Curvelet-based Migration Preconditioning

Advantages of a Diagonal Scaling Curvelet Preconditioner

Cody R. Brown
cbrown@eos.ubc.ca

Felix J. Herrmann
fherrmann@eos.ubc.ca

Peyman Poor Moghaddam
ppoor@eos.ubc.ca

Yogi A. Erlangga
yerlangga@eos.ubc.ca

Seismic Laboratory for Imaging & Modeling (SLIM)
Department of Earth & Ocean Sciences
The University of British Columbia

SEG 2008 Annual Meeting, Las Vegas
Tuesday, November 11th, 2008

Principle of Least Effort

- George Kingsley Zipf's principle states that people and even well designed machines will naturally choose the path of least effort.

- This is the same for us!
- If we can get to the same solution, lets choose the path that requires the least amount of work.

Lazyman Rubik's Cube

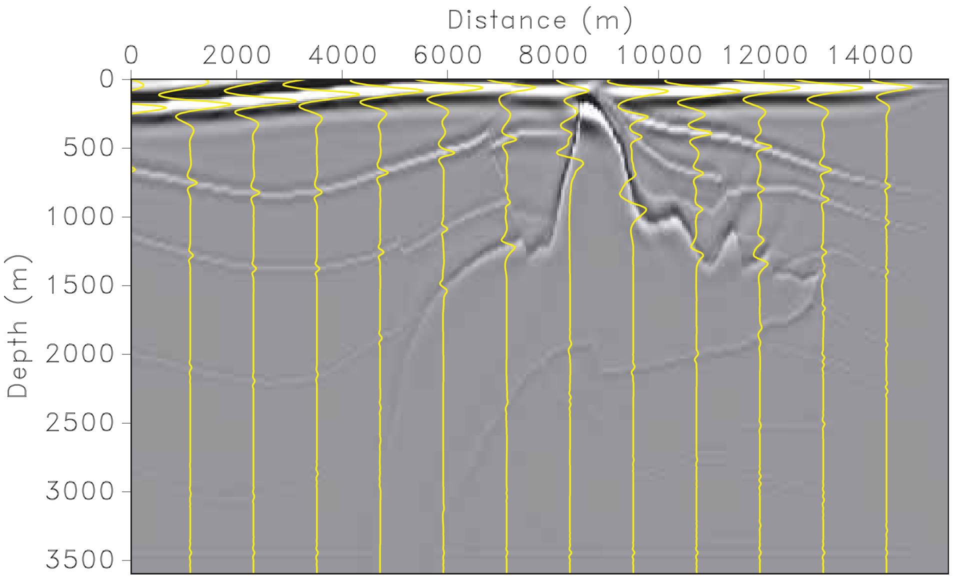
Outline

- Motivation
- Definition of our Problem
- Preconditioning Levels:
  - Level I - Fractional Differentiation
  - Level II - Depth Correction
  - Level III - Curvelet-based Diagonal Estimation
- Some Data Examples
  - Simple Synthetic Reflector w/ Lens Velocity
  - SEG AA' Model w/ Smooth Velocity
- Conclusions

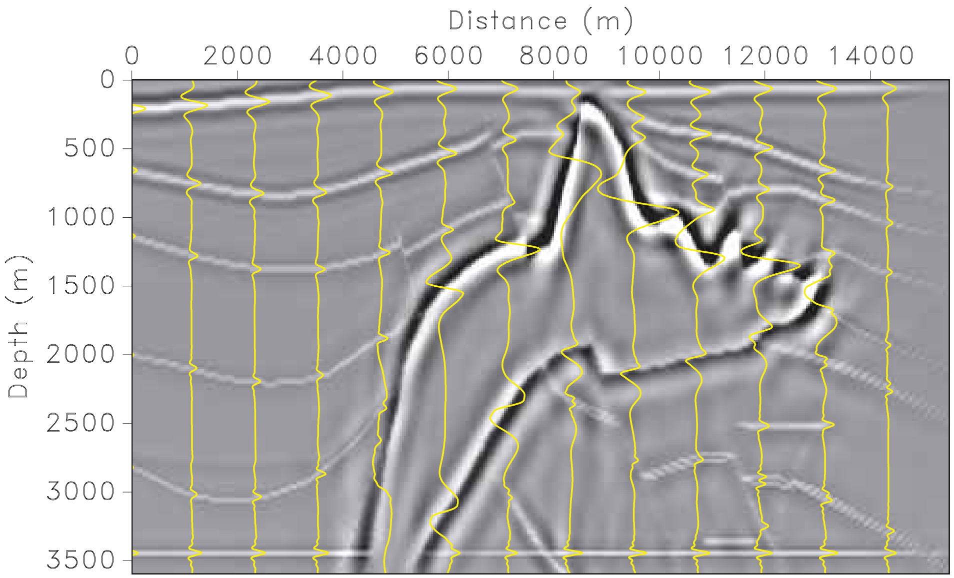
What We Want

- We want to *correct amplitudes* and *regularize reflector* information throughout the image.

- *Stabilize* the problem and *improve convergence* rates.



No Preconditioning



Full Preconditioning

Why do we need a *(pre)*conditioner?

- In the seismic world, we deal with extremely large data-sets.
  – Requires a lot of time to do simple operations.
  – Even more time to apply just one migration!

- Iterative solvers require significant resources and time.
  – We need to reduce the number of iterations.

- We would like to stabilized the problem.
  – Applying small changes will still allow our LSQR algorithm to converge.

- **Principle of Least Effort!**
  – We want to do all these with the least amount of work.

Why do we need a *(pre)*conditioner?

Solution? Preconditioning!

- Preconditioning allows us to increase the convergence of iterative solvers.
  - Reduces the *number of iterations*.
  - Reduces the *overall time* required.
  - And gives us an *improved result*!

- Preconditioners don’t have to be exact.
  - Our examples none of the preconditioners were computed to convergence.
  - Still see *significantly improved amplitudes*.

- **Satisfies Principle of Least Effort!**

Our Problem

- During seismic imaging, the following system of equations needs to be solved:

$$\mathbf{A}\mathbf{x} \approx \mathbf{b}$$

- Inverting this equation we get:

$$\widetilde{\mathbf{x}}_{LS} = \left(\mathbf{A}^*\mathbf{A}\right)^{-1}\mathbf{A}^*\mathbf{b} := \mathbf{A}^\dagger\mathbf{b}$$

- This involves the inversion of the normal equations.
  - With large data, these become quite difficult to compute efficiently.

- **Inverting this is not so trivial and we will need to use iterative matrix-free methods such as LSQR.**

[Symes, 2008]
[Rickett, 2003]
[De Roeck, 2002]
[Clearbout and Nichols, 1994]

Our Problem

- Inverting this is not so trivial because of the size:

$$\widetilde{\mathbf{x}}_{LS} = \arg\min_{\mathbf{x}} \frac{1}{2}\|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2^2$$

- We want to **condition** this as well as possible.

- With accurate background velocity this iterative solution is known to converge quickly.
  - The sheer size of the problem however makes this a very time consuming problem.

- *A reduction in the number of iterations will be necessary!*

[Paige and Saunders, 1982]

Our Solution

- We propose to do this by replacing our initial system with a *series of preconditioning* levels:

$$\mathbf{M}_L^{-1}\mathbf{A}\mathbf{M}_R^{-1}\mathbf{u} \approx \mathbf{M}_L^{-1}\mathbf{b}, \qquad \mathbf{x} := \mathbf{M}_R^{-1}\mathbf{u}$$

- This involves a series of *right* and *left* preconditioning matrices.

- These preconditioning matrices all compound together and produce a **solid reduction of residual errors** per iteration.

- The cost for applying these preconditioners is just a matrix multiplication in the respected domain.

Our Solution

- Our preconditioners are derived from the following three observations:

  - the normal operator is in d dimensions a (d-1)-order pseudo-differential operator

  - migration amplitudes decay with depth due to spherical spreading of seismic body waves

  - zero-order pseudo-differential operators can be approximated by a diagonal scaling in the curvelet domain

[Symes, 2008]

[Herrmann et al., 2008]

Levels of Preconditioning

- We propose three levels of preconditioning:

- Level I - Scaling in the *Fourier* domain.
  - Fractional differentiation.
  - Approximate a (d-1)-order pseudo-differential operator.
  - Improve low-frequency components.

- Level II - Scaling in the *physical* domain.
  - Depth correction.
  - Corrects for amplitude decay of the migration code.

- Level III - Scaling in the *curvelet* domain.
  - Curvelet-based diagonal estimation.
  - Restores amplitudes throughout the image.

Levels of Preconditioning

L1 L2 L3

- In data space we apply a multiplication in the temporal Fourier domain.

- This can be thought as a *left* preconditioning through fractional differentiation:

$$\mathbf{M}_L^{-1} := \partial_{|t|}^{-1/2}$$

- Some low-frequency content is restored.

- Sets up the curvelet-based diagonal estimation by approximating a (d-1)-order pseudo-differential operator.

Levels of Preconditioning

L1 L2 L3

- *Right* preconditioning by scaling in the physical domain:

$$\mathbf{M}_R^{-1} = \mathbf{D}_z := \mathrm{diag}\left(\mathbf{z}\right)^{\frac{1}{2}}$$

- Reflected waves travel from the source at the surface down to the reflector and back.
- This gives a quadratic depth dependence.

- Everything is compounded together. This can be removed if desired.

Levels of Preconditioning

L1 L2 L3

- *Right* preconditioning by scaling in the curvelet domain:

$$\Psi \mathbf{r} \approx \mathbf{C}^* \mathbf{D}_\Psi^2 \mathbf{C} \mathbf{r}, \quad \mathbf{D}_\Psi^2 := \mathrm{diag}\left(\mathbf{d}^2\right)$$

$$\mathbf{M}_R^{-1} = \mathbf{D}_z \mathbf{C}^* \mathbf{D}_\Psi^{-1}$$

- Estimation of the diagonal in the curvelet domain.

- The cost to compute this diagonal is *one migration and one remigration*.
  - This is equivalent to one iteration of LSQR.

- **Improves amplitudes throughout the image.**

[Herrmann et al., 2008]

Levels of Preconditioning

L1 L2 L3

CONSTRUCTING THE CURVELET DIAGONAL.

- We require a migrated and re-migrated image.
- We use one lambda parameter to control smoothing.

- We then solve the system with a limited memory Quasi-Newton method: L-BFGS.
  - No need to solve to convergence, approximating the diagonal is good enough.
  - Can see a rough approximation already improves imaged results.

- More information about this process can be found in the references.

[Herrmann et al., 2008]

Levels of Preconditioning

L1 L2 L3

WHY CURVELETS?

- Well-documented approximate invariance of curvelets under the linearized Born-scattering operator.
  - Consequently the columns of the preconditioned system are curvelet like.
  - For instance, small shifts over the support of a curvelet will not adversely affect the corresponding curvelet coefficient.
- Redundancy of the curvelets.
  - Makes this transform less prone to errors in individual entries in the curvelet vector.
- Redundancy spreads coherent noise over more coefficients.
  - A small subset of localized curvelets contribute to a particular feature. Thus only a small fraction of the 'noise' will contribute to the reconstruction.

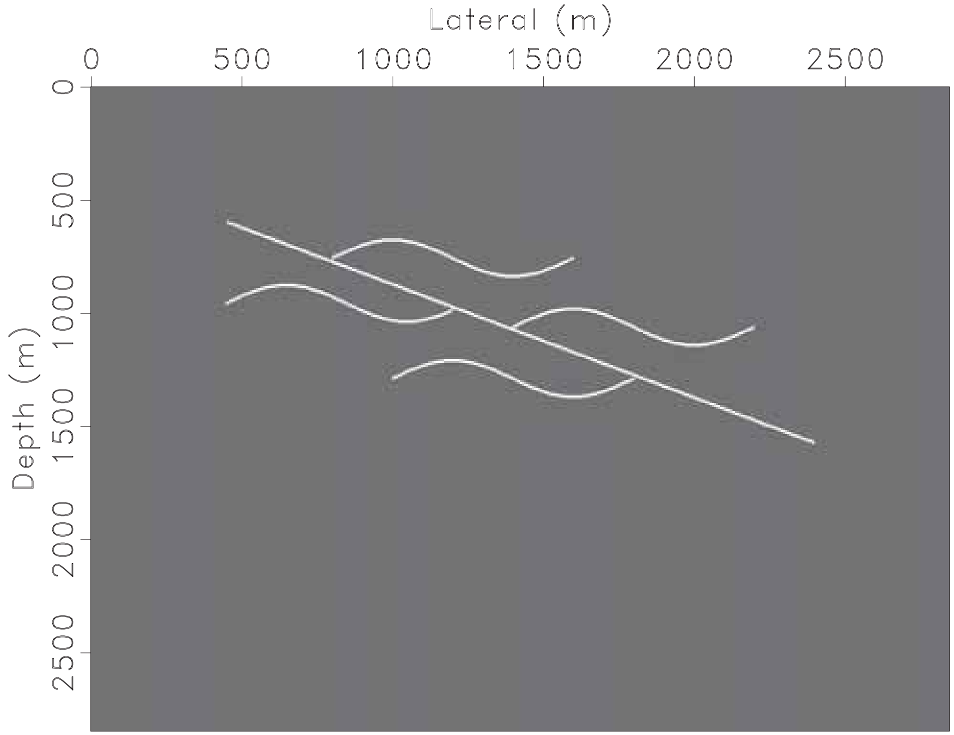
[Candès and Demanet, 2005]
[Douma and de Hoop, 2007]
[Chauris and Nguyen. 2008]
[Anderson et al., 2008]

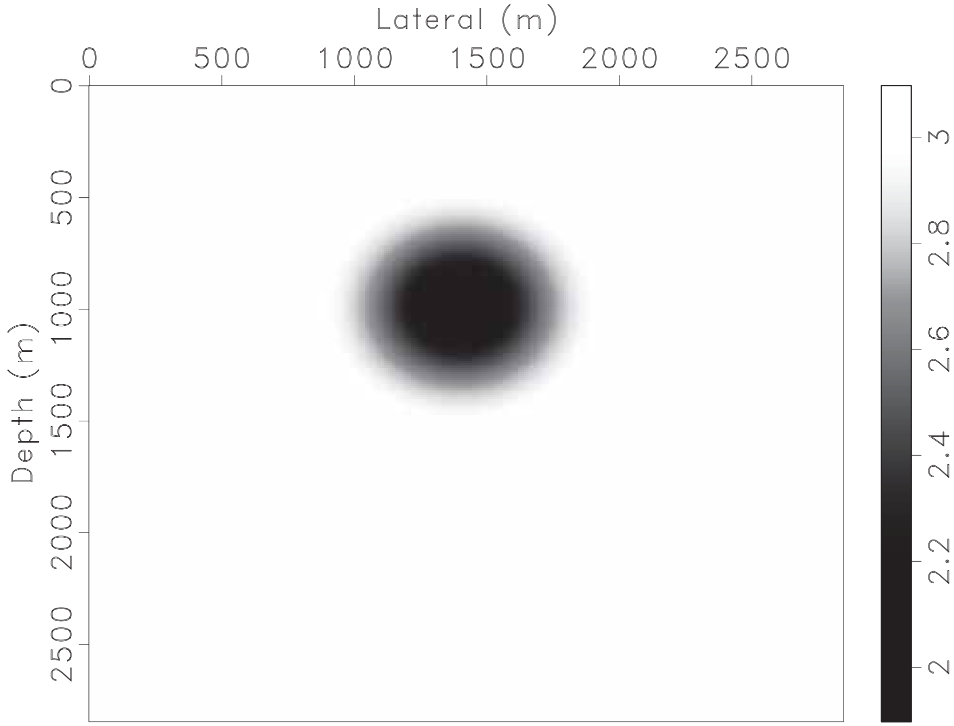
Simple Synthetic Reflector w/ Lens Velocity

- We will look at a simple three reflector w/ fault model.
- Our hope is to *correct amplitudes* in the model.
  - Each preconditioning level should improve amplitudes further.
- We also want to *increase residual decay* per iteration for our iterative method.

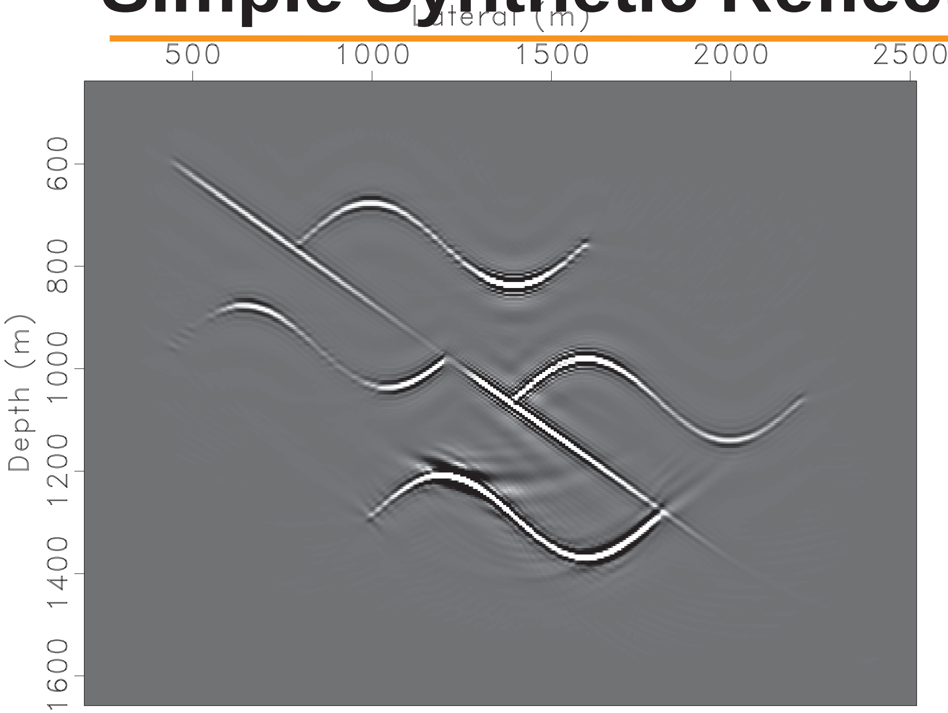
Simple Synthetic Reflector w/ Lens Velocity

- Simple reflector w/ fault reflectivity.
- Low velocity lens model.
- 40 shots.
- We use the linearized Born-scattering forward modeling operator to produce the data.

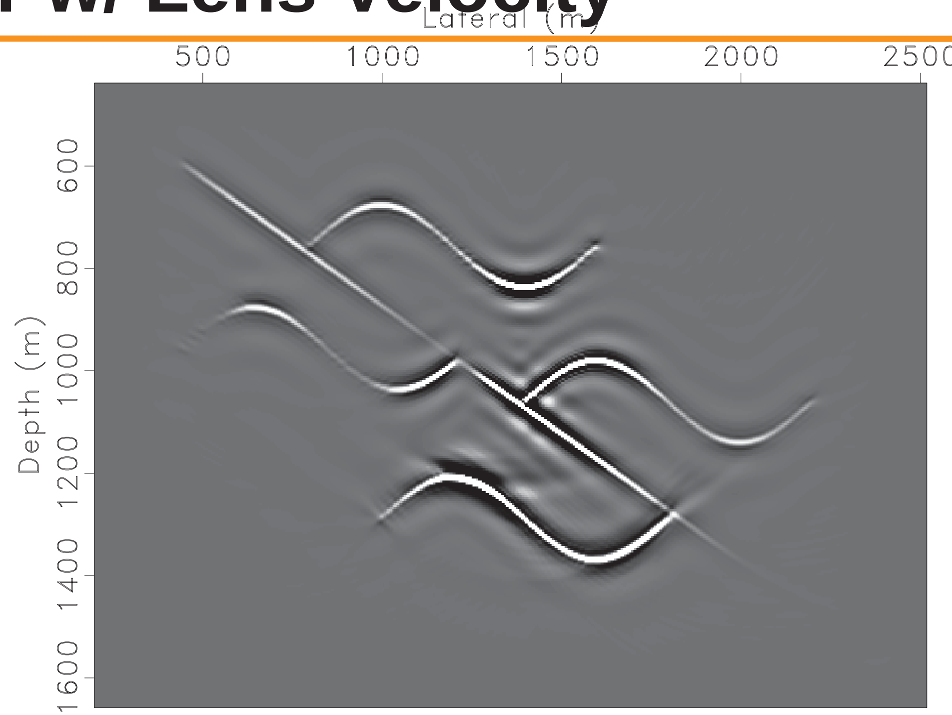




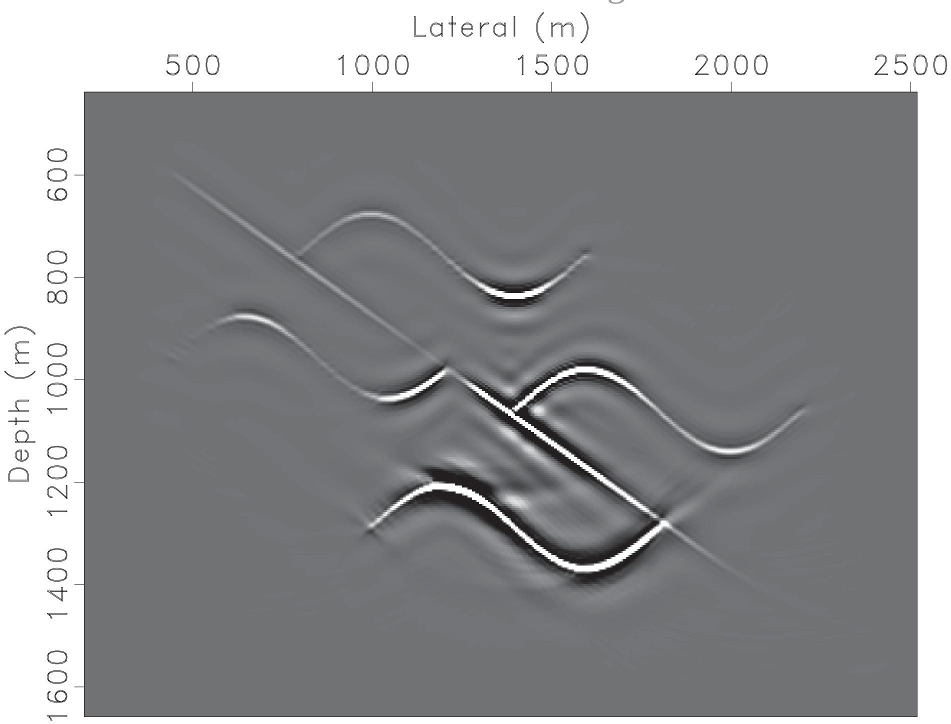
Simple Synthetic Reflector w/ Lens Velocity



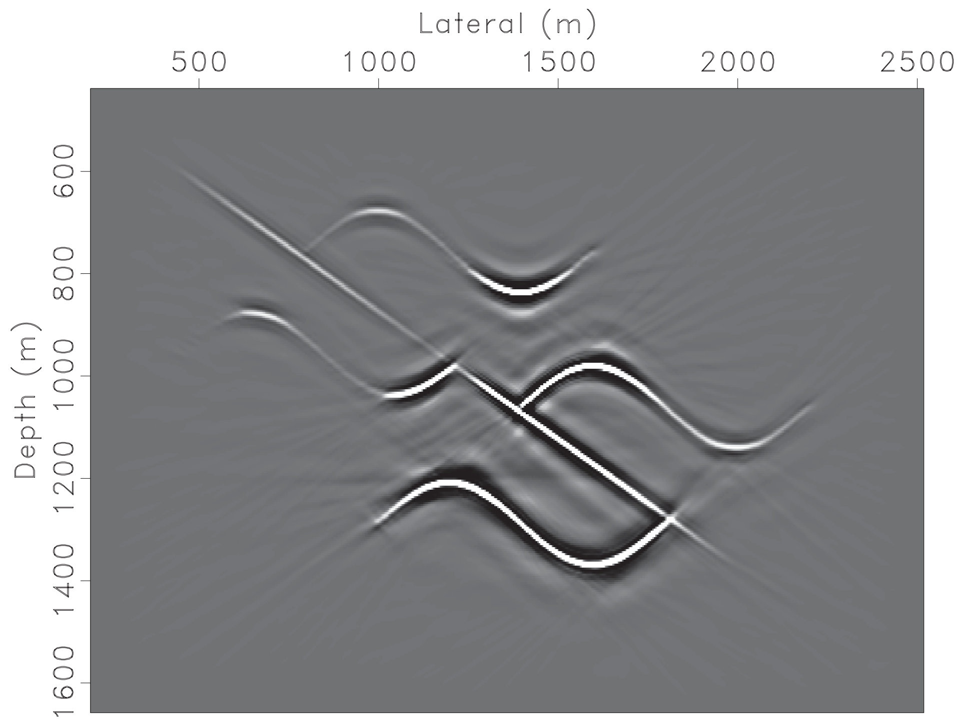
No Preconditioning



Level I

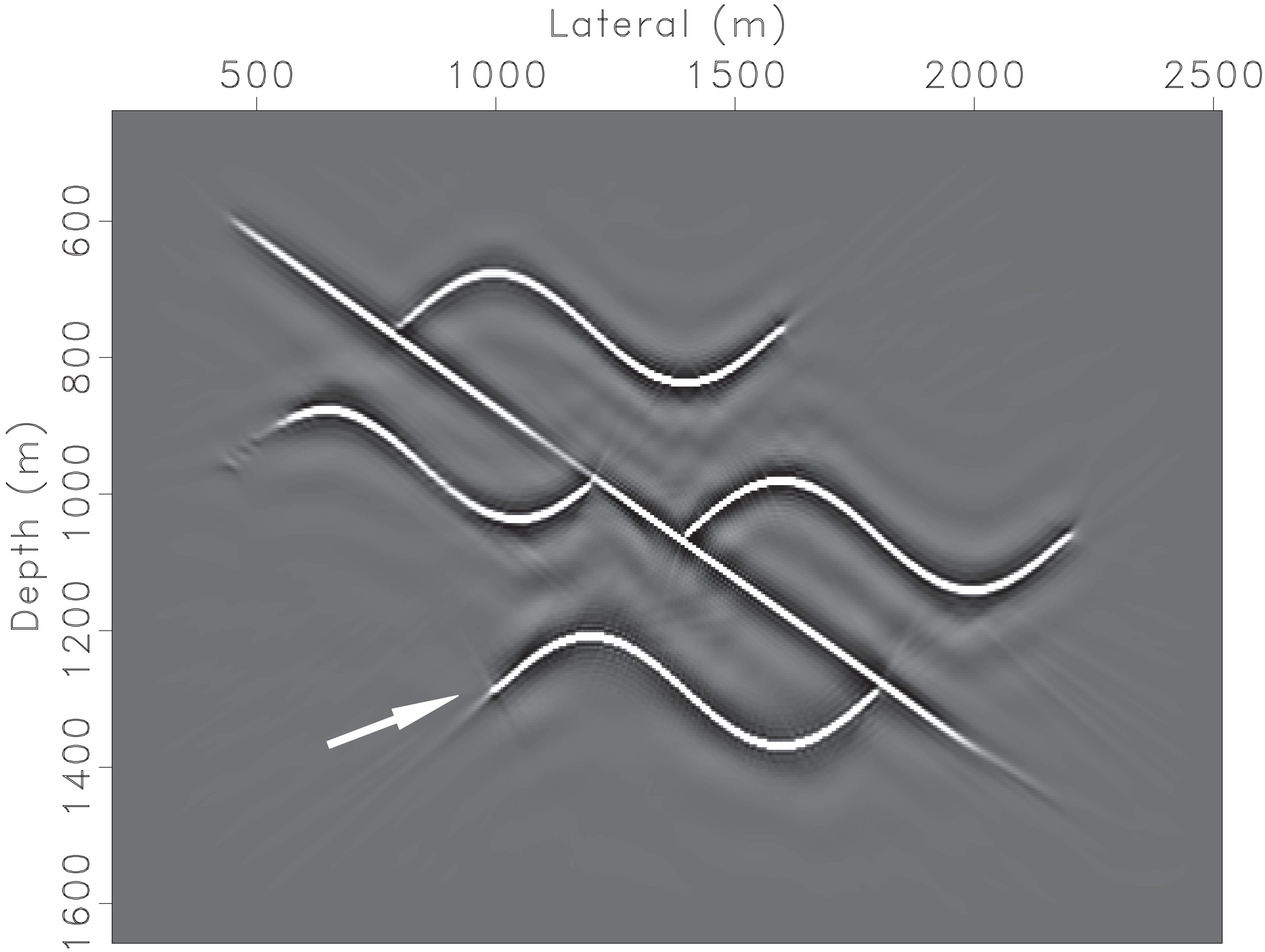


Level II



Level III

Simple Synthetic Reflector w/ Lens Velocity



LSQR Result w/ Level III Preconditioning

Simple Synthetic Reflector w/ Lens Velocity

- Signal-to-Noise Ratio (SNR) to original reflectivity, after one iteration.
- Defined as follows, with L2 values normalized to one:

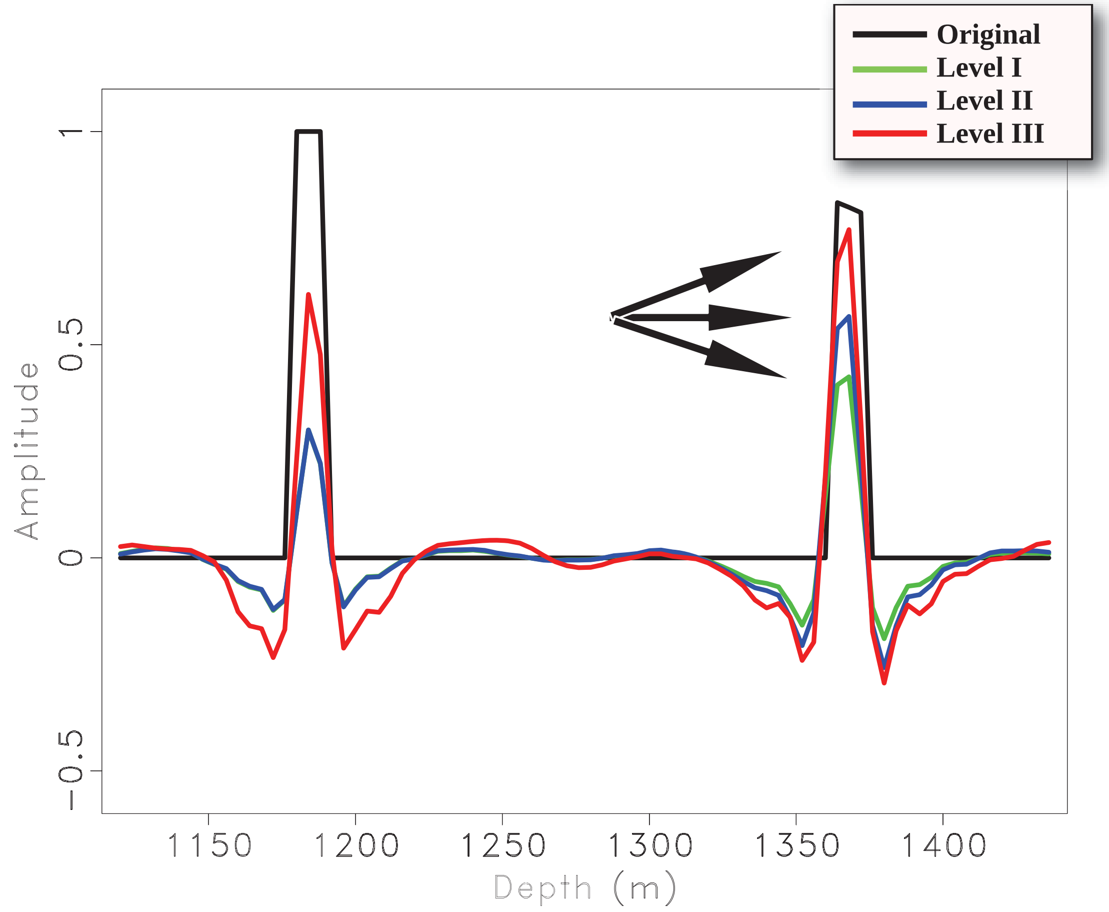
$$SNR = 20 \log \|\mathbf{x}_s\|_2 / \|\mathbf{x}_n - \mathbf{x}_s\|_2$$

	One iteration SNR
No Preconditioning	0.9414
Level I	1.2779
Level II	1.0652
Level III	1.7166

Simple Synthetic Reflector w/ Lens Velocity

- Vertical trace at 1424m.
- Each preconditioning level is restoring the amplitudes closer to the original black line.
- Level III (curvelet-based diagonal) is doing the most significant amplitude recovery in this case.

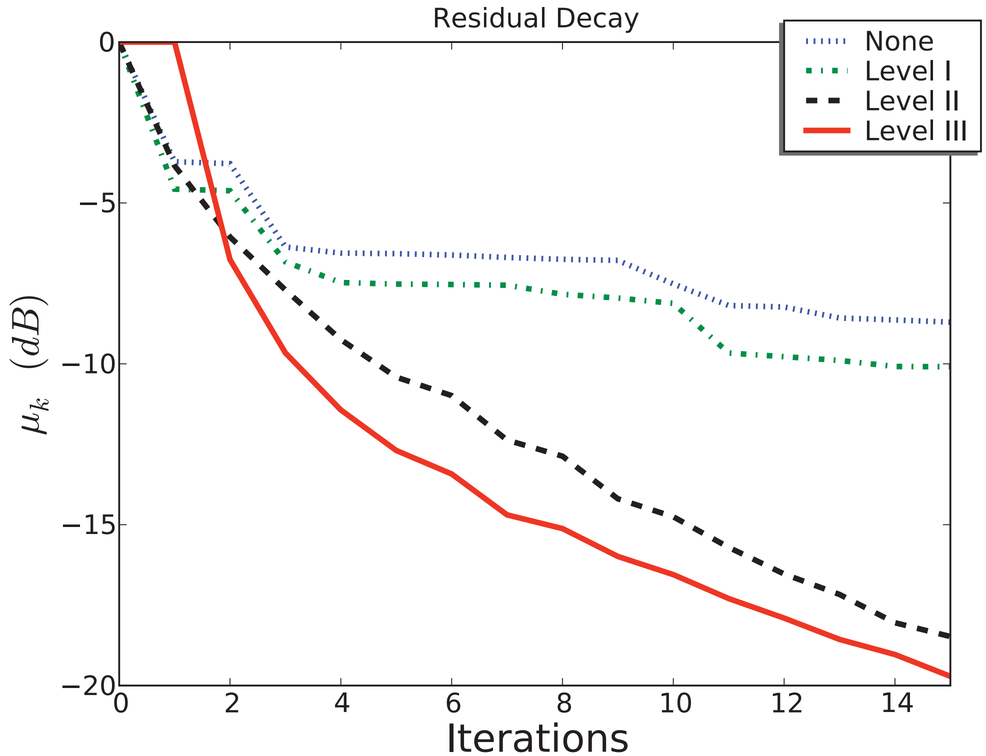
Vertical trace near the center of the model.



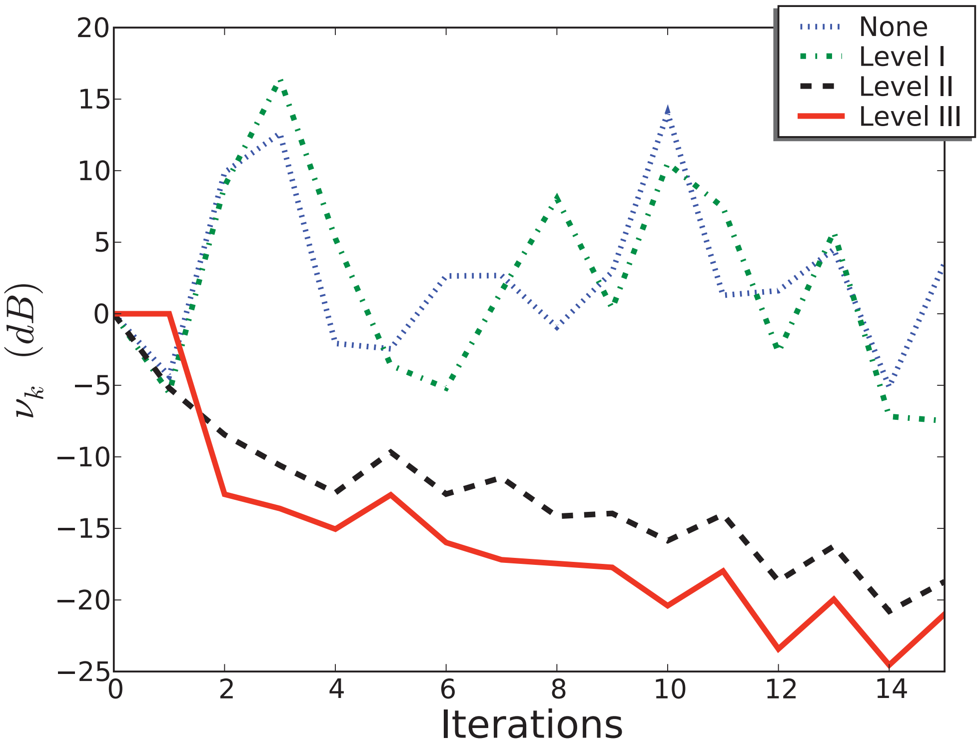
Simple Synthetic Reflector w/ Lens Velocity

- Residual decay for the data-space and model-space residuals.
- Even after our first iteration of level III preconditioning, we are always below the other cases in each figure.

- The red line has already seen one migration-remigration due to the curvelet diagonal estimation process.



$\mu_k = 20 \log \|\mathbf{A}\mathbf{u}_k - \mathbf{b}\|_2 / \|\mathbf{b}\|_2$



$\nu_k = 20 \log \|\mathbf{A}^*(\mathbf{A}\mathbf{u}_k - \mathbf{b})\|_2 / \|\mathbf{A}^*\mathbf{b}\|_2$

[De Roeck, 2002]

SEG AA' Model w/ Smooth Velocity

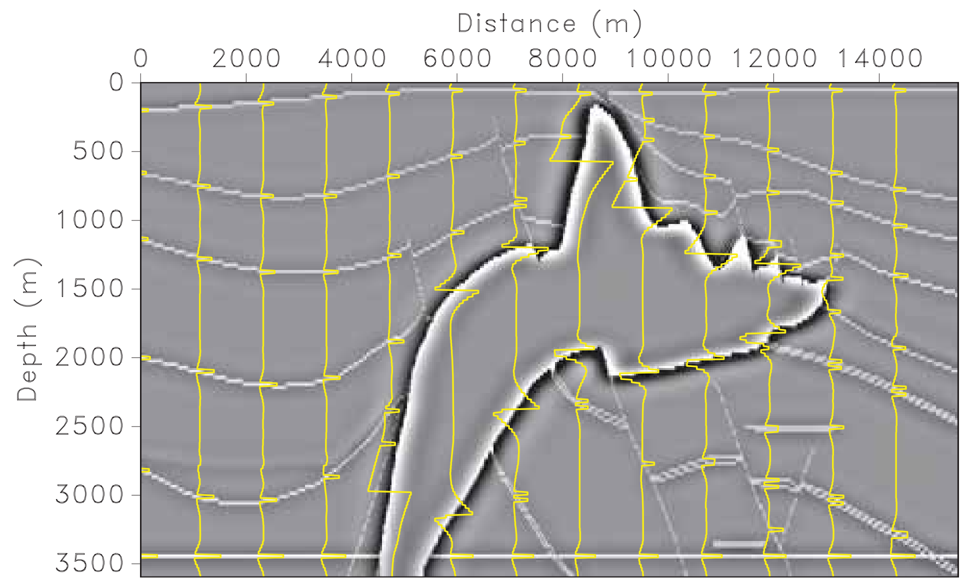
- SEG AA' salt model.

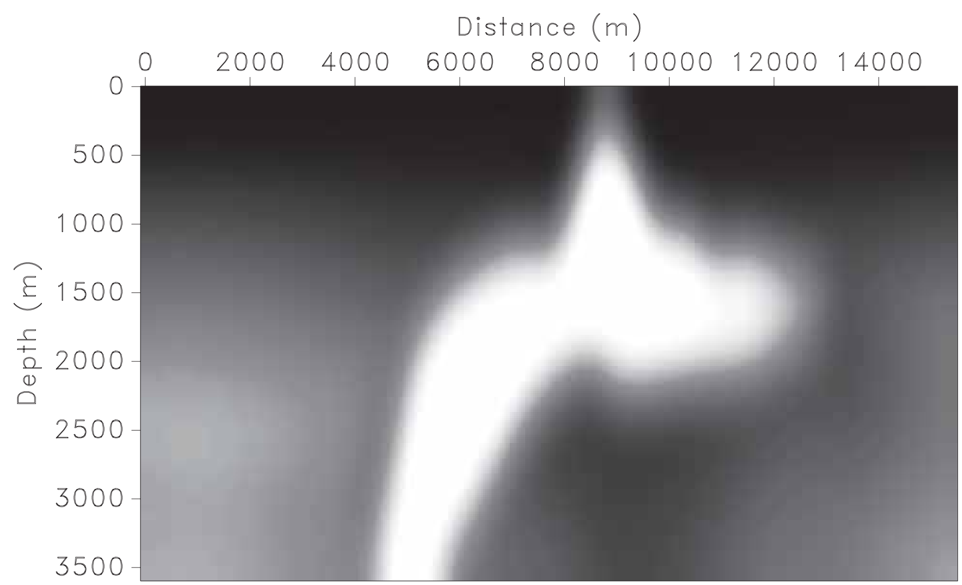
- Our goal is to *improve amplitude recovery*, especially for the reflectors under the salt model.

- We also want to *increase residual decay* for our iterative method.

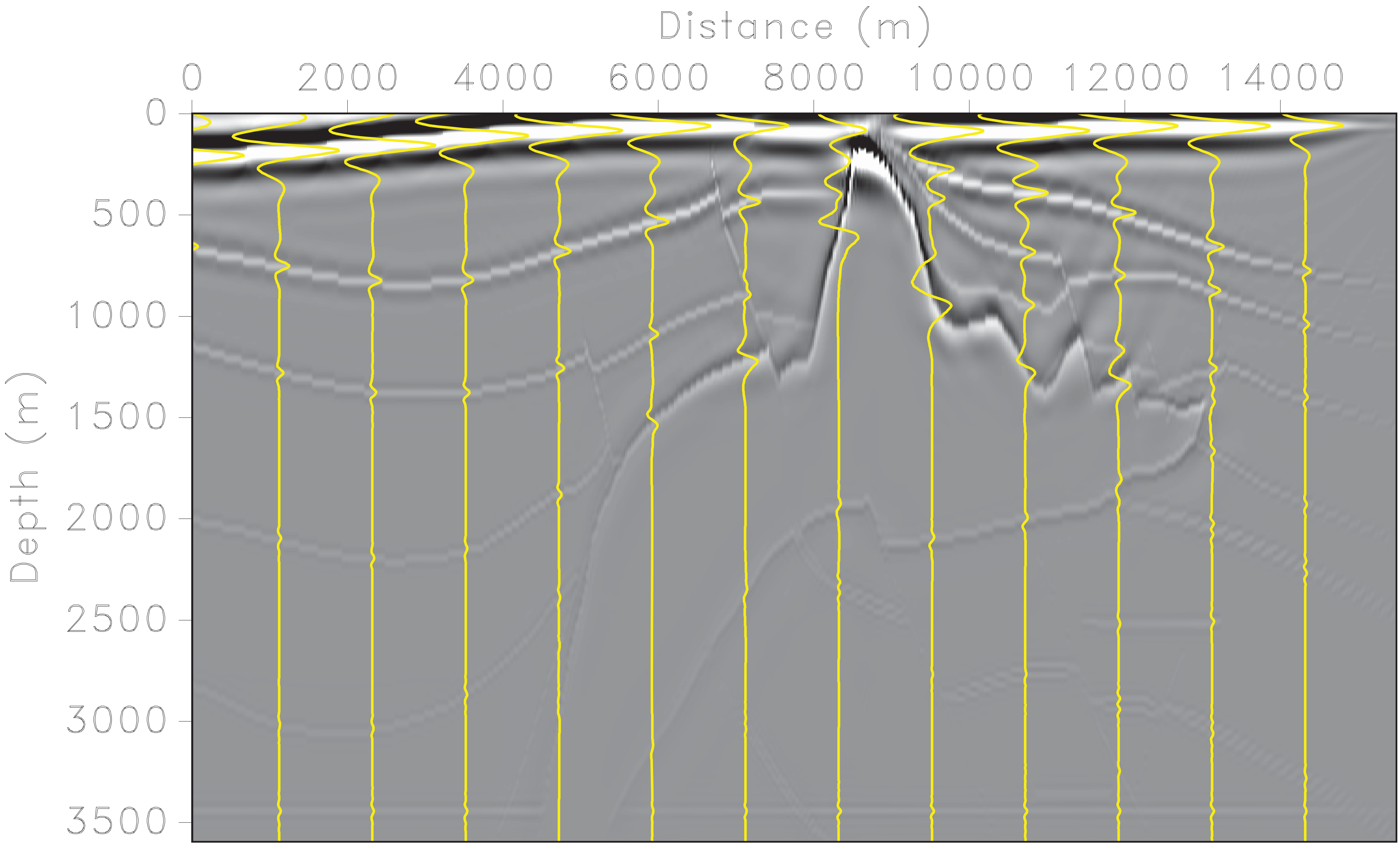
SEG AA' Model w/ Smooth Velocity

- SEG AA' salt model.
- Smooth velocity model.
- 324 shots.
- Each shot 176 traces of 6.4s with a trace interval of 24m.
- Maximum offset of the data is 4224m.



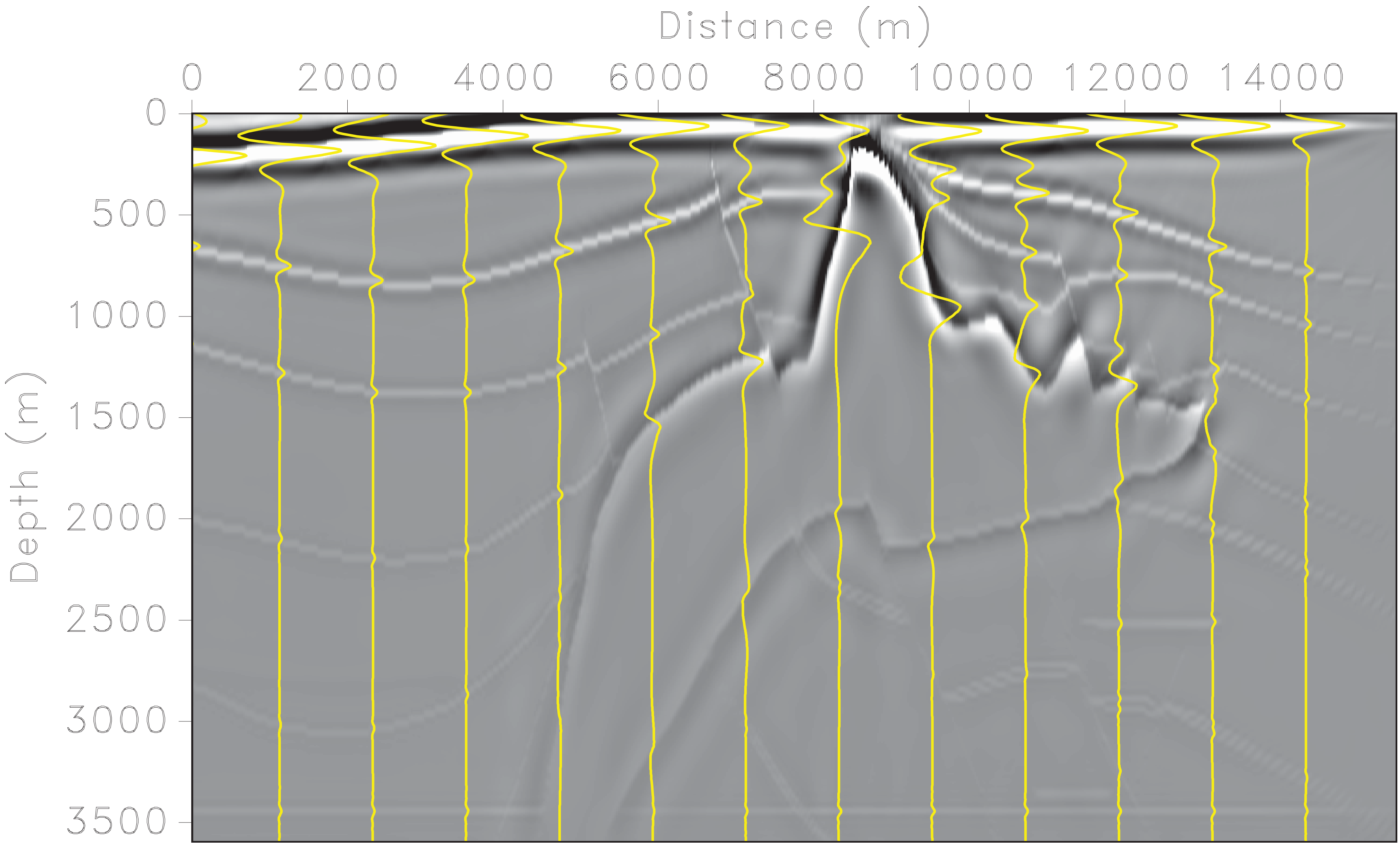


SEG AA’ Model w/ Smooth Velocity



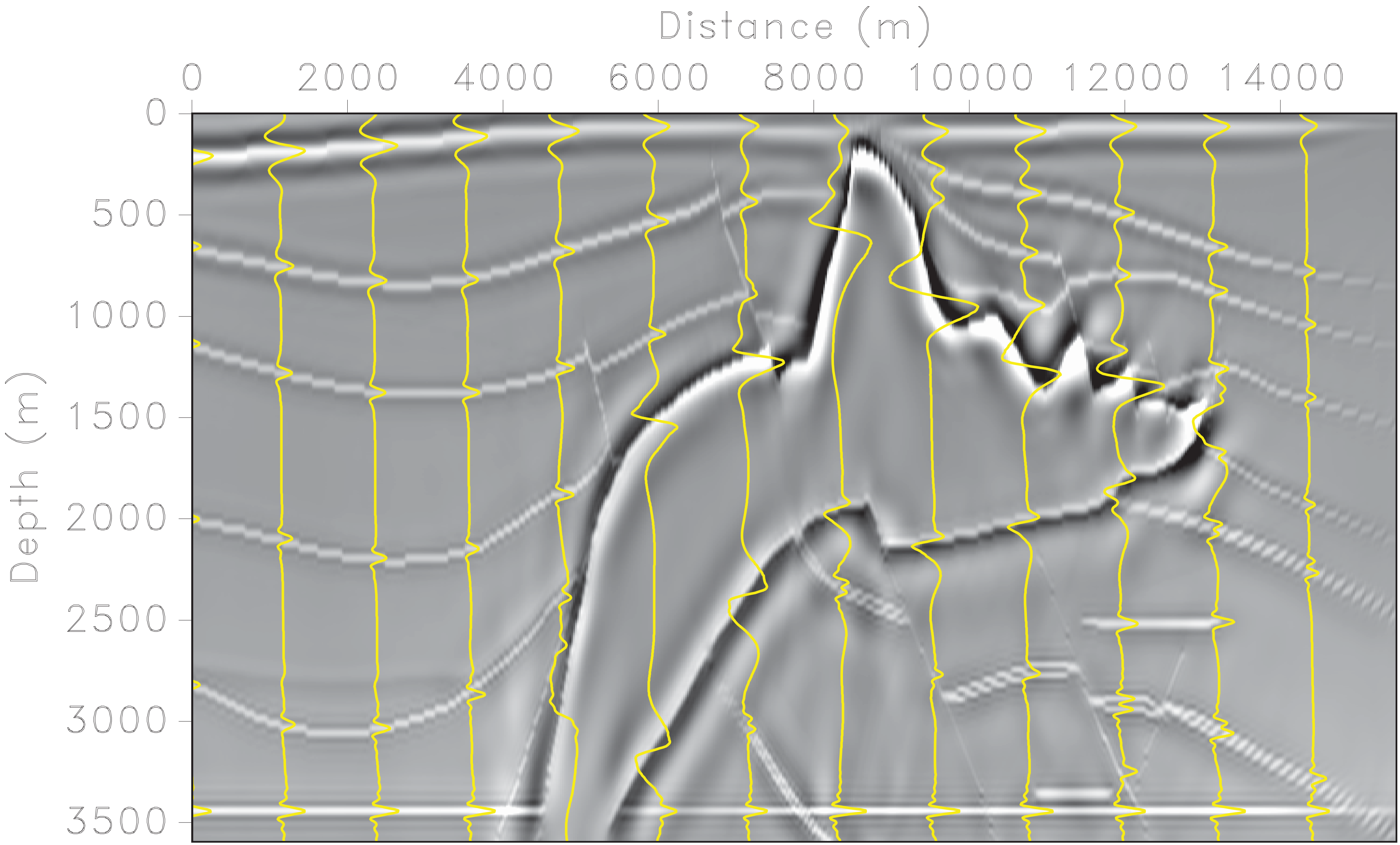
No Preconditioning

SEG AA’ Model w/ Smooth Velocity



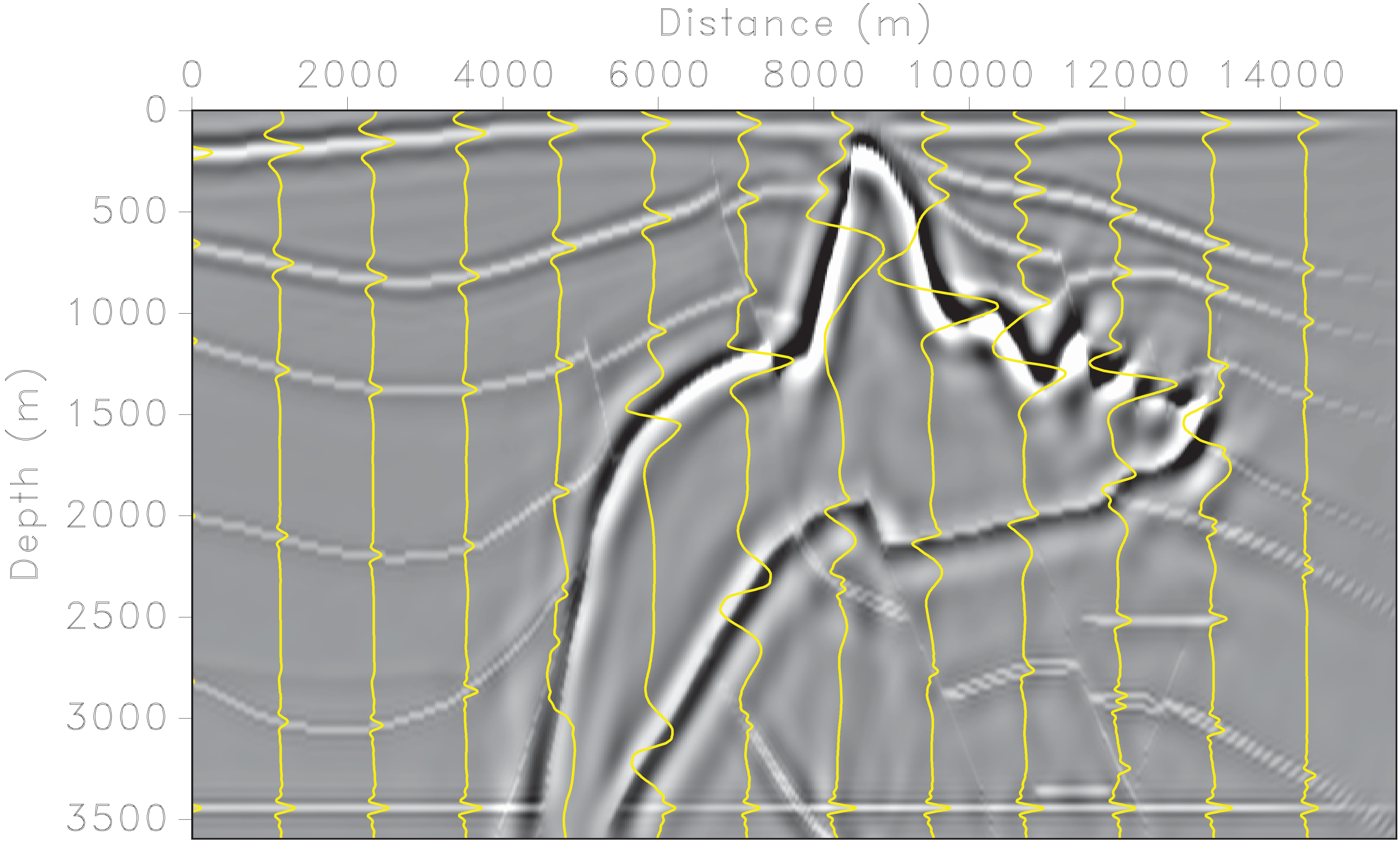
Level I

SEG AA' Model w/ Smooth Velocity



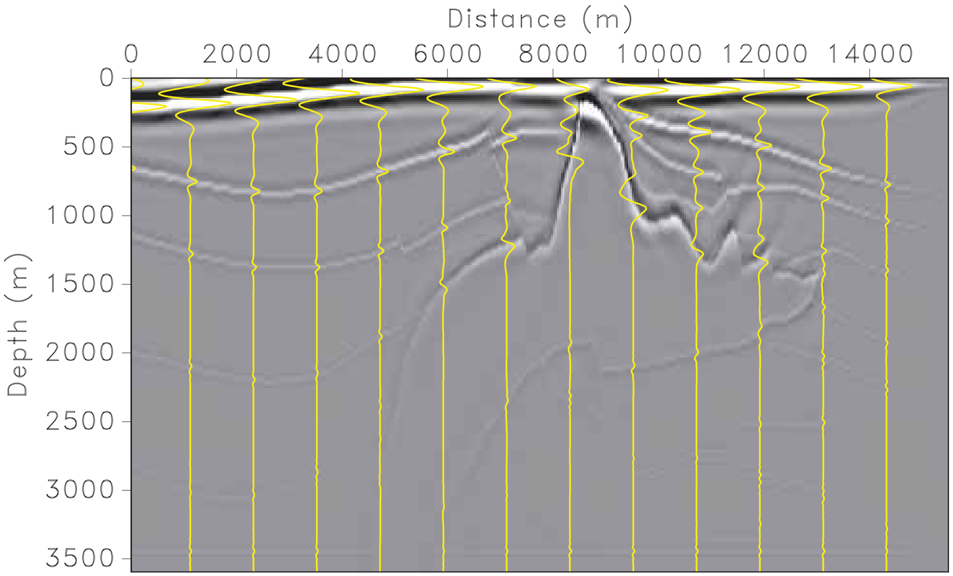
Level II

SEG AA' Model w/ Smooth Velocity

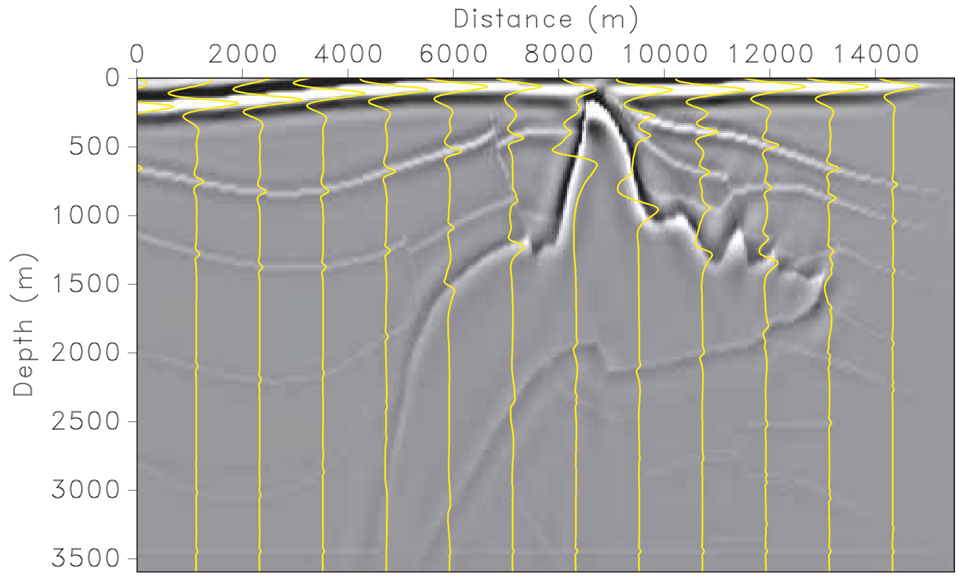


Level III

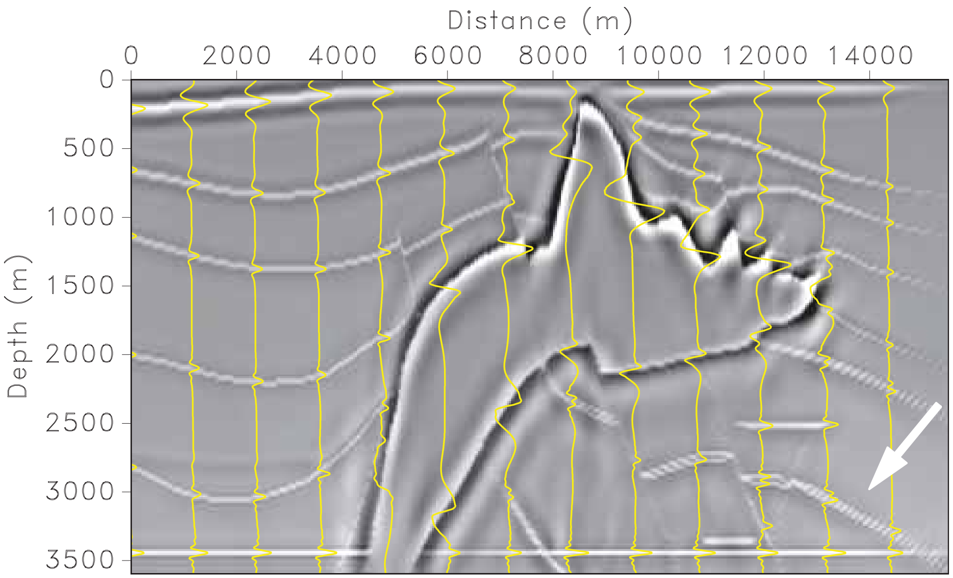
SEG AA' Model w/ Smooth Velocity



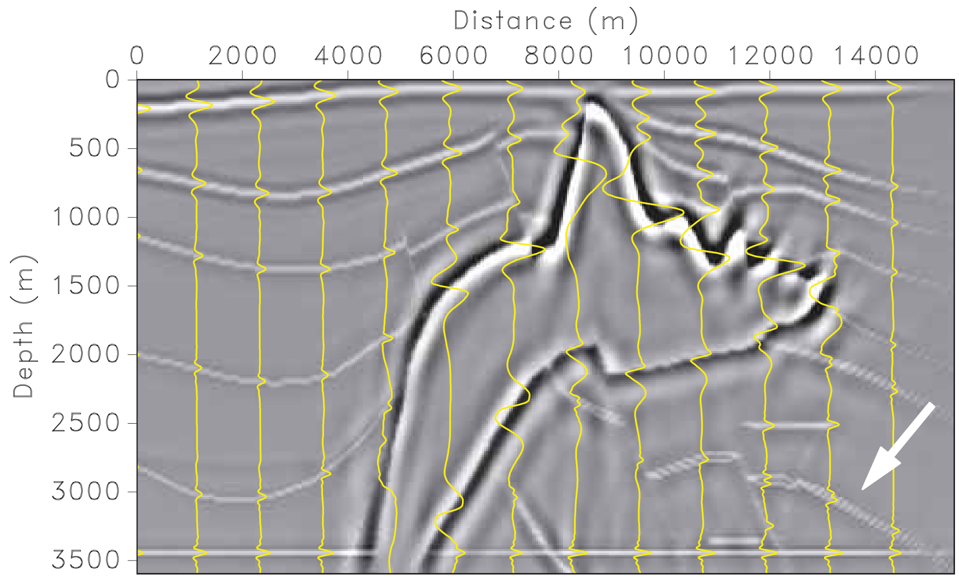
No Preconditioning



Level I

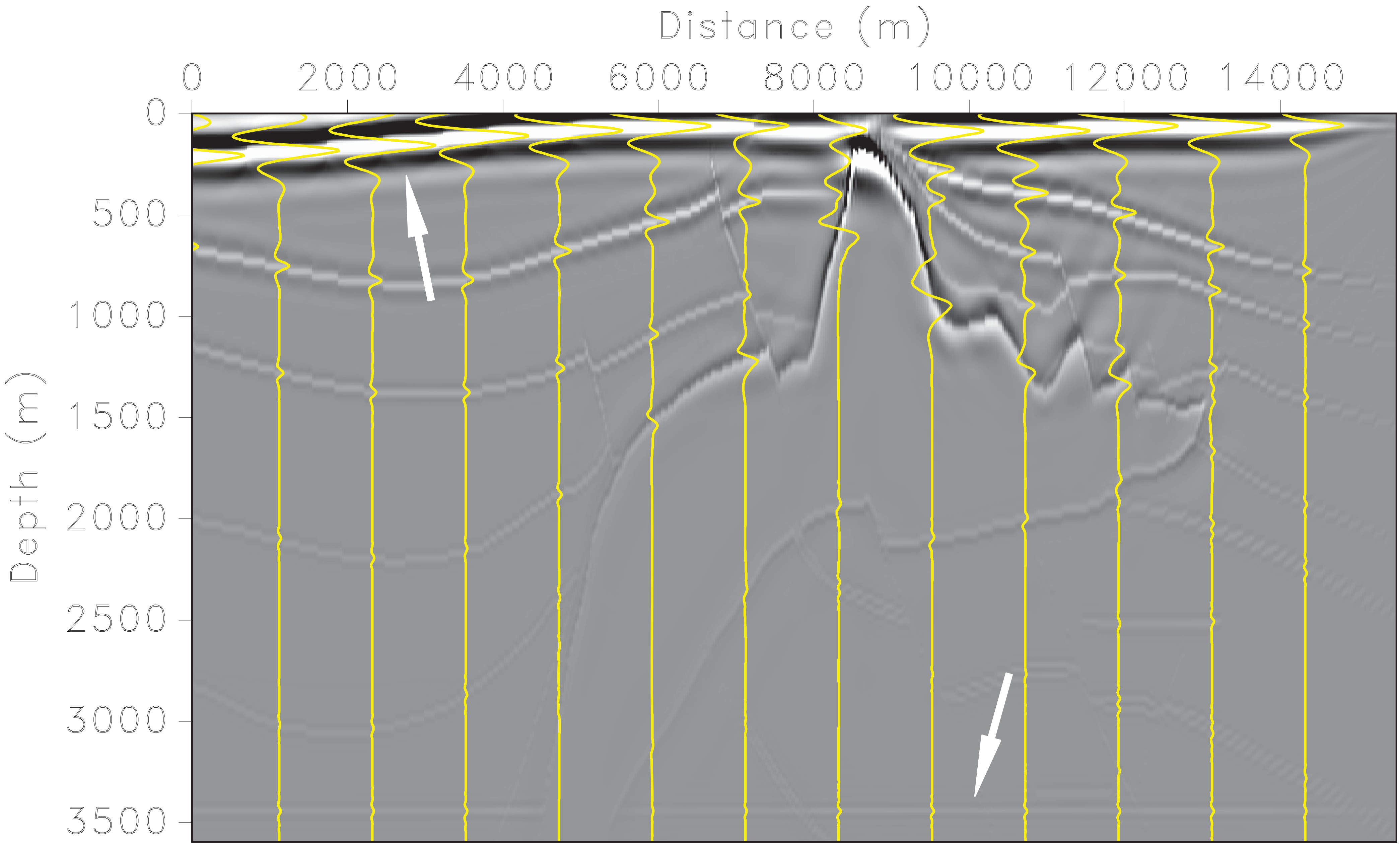


Level II



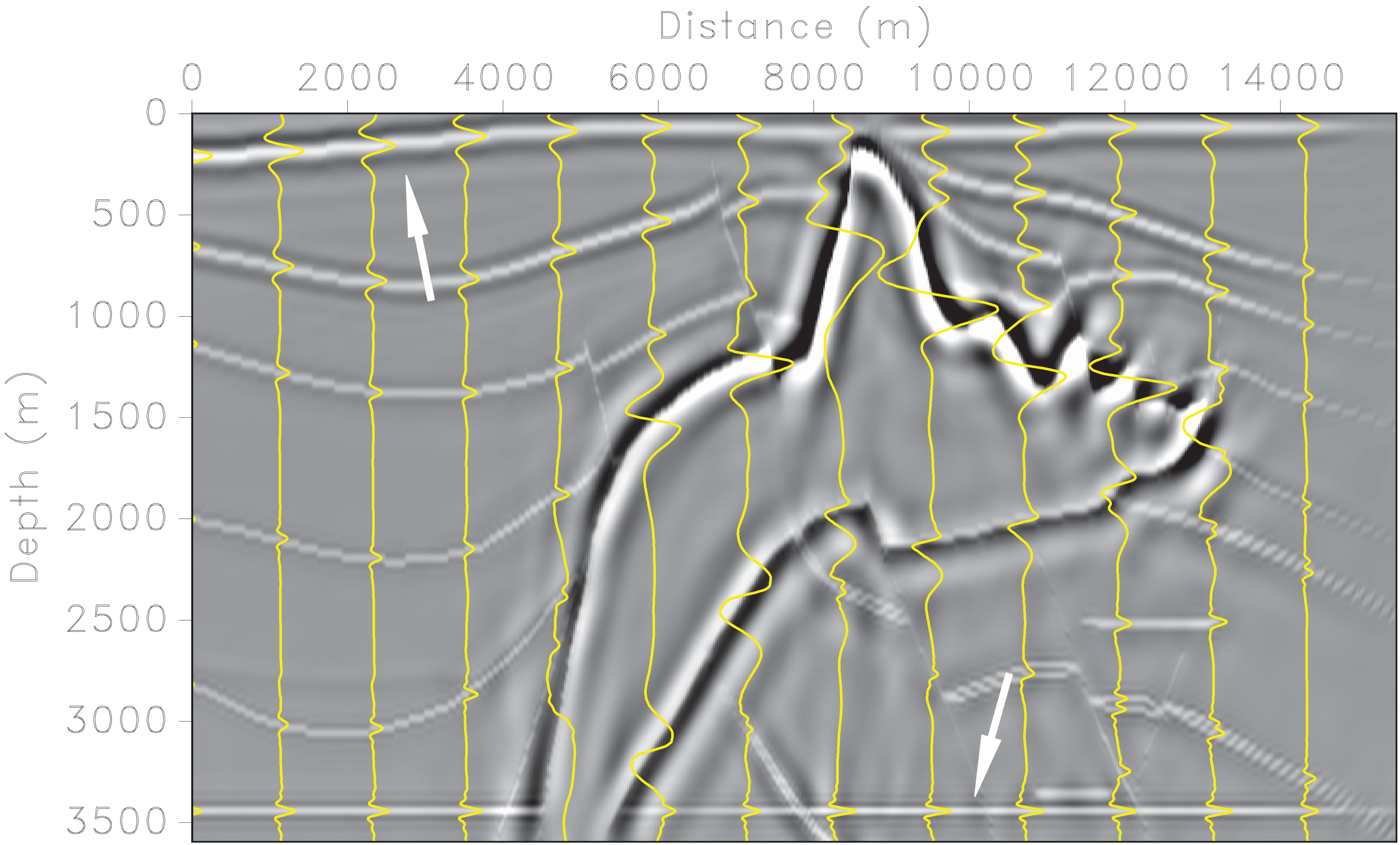
Level III

SEG AA' Model w/ Smooth Velocity



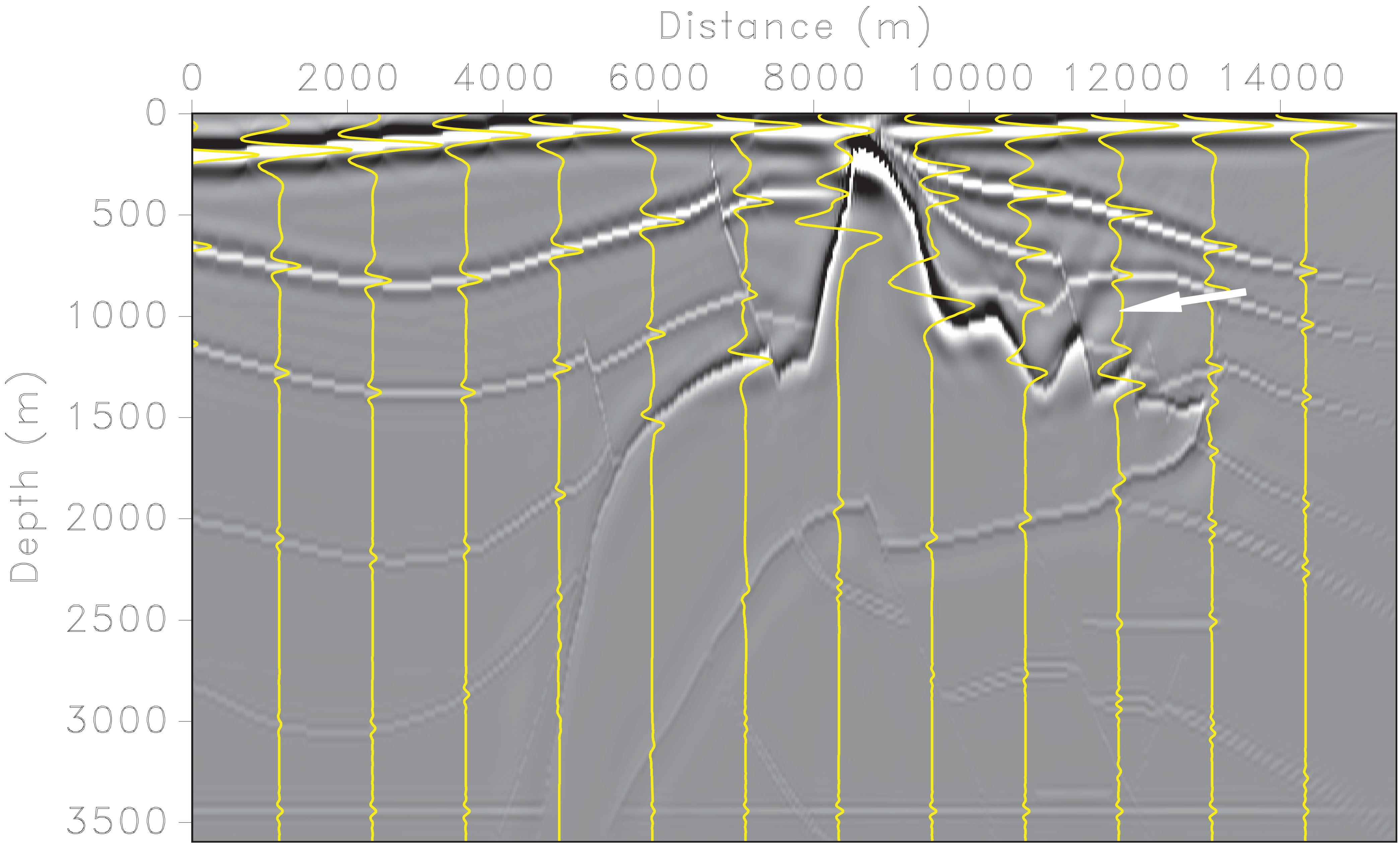
No Preconditioning

SEG AA' Model w/ Smooth Velocity



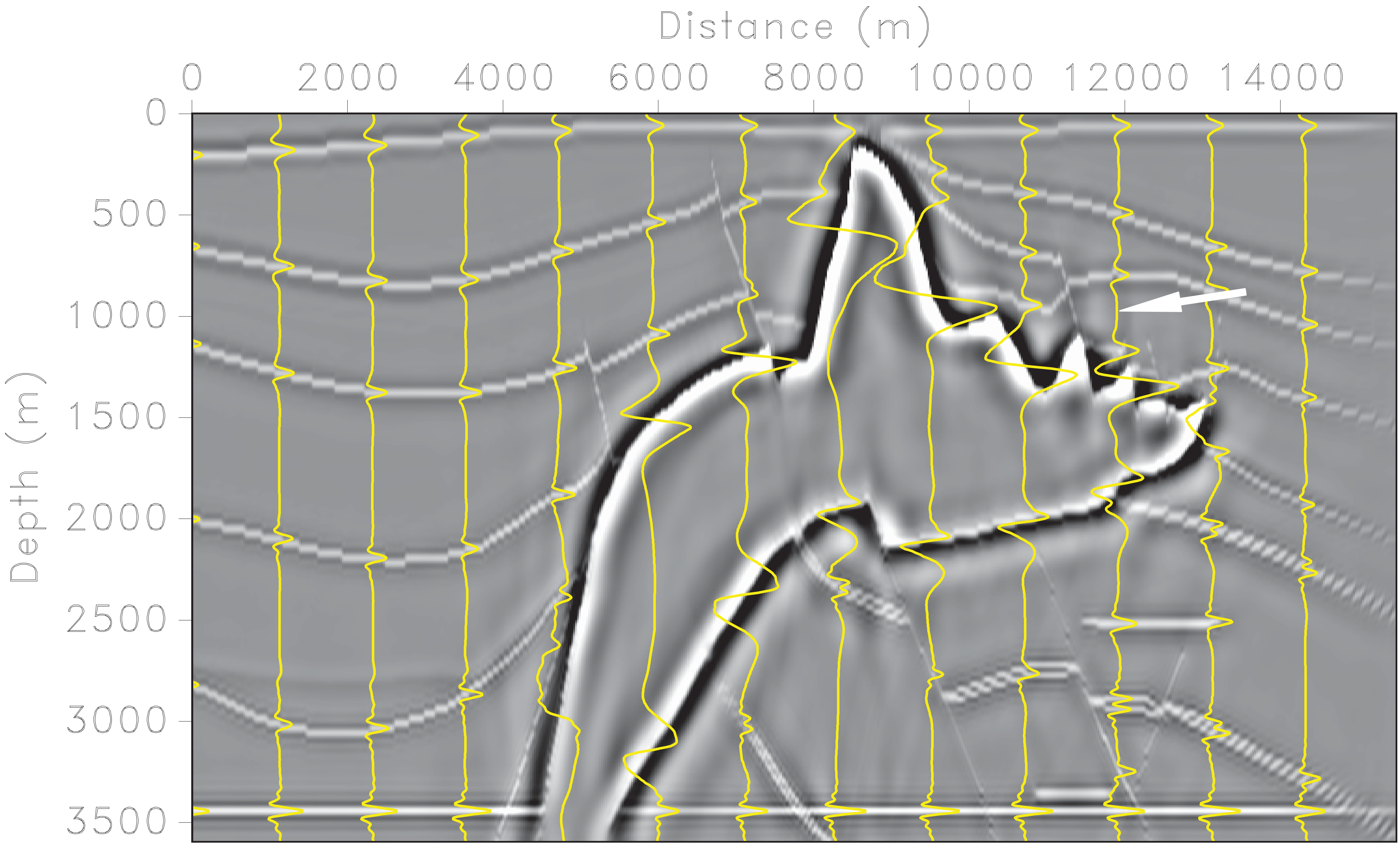
Level III

SEG AA’ Model w/ Smooth Velocity - LSQR Results



LSQR 10 iterations - No Preconditioning

SEG AA' Model w/ Smooth Velocity - LSQR Results



LSQR 10 iterations - Level III

SEG AA' Model w/ Smooth Velocity

- Signal-to-Noise Ratio (SNR) to original reflectivity.

- Defined as follows, with L2 values normalized to one:

$$SNR = 20 \log \|\mathbf{x}_s\|_2 / \|\mathbf{x}_n - \mathbf{x}_s\|_2$$

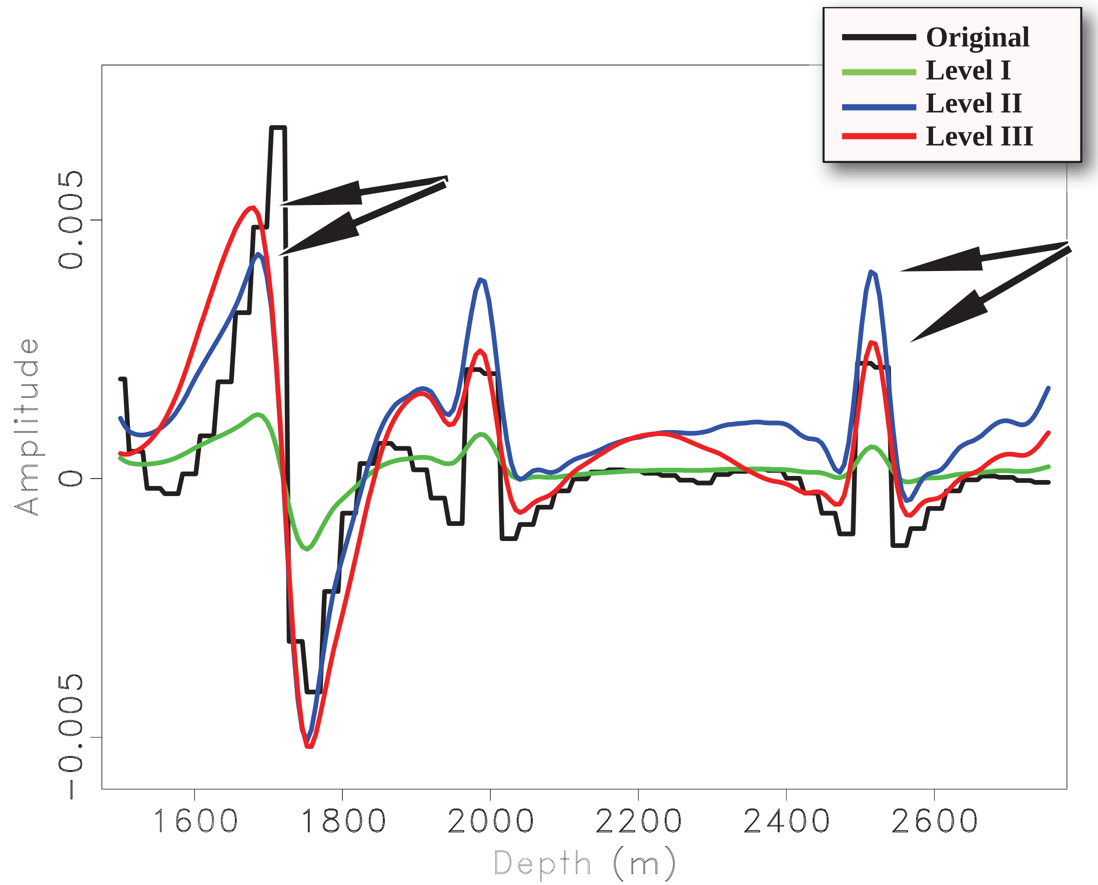
	One iteration SNR	**LSQR results* SNR**
No Preconditioning	-1.9803	-0.9939
Level I	-1.4147	0.3312
Level II	0.4030	3.2690
Level III	1.3122	3.3230

*LSQR to 10 iterations

SEG AA’ Model w/ Smooth Velocity

- Vertical trace at 12720m through the salt model.
- Each preconditioning level is restoring the amplitudes closer to the original.
- Increase or decrease amplitudes, not just a direct linear scaling.
- Level III (curvelet-based diagonal combination) is doing the most significant amplitude recovery in this case.

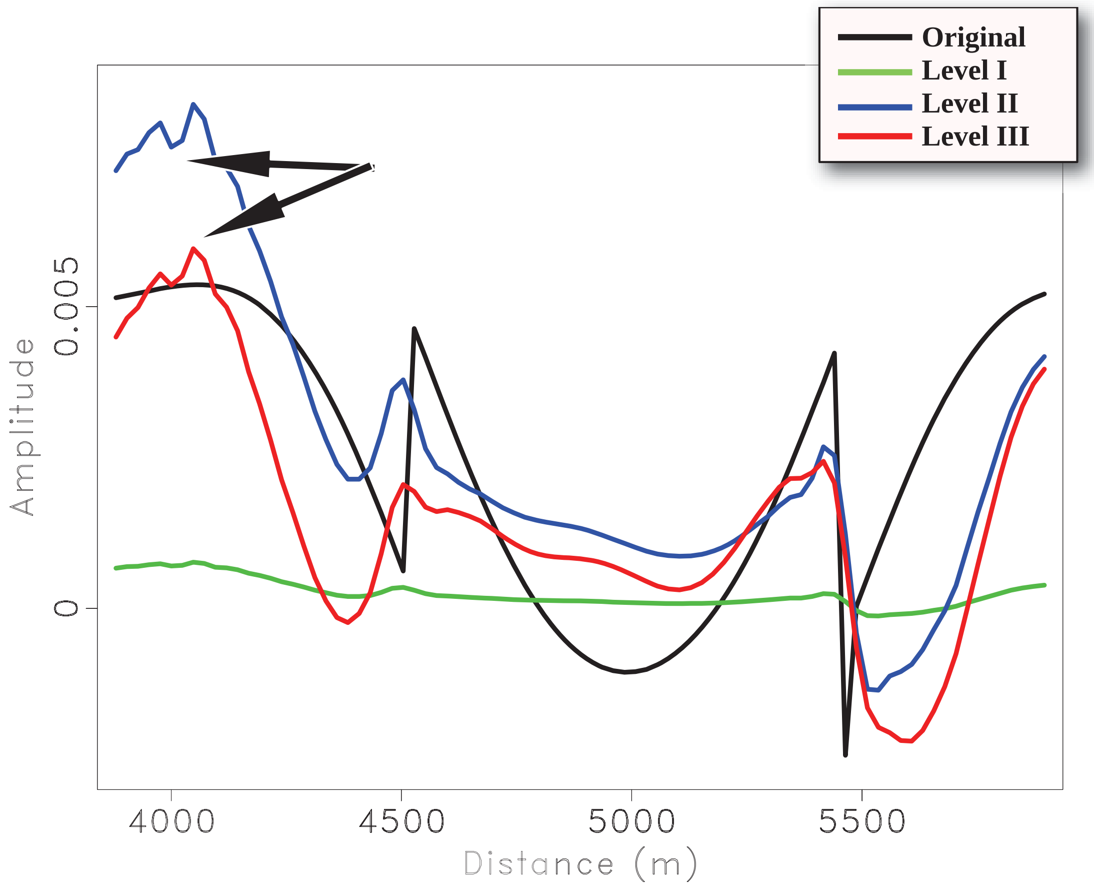
Vertical trace near the tip of the salt model.



SEG AA' Model w/ Smooth Velocity

- Horizontal trace at 3438m through the reflector at the bottom.
- Section where the salt model meets the reflector.
- Can see our preconditioner is improving amplitude corrections.

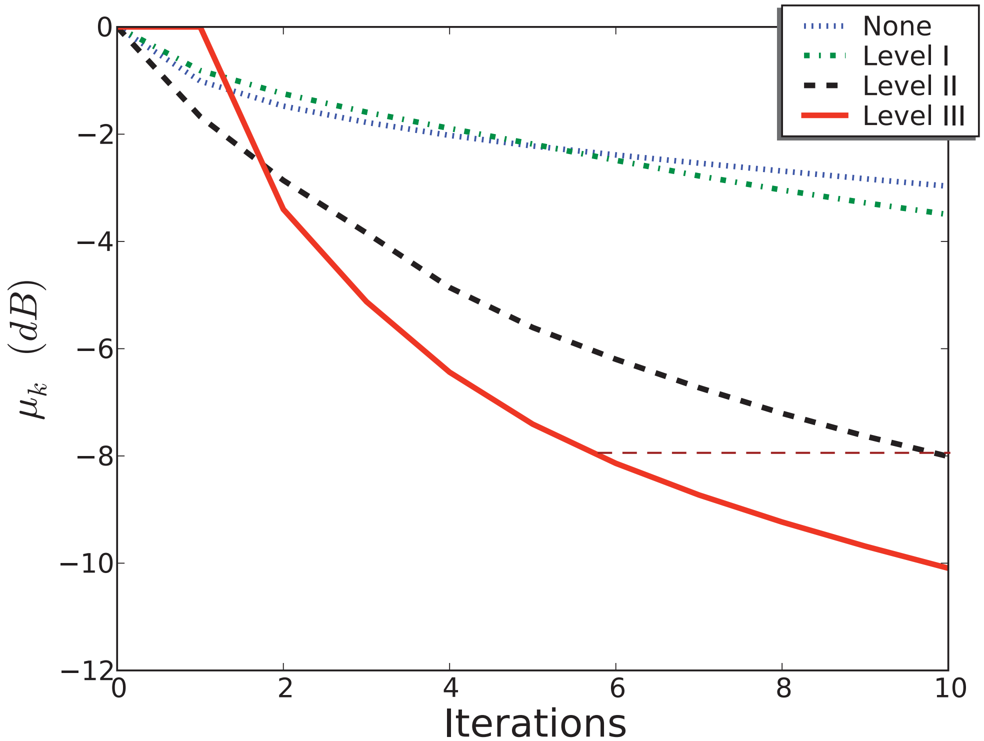
Horizontal trace where salt model meets the bottom reflector.

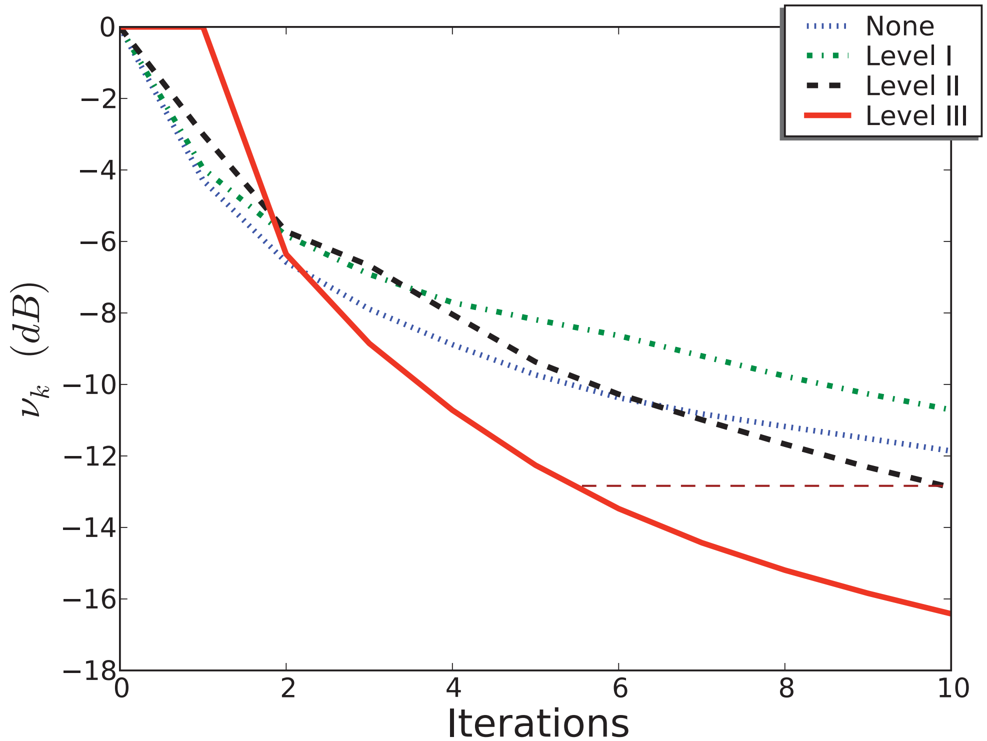


SEG AA' Model w/ Smooth Velocity

- Residual decay for the data-space and model-space residuals.
- Even after our first few iterations of level III preconditioning, we quickly improve upon the other levels in each figure.

- The red line has already seen one migration-remigration due to the curvelet diagonal estimation process.





$$\mu_k = 20 \log \|\mathbf{A}\mathbf{u}_k - \mathbf{b}\|_2 / \|\mathbf{b}\|_2$$

$$\nu_k = 20 \log \|\mathbf{A}^*(\mathbf{A}\mathbf{u}_k - \mathbf{b})\|_2 / \|\mathbf{A}^*\mathbf{b}\|_2$$

[De Roeck, 2002]

Conclusions

- We can achieve **significant residual decay** using our series of preconditioning matrices.

- Amplitudes throughout the model are **recovered more accurately** to the original reflectivity.

- We do the same amount of work, but get a better result.

- **We satisfy Zipf's Principle of Least Effort!**

Speculations on Real Data

- On real data our curvelet-based diagonal estimation should greatly improve the image.
  - Curvelets add robustness to the presence of coherent noise.
  - Also moderates errors in the linearized Born modeling operator.

- Small shifts over the support of a curvelet will not adversely affect the corresponding curvelet coefficient.
  - Allow imperfections in the velocity model.

Acknowledgments

- All the examples were computed using a SLIMpy script to Madagascar with a wrapper to Symes' RTM Code.

- We would like to thank:
  - Bill Symes for use of his 2D Acoustic Post-Stack Reverse-Time Migration code.
  - Madagascar Development Team (http://reproducibility.org/).
  - CurveLab Developers (http://www.curvelet.org/).
  - SLIMpy Developers (http://slim.eos.ubc.ca/SLIMpy/).

- This work was in part financially supported by the NSERC Discovery (22R81254) and CRD Grants DNOISE (334810-05) of F.J.H. and was in part carried out within the SINBAD project with support, secured through ITF, from BG Group, BP, Chevron, ExxonMobil and Shell.

SLIMpy Web Pages

- More information about SLIMpy can be found at the SLIM homepage:

  http://slim.eos.ubc.ca

- Auto-books and tutorials can be found at the SLIMpy generated websites:

  http://slim.eos.ubc.ca/SLIMpy/

References

- For more information please look at a recently submitted letter to Geophysics:
  - Herrmann, F. J., C. R. Brown, Y. A. Erlangga, and P. P. Moghaddam, 2008, Curvelet-based migration preconditioning, http://slim.eos.ubc.ca/Publications/Public/Journals/herrmann08cmp.pdf.

- Other papers to consider looking at:

  - De Roeck, Y., 2002, Sparse linear algebra and geophysical migration: A review of direct and iterative methods: Numerical Algorithms, 29, 283–322.
  - Herrmann, F. J., P. P. Moghaddam, and C. C. Stolk, 2008, Sparsity- and continuity-promoting seismic imaging with curvelet frames: Journal of Applied and Computational Harmonic Analysis, 24, 150–173. (doi:10.1016/j.acha.2007.06.007).
  - Paige, C. C. and M. A. Saunders, 1982, LSQR: An algorithm for sparse linear equations and sparse least squares: ACM TOMS, 8, 43–71.
  - Symes, W. W., 2008, Approximate linearized inversion by optimal scaling of prestack depth migration: Geophysics, 73, R23–R35. (10.1190/1.2836323).