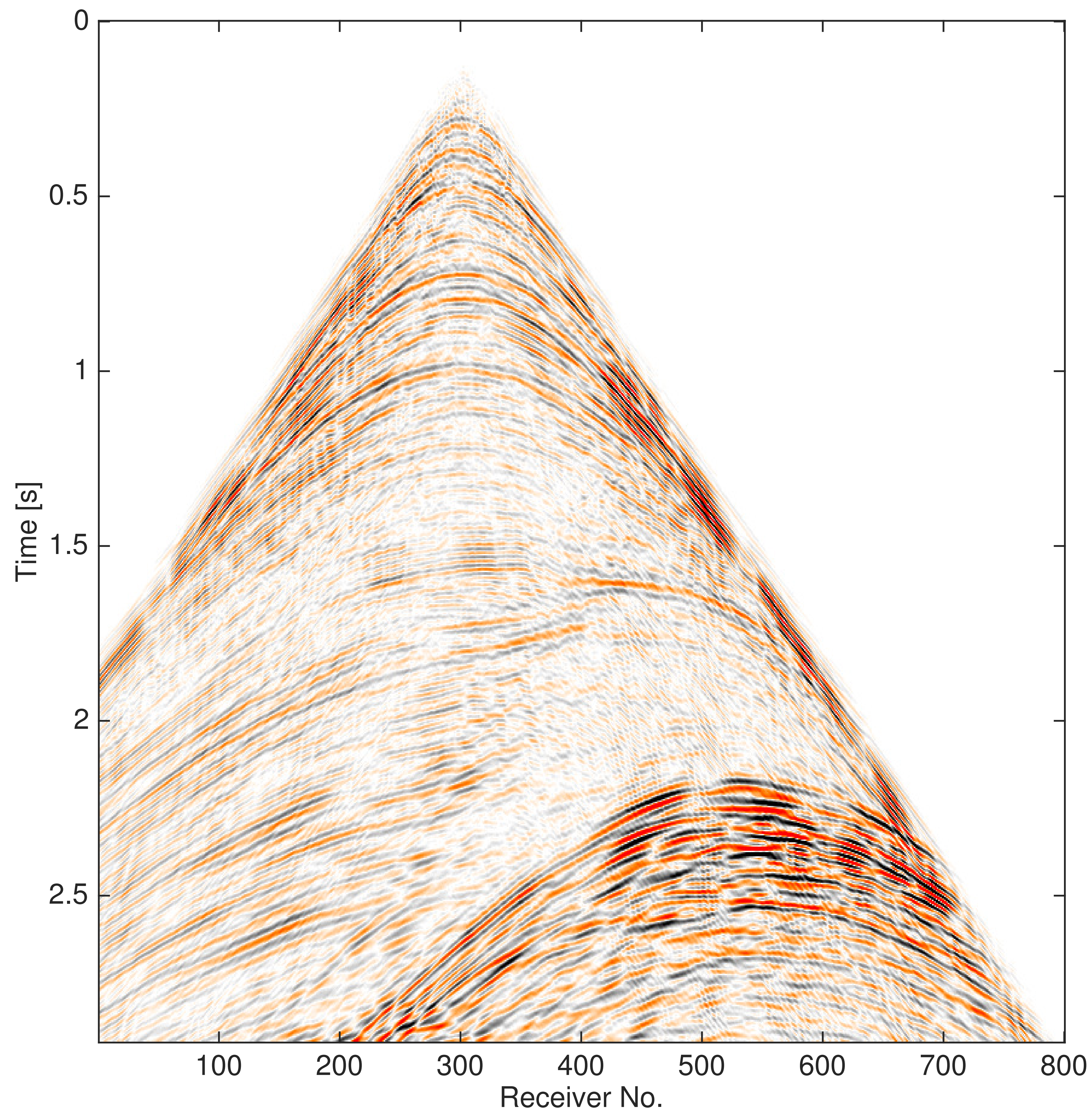


Observed and modeled shot records

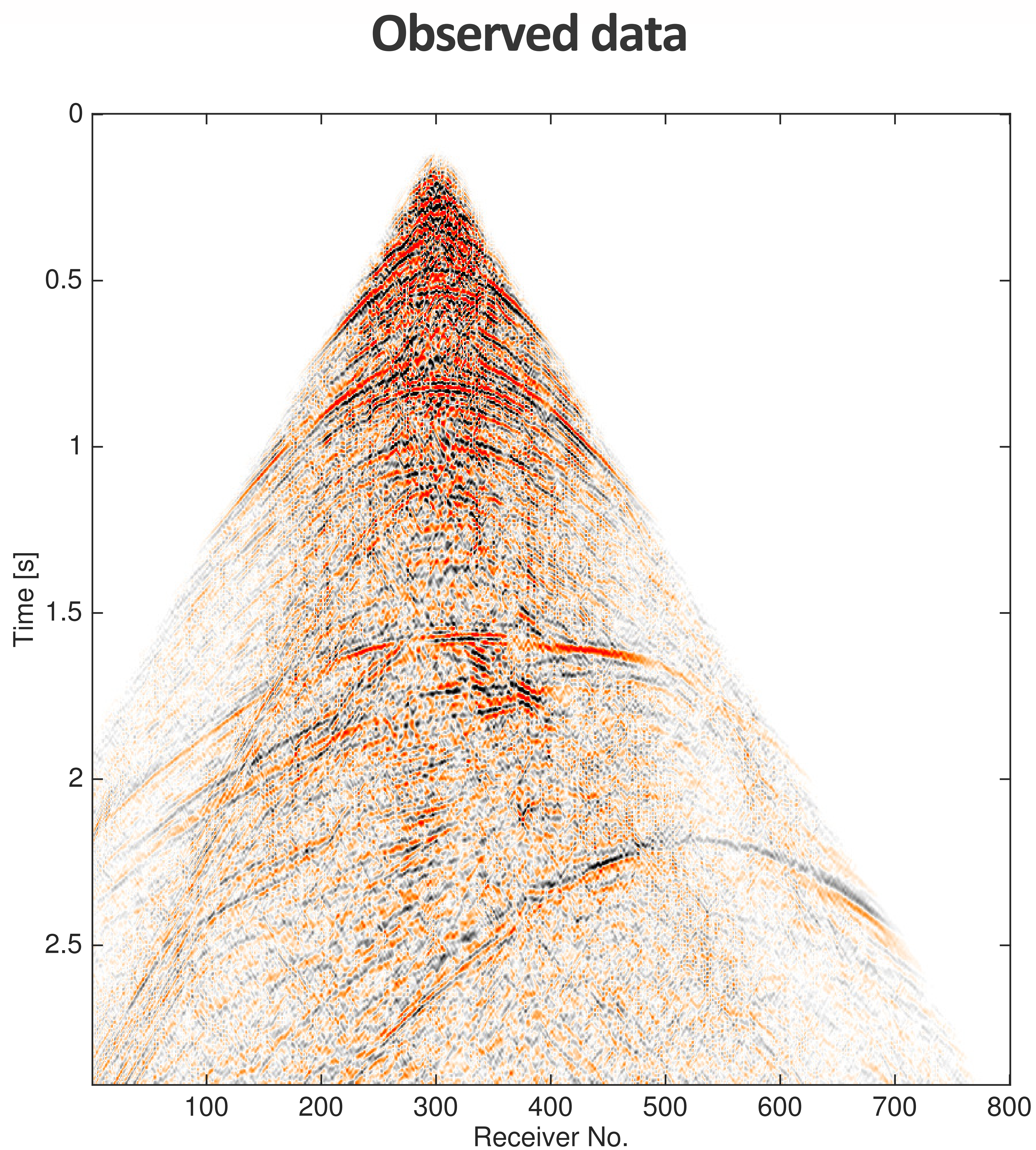
Observed data



Iteration 10: modeled data after source estimation

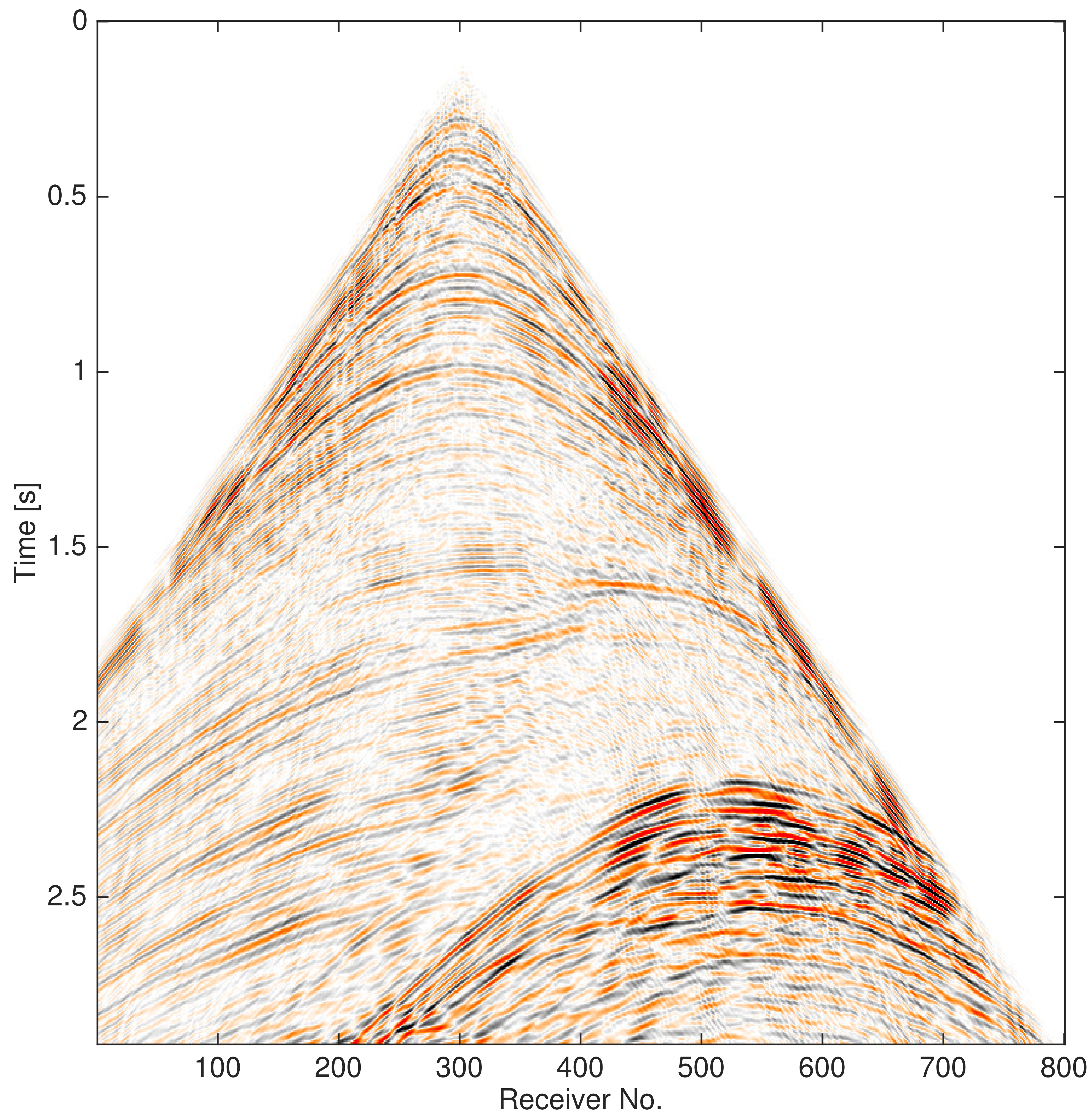


Observed and modeled shot records

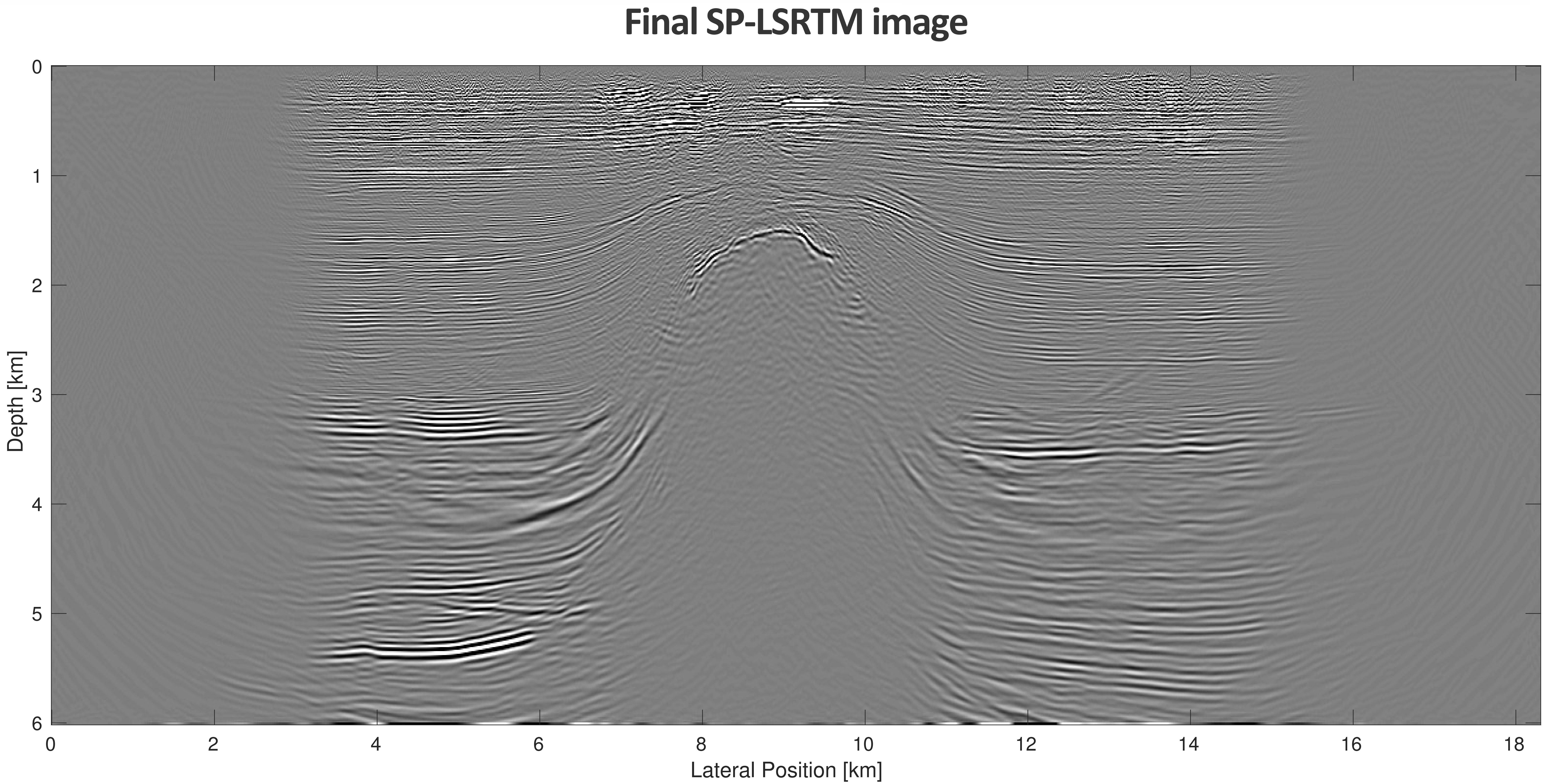


Observed and modeled shot records

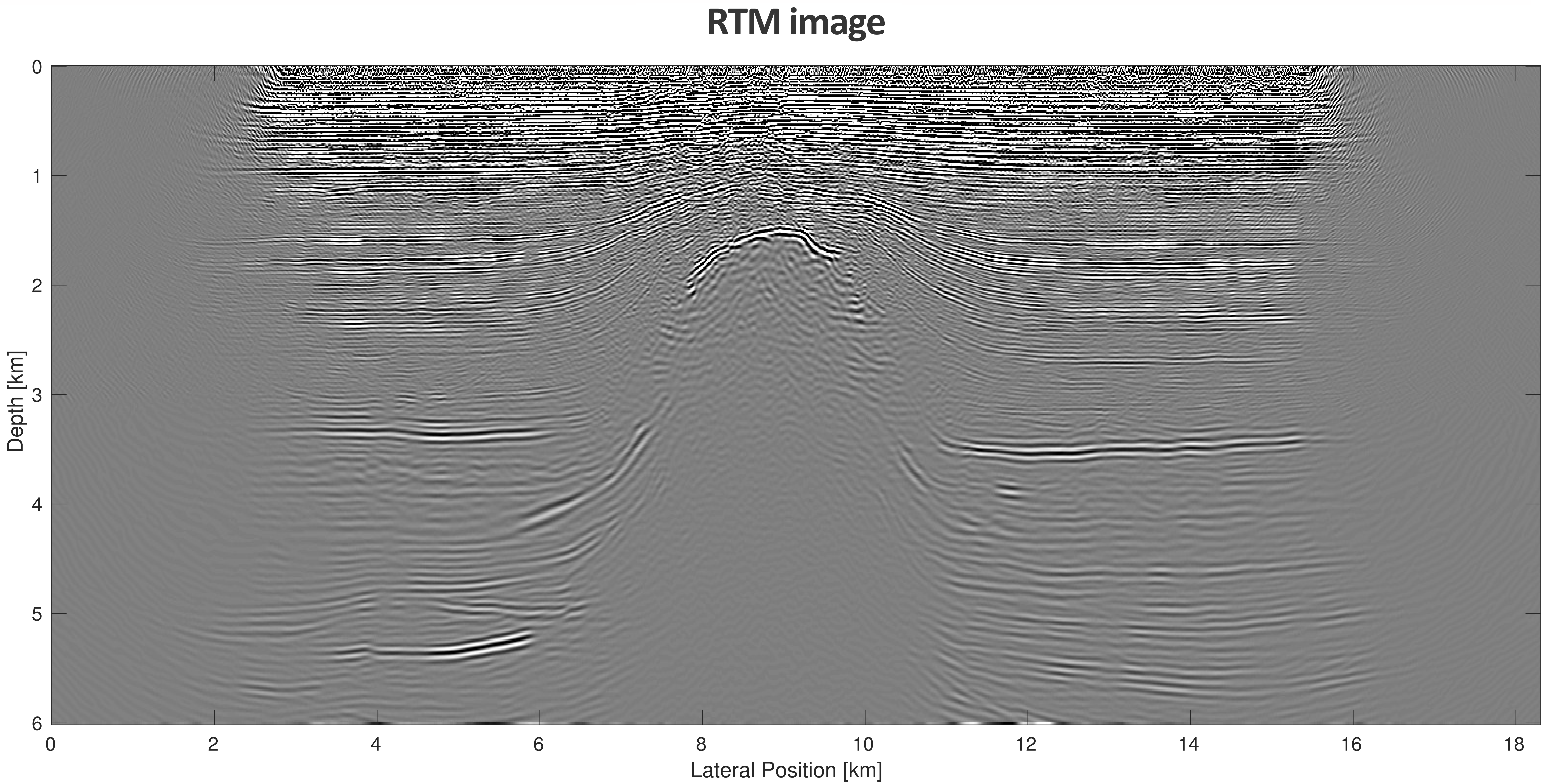
Iteration 10: modeled data after source estimation



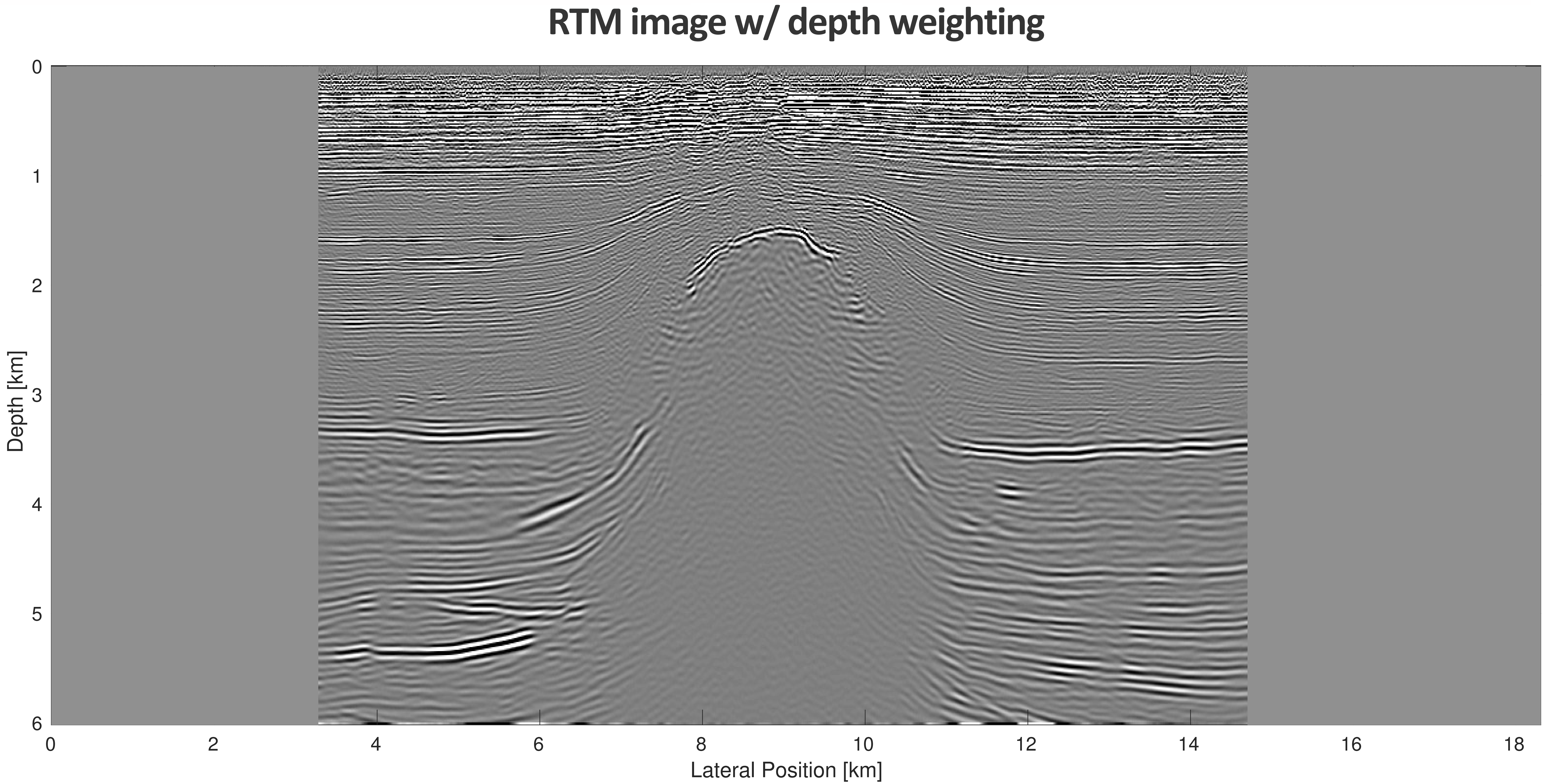
Machar results



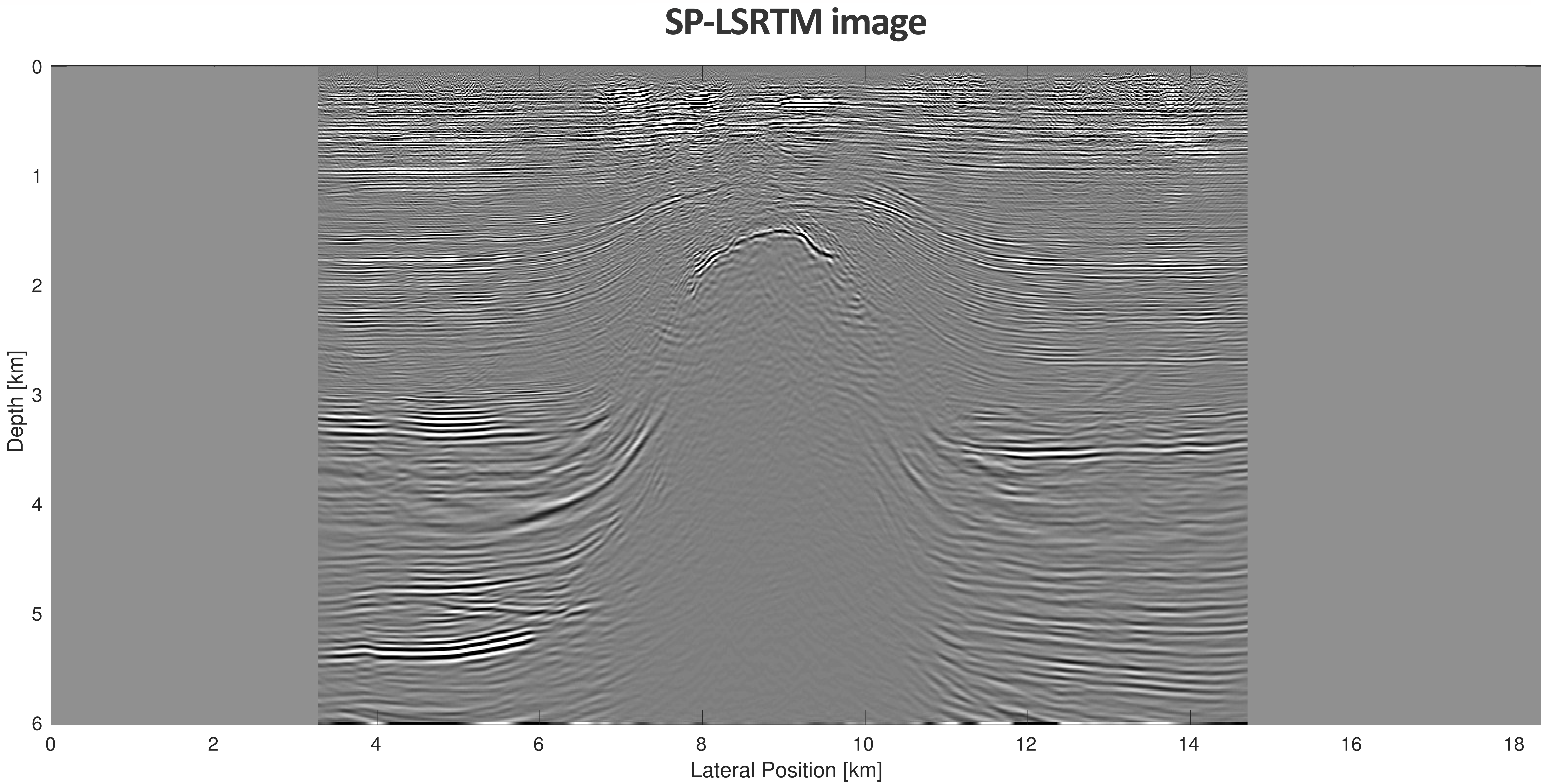
Machar results



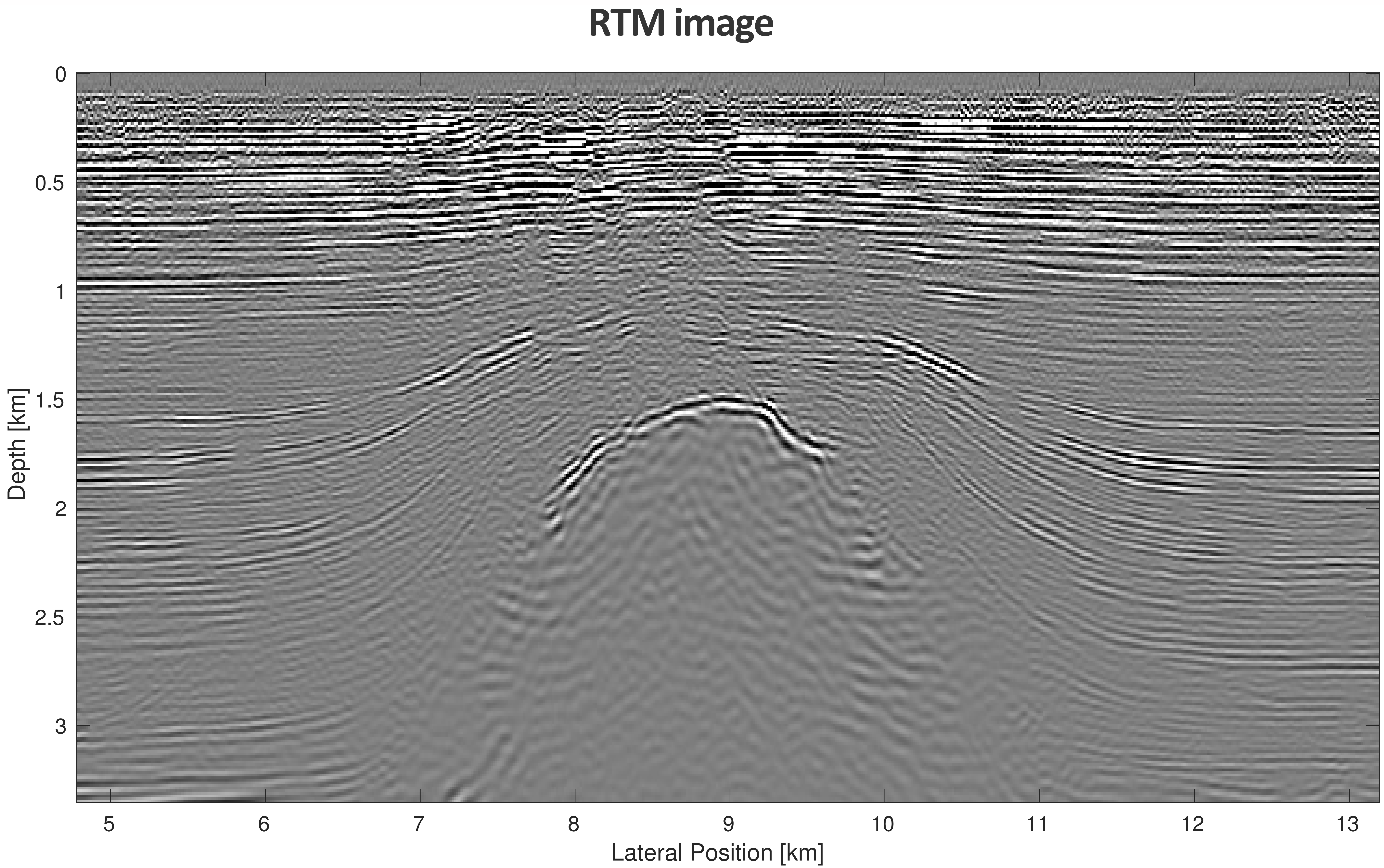
Machar results



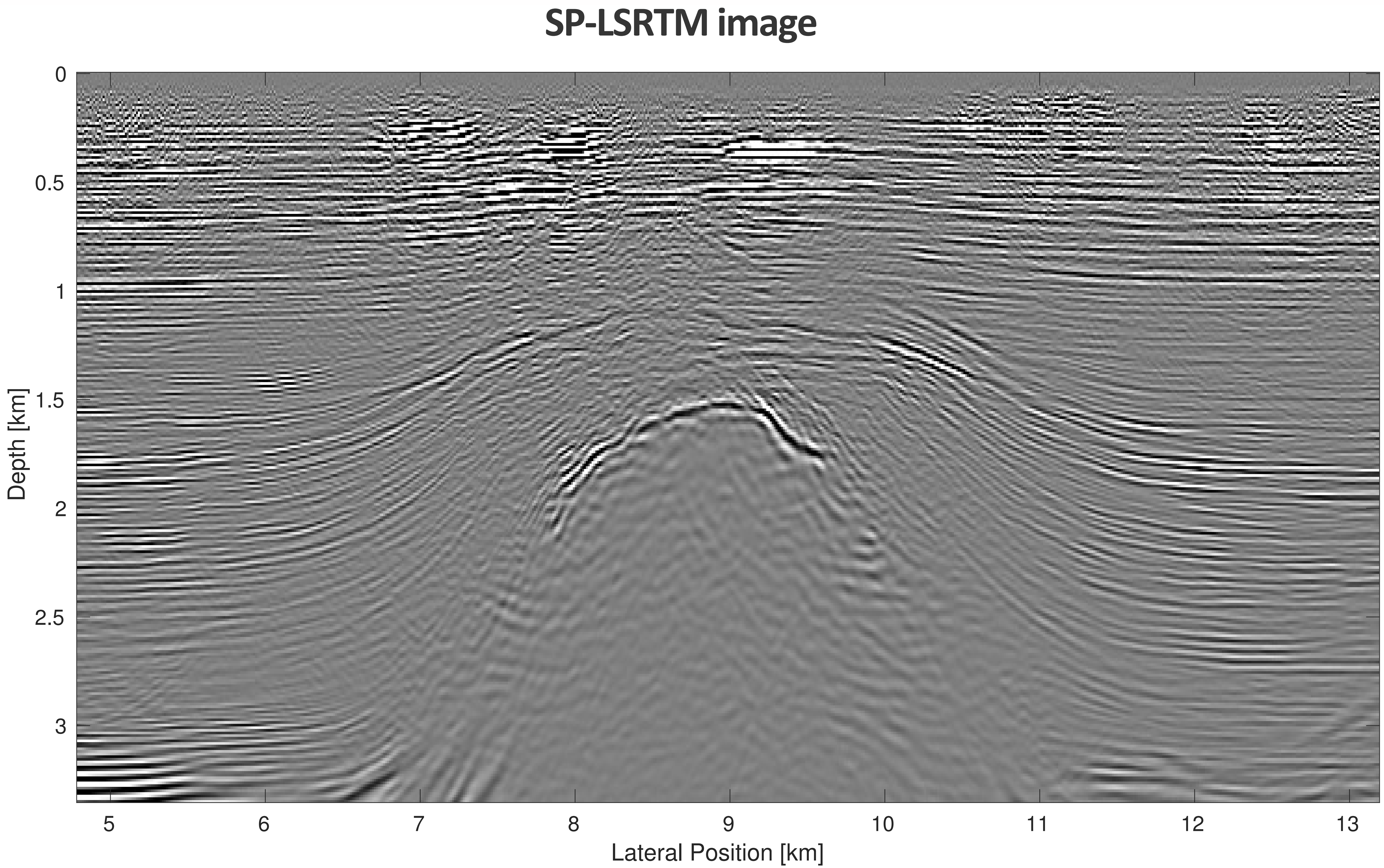
Machar results



Machar results



Machar results



Summary

Introduction of our time-domain LSRTM workflow w/ sparsity promotion:

- Matlab implementation with (pseudo-)acoustic wave equation
- linearized Bregman method as solver
- various pre-conditioners to improve convergence
- source estimation

Influence of correct gradients for LSRTM:

- incorrect gradients + non-linearized data worsen convergence
- still significantly better convergence w/ correct gradient

Outlook

Future projects:

- Application to a sparse data set w/ aliasing
- 3D imaging with our new Julia/Devito workflow

Acknowledgements

This research was carried out as part of the SINBAD project with the support of the member organizations of the SINBAD Consortium.

SINBAD

Acknowledgements

The authors wish to acknowledge the SENAI CIMATEC Supercomputing Center for Industrial Innovation, with support from BG Brasil, Shell, and the Brazilian Authority for Oil, Gas and Biofuels (ANP), for the provision and operation of computational facilities and the commitment to invest in Research & Development.

Acknowledgements

Many thanks to BP for providing us the Machar data set and for letting us show the results