Mismatches between physics and operators for least squares Kirchhoff and RTM

Daniel Trad

Overview

- Least squares inversion
- LSRTM: differences between finite differences and reflector modeling
- LS Kirchhoff: the accumulation of noise because of mismatches.
- Mapping operators deficiencies into the residuals and model
- Controlling the gradient by adaptive data and model weights
- Data and model space mappings
- Conclusions

Least squares formulation: modeling vs migration

Undesired features = discrepancy between prediction and data + size of model

L the operator (L modeling, L^H adjoint).
d acquired data
m model

$$\mathbf{d} = \mathbf{W_d L W_m m}$$

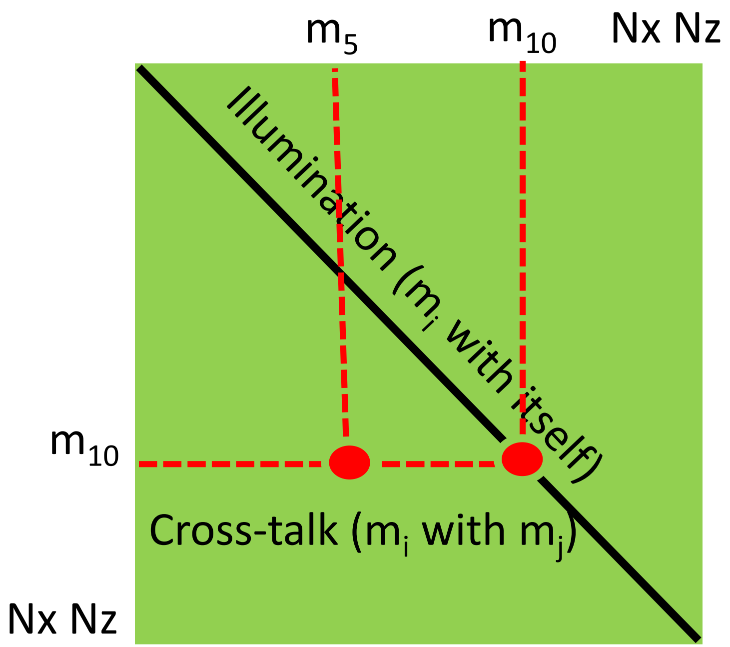
Wd data space weights
Wm model space weights

Least Squares inversion

$$J = \underbrace{\|\mathbf{d} - \mathbf{Lm}\|^2}_{\text{Data residuals in a particular norm choice}} + \lambda \underbrace{\|\mathbf{m}\|_{\mathbf{W}}}_{\text{Model size in a particular norm choice}}$$

W weights to inforce a particular solution

Hessian= L^HL



Iteratively solve…

$$\mathbf{m} = \underbrace{(\mathbf{L^H L} + \mathbf{W^H W})^{-1}}_{\text{Inverse of Hessian}} \underbrace{\mathbf{L}^H \mathbf{d}}_{\text{mapping}}$$

Kirchhoff forward and adjoint operators

Adjoint Operator

$$m(\overrightarrow{x}) = \int W(t, \overrightarrow{x}, vel)d(\overrightarrow{\varepsilon}, t = t_s + t_r)d\overrightarrow{\varepsilon}$$

$$\mathbf{m} = \mathbf{L^H d}$$

Forward Operator

$$d(\overrightarrow{\varepsilon}, t = t_s + t_r) = \int W^*(t, \overrightarrow{x}, vel)m(\overrightarrow{x})d\overrightarrow{x}$$

$$\mathbf{d} = \mathbf{Lm}$$

RTM Forward and adjoint operators

Adjoint Operator

Forward time propagate source wavefield

$$\frac{\partial^2 \phi_s}{\partial x^2} + \frac{\partial^2 \phi_s}{\partial z^2} - \frac{1}{v^2}\frac{\partial^2 \phi_s}{\partial t^2} = f_s$$

Reverse time propagate data

$$\frac{\partial^2 \phi_r}{\partial x^2} + \frac{\partial^2 \phi_r}{\partial z^2} - \frac{1}{v^2}\frac{\partial^2 \phi_r}{\partial t^2} = 0$$

$$\text{with } \phi_r(x, z = 0, t) = d_{obs}(x, z, t)$$

cross-correlation IC in time

$$I_{croscorr} = \int \phi_s(x, z, t)\phi_r(x, z, T - t)dt$$

$$\mathbf{m} = \mathbf{L}^{\mathbf{H}}\mathbf{d}$$

Forward Operator

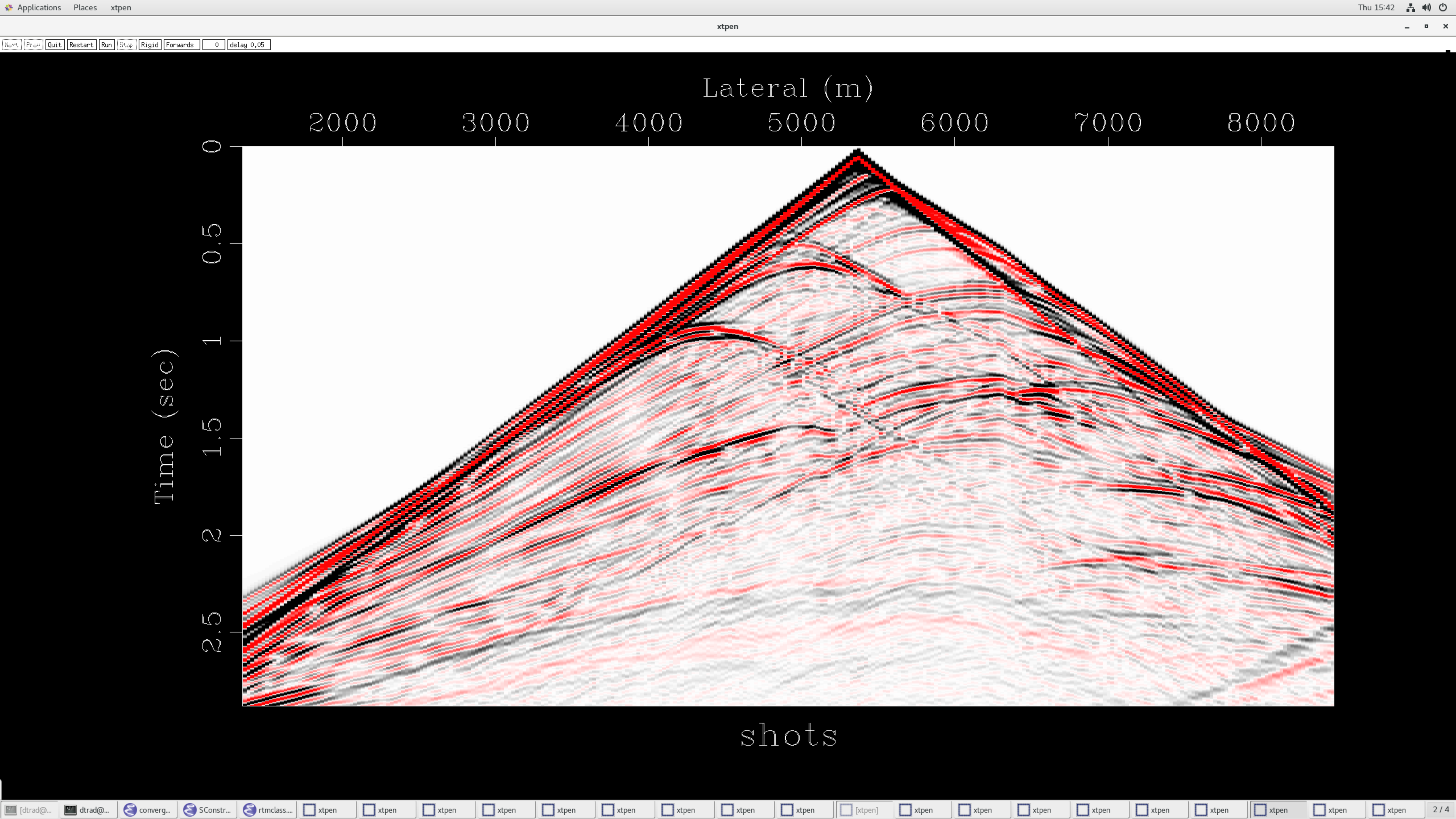
$$\frac{\partial^2 \phi_s}{\partial x^2} + \frac{\partial^2 \phi_s}{\partial z^2} - \frac{1}{v^2}\frac{\partial^2 \phi_s}{\partial t^2} = f_s$$

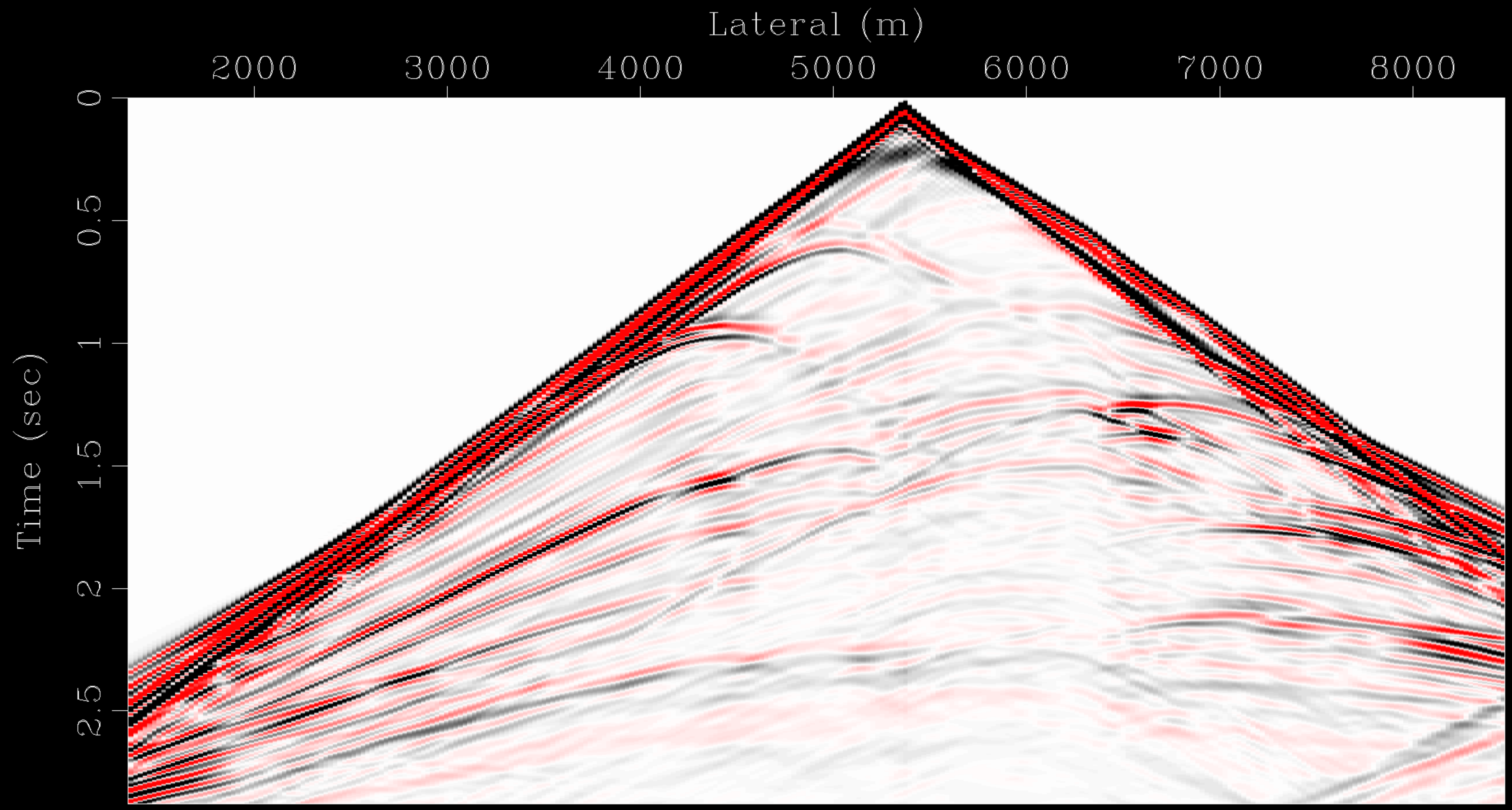
Born modeling

$$\phi_r(x, z, t) = \int \phi_s(x, z, t) \times I_r(x, z)dt$$

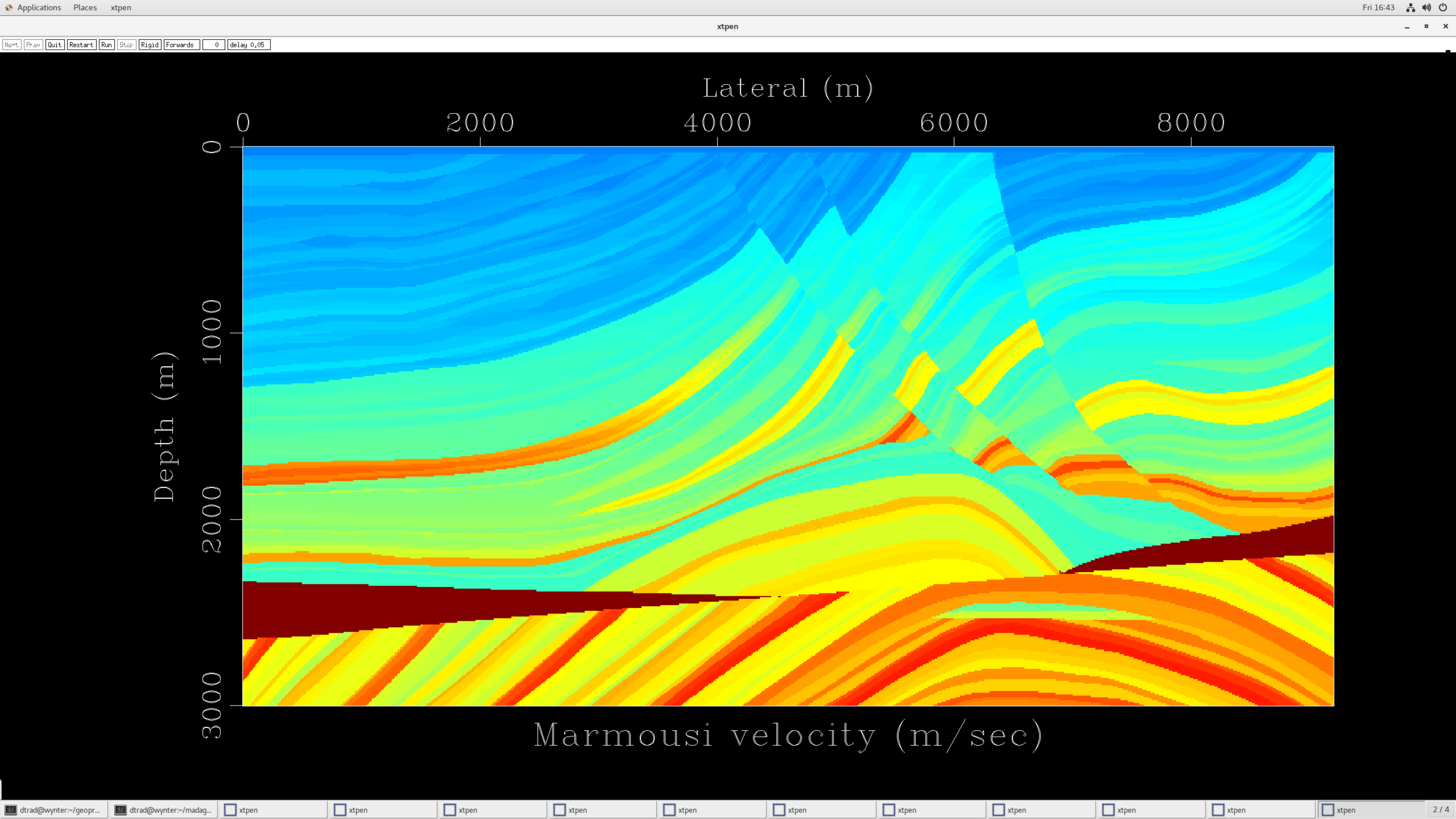
$$\mathbf{d} = \mathbf{L}\mathbf{m}$$

Smooth model

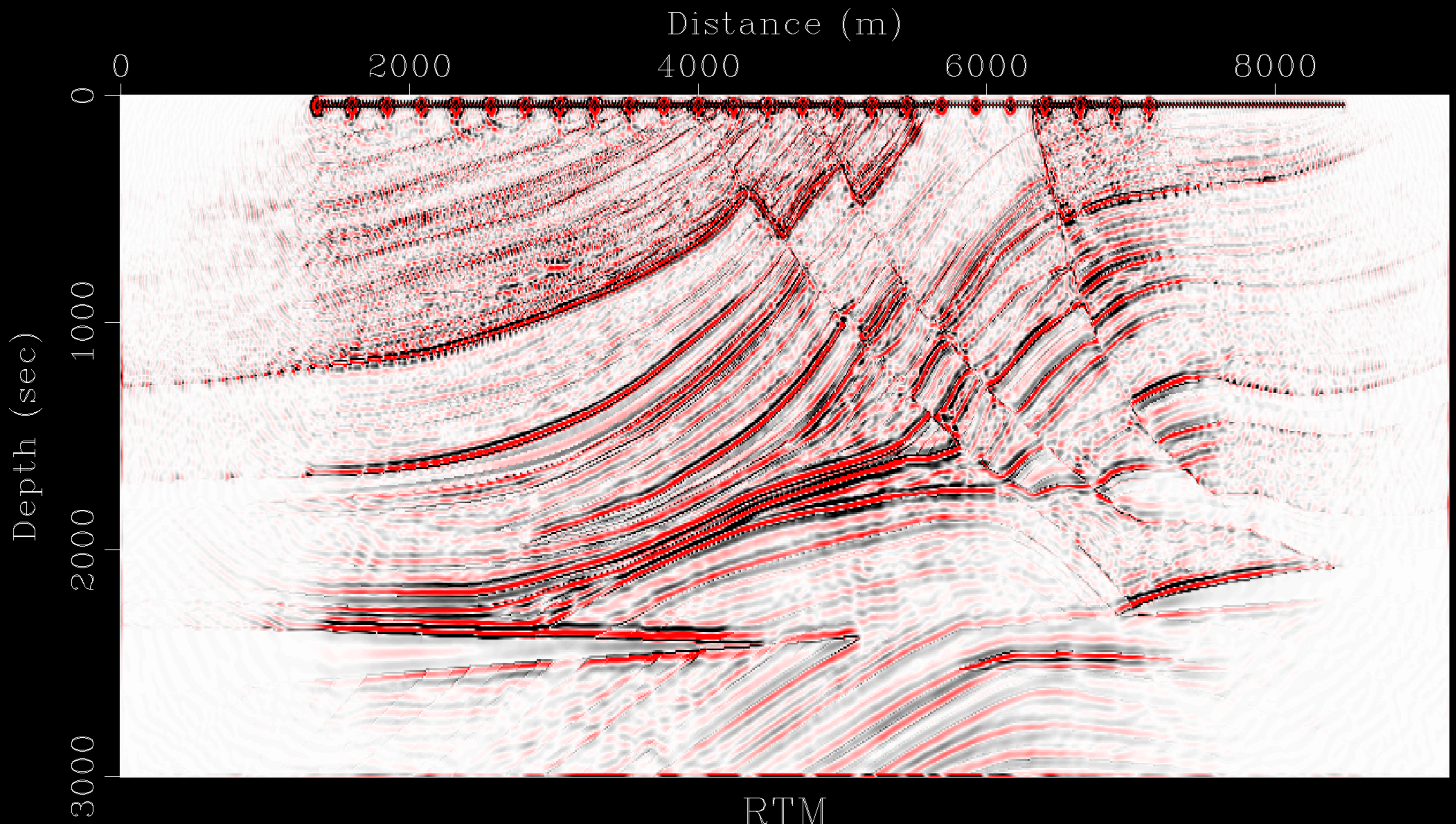

Applications
Places
xtpen
Thu 15:42
xtpen
Next
Prev
Quit
Restart
Run
Stop
Rigid
Forwards
0
delay 0.05
Lateral (m)
2000
3000
4000
5000
6000
7000
8000
0
0.5
1
1.5
2
2.5
Time (sec)
shots
2 / 4



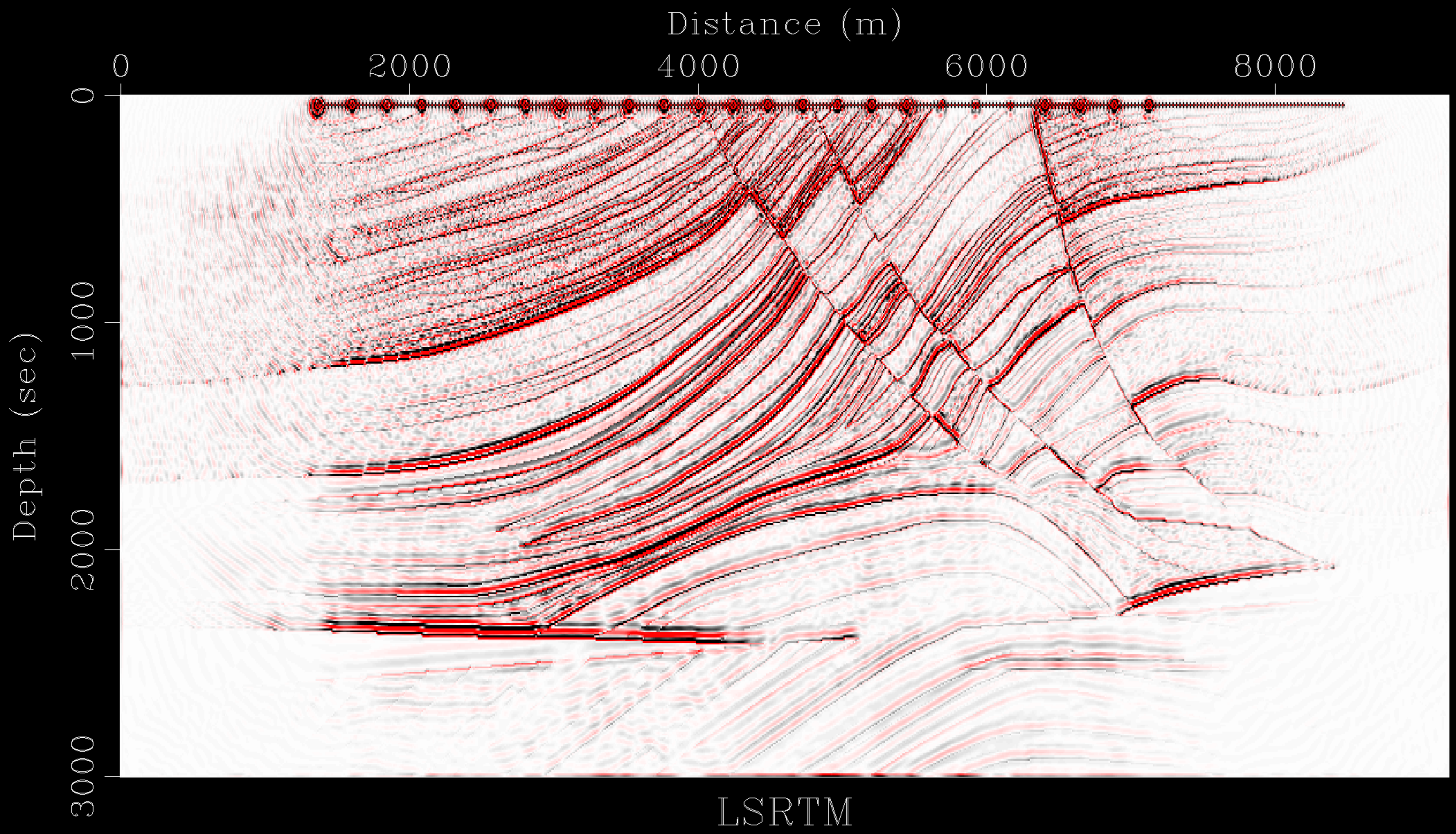
Predicted from smooth model from one shot (1 iteration)


Applications
Places
xtpen
Fri 16:43
xtpen
Next
Prev
Quit
Restart
Run
Stop
Rigid
Forwards
0
delay 0.05
Lateral (m)
0
2000
4000
6000
8000
Depth (m)
0
1000
2000
3000
Marmousi velocity (m/sec)
dtrad@wynter:~/geopr...
dtrad@wynter:~/madag...
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
xtpen
2 / 4

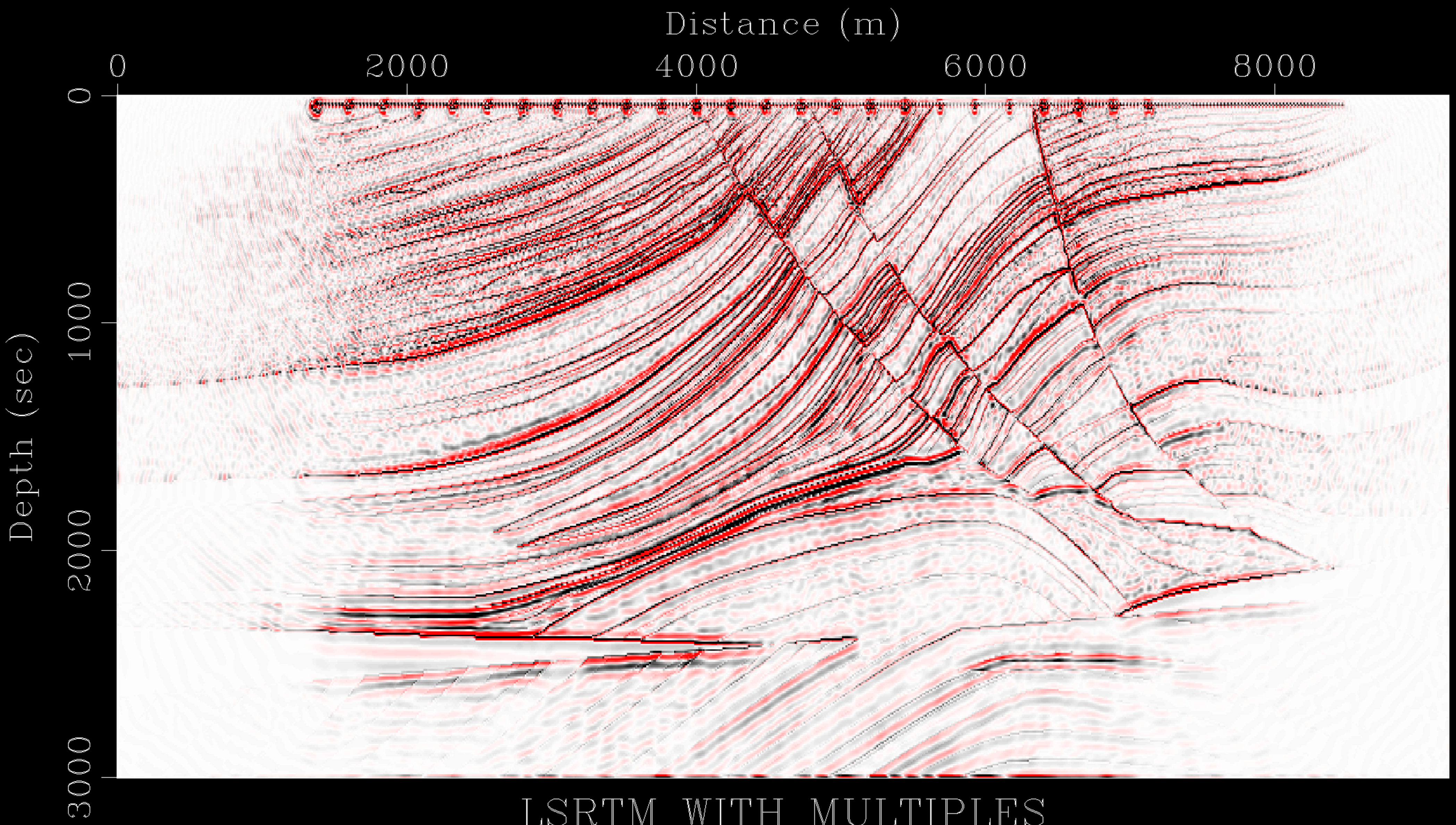
RTM (25 shots)


Distance (m)
0
2000
4000
6000
8000
Depth (sec)
0
1000
2000
3000
RTM

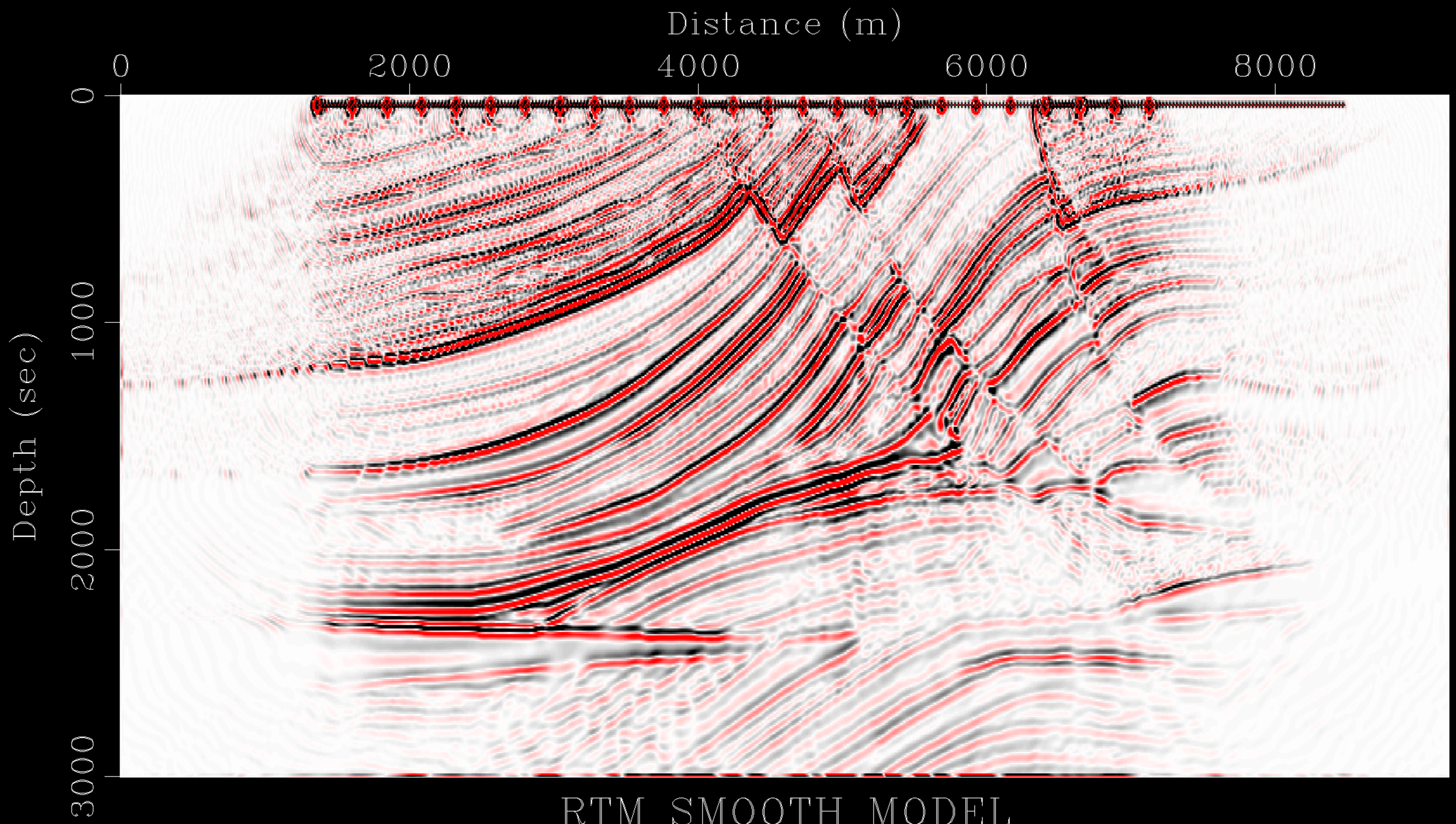
LSRTM (9 iterations)



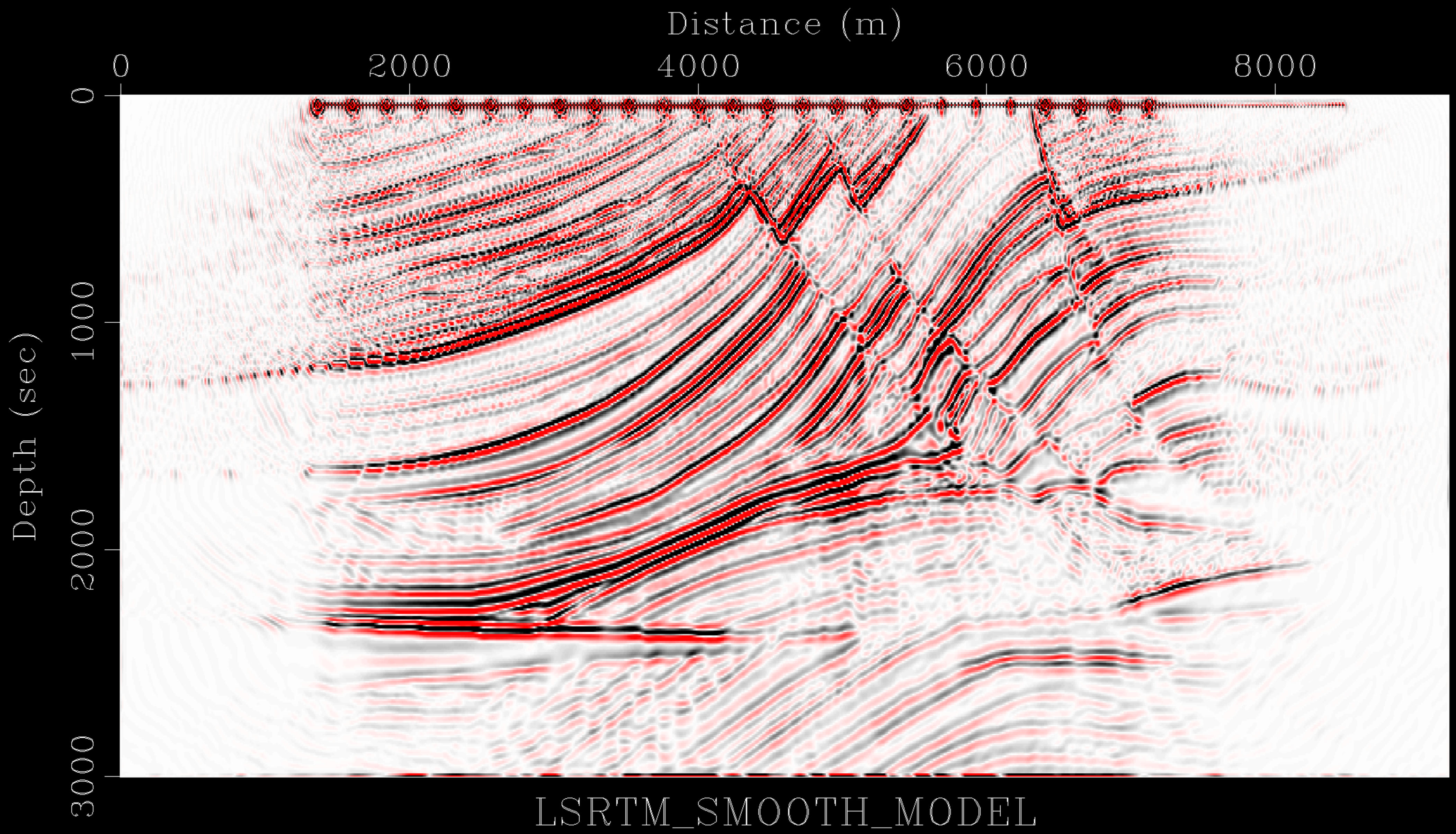
LSRTM of data with surface multiples



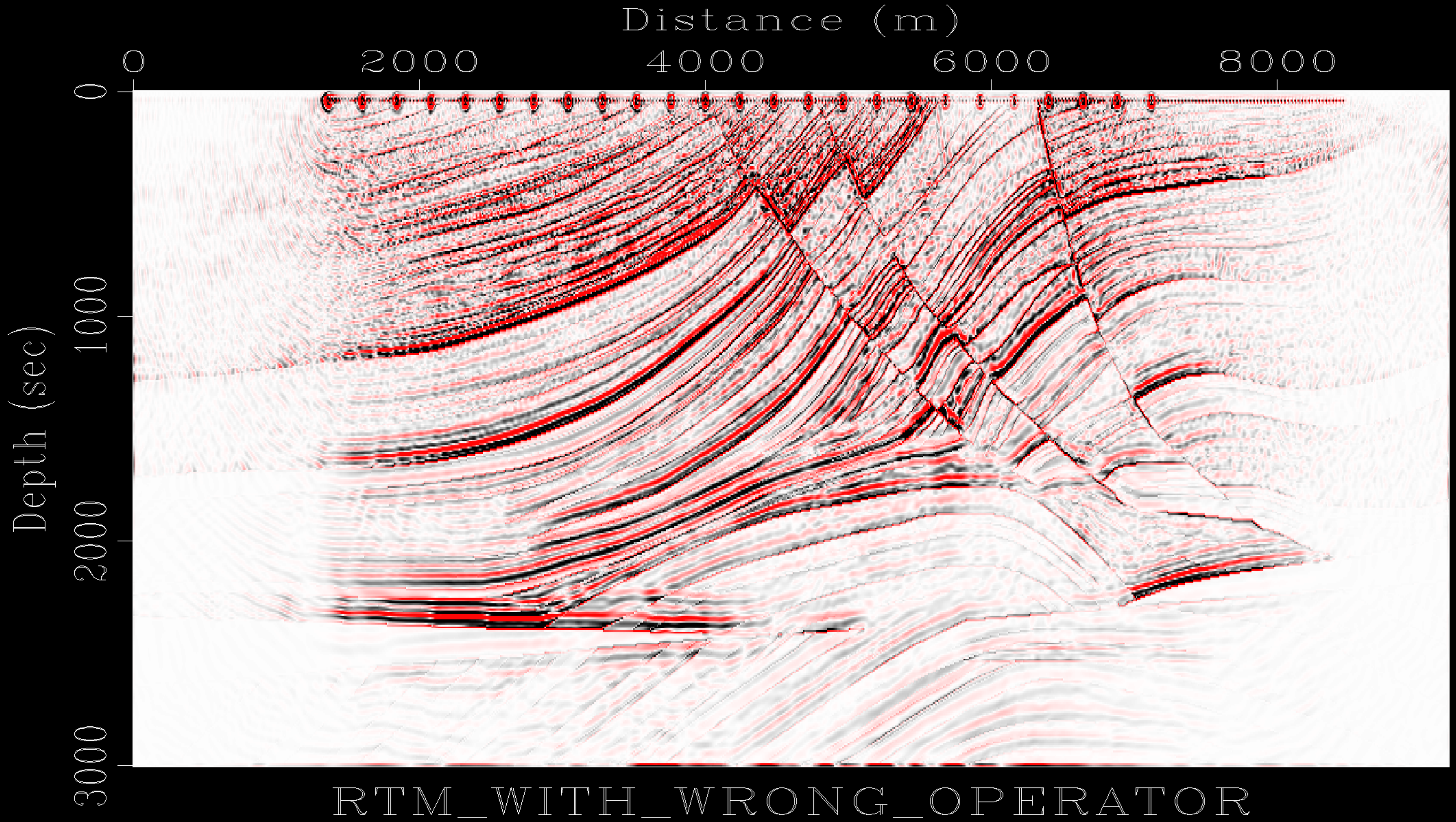
RTM with smooth model



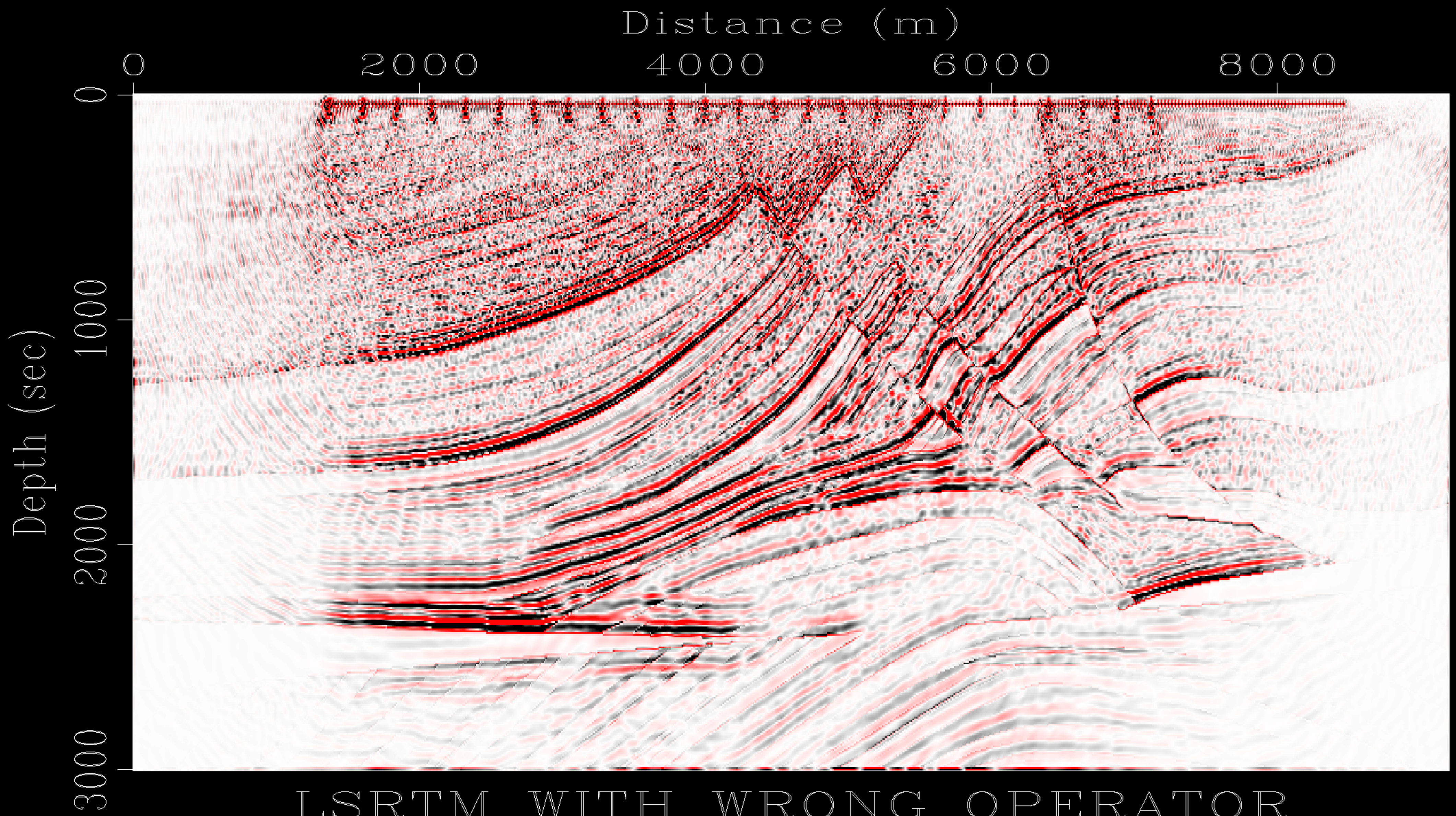
LSRTM from smooth model



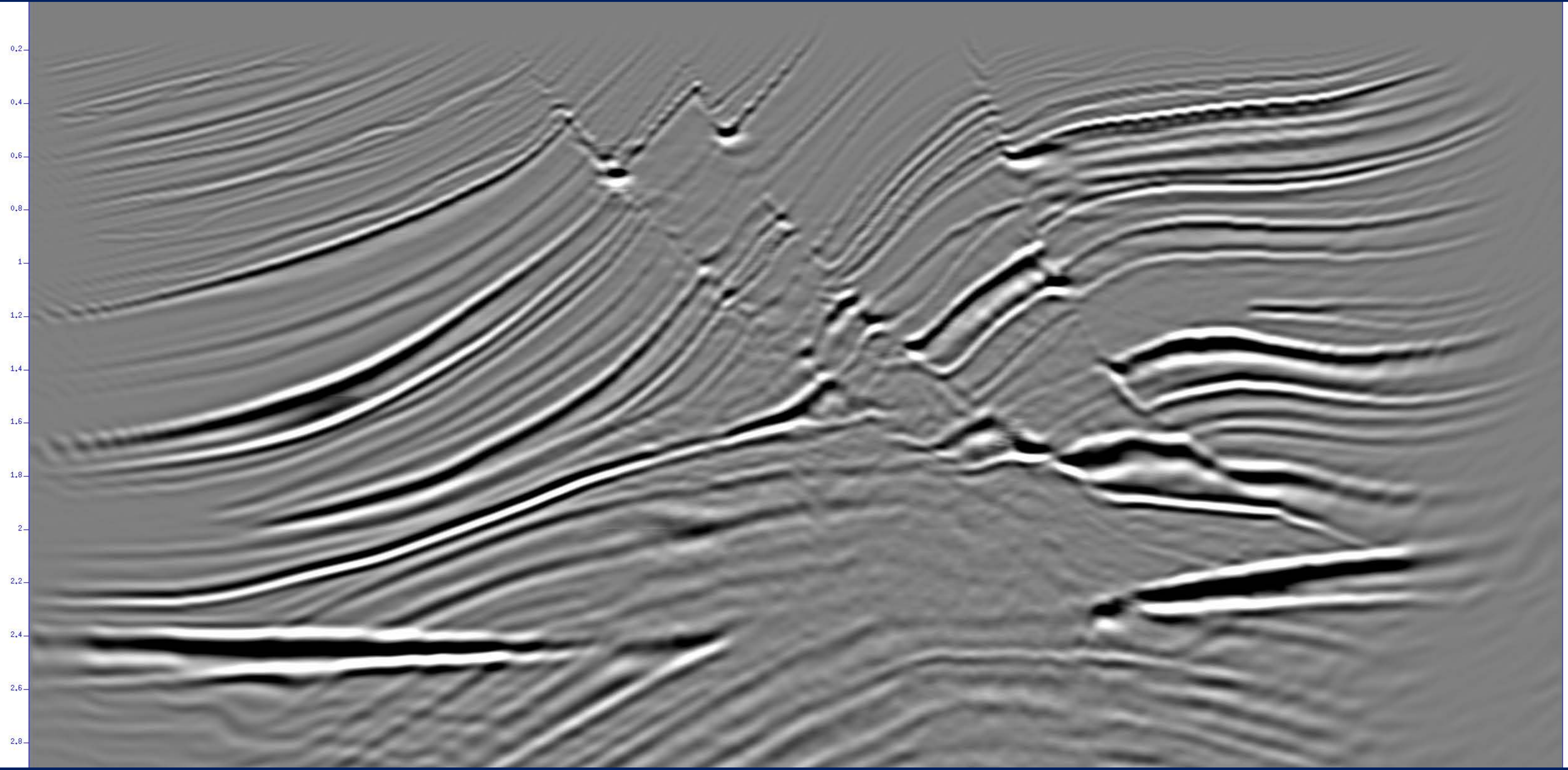
RTM with wrong operator



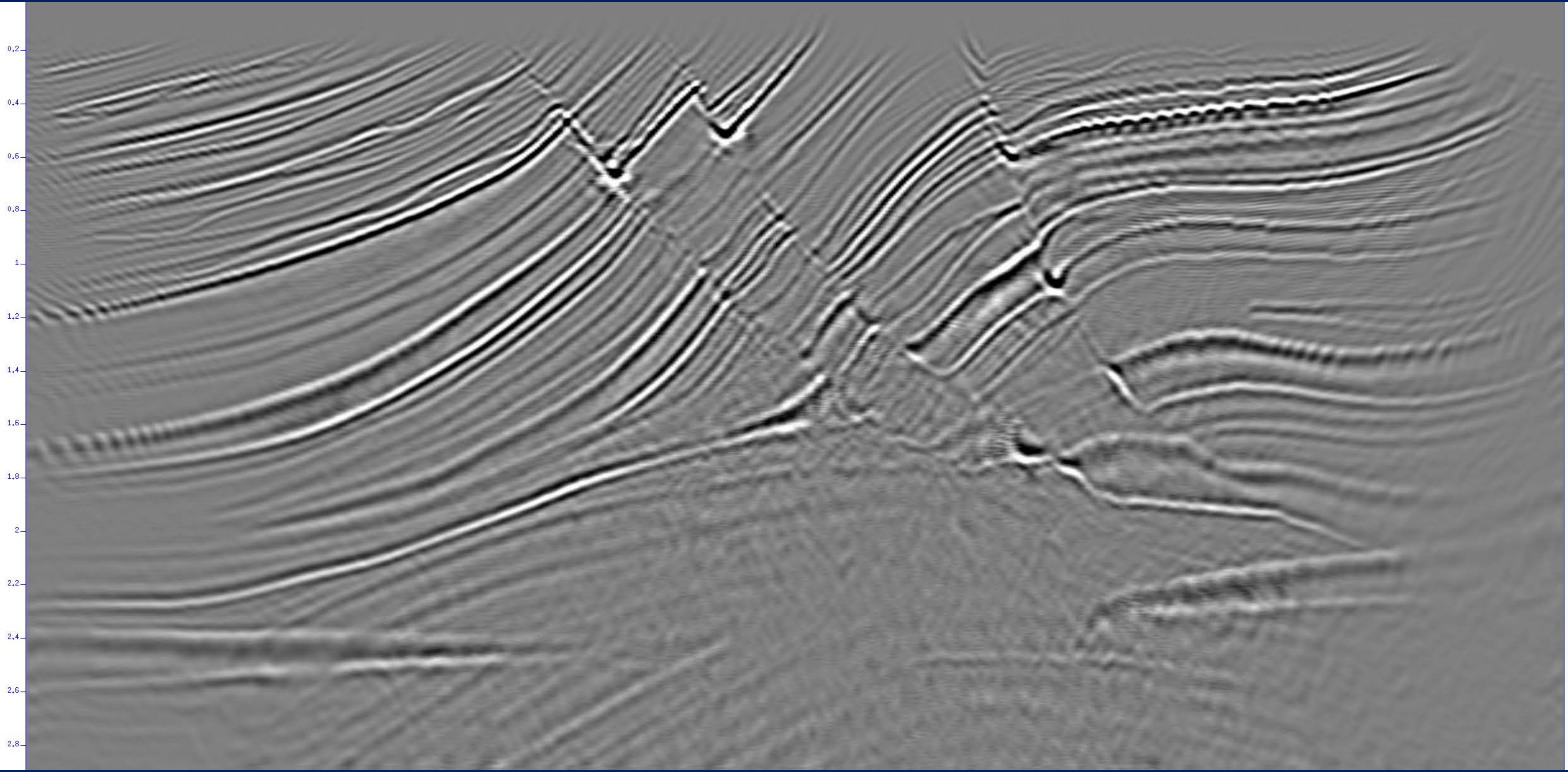
LSRTM with wrong operator



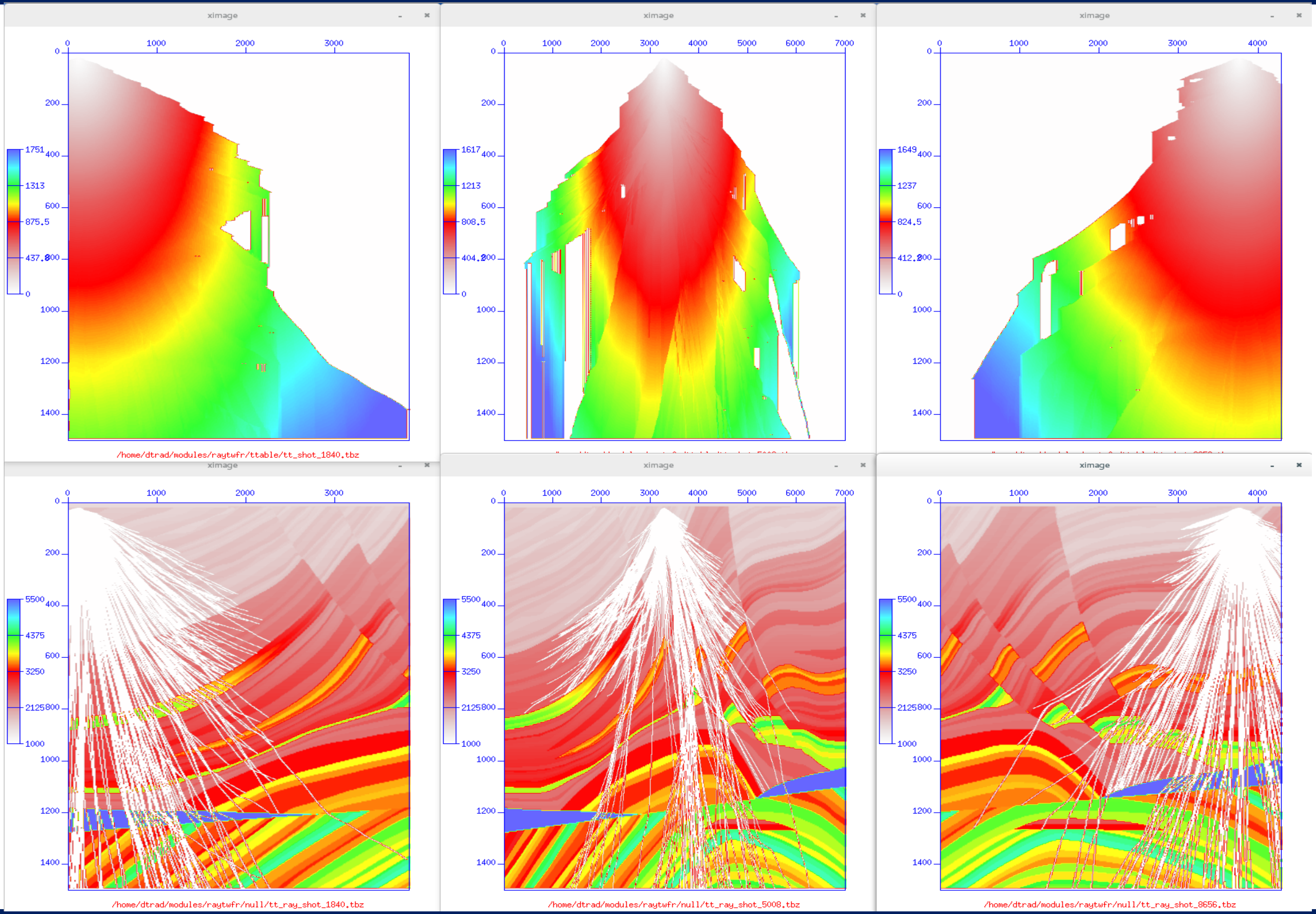
Kirchhoff Migration Marmousi



LS Kirchhoff Migration Marmousi



Traveltime tables Marmousi: multipathing and shadow zones



Improving data fitting

In general we solve expressions like $J = \left\| \mathbf{d} - \mathbf{Lm} \right\|^2$

through iterative methods that use residuals $R = \mathbf{d} - \mathbf{Lm}$

We assume that R contains the data produced by the missing parts of the model, and use a mapping from R to the model.

However, R contains two components. $R = R_1 + R_2$

$R_1 = \mathbf{d} - \mathbf{L}\Delta\mathbf{m}$ → This is the part we want for inversion

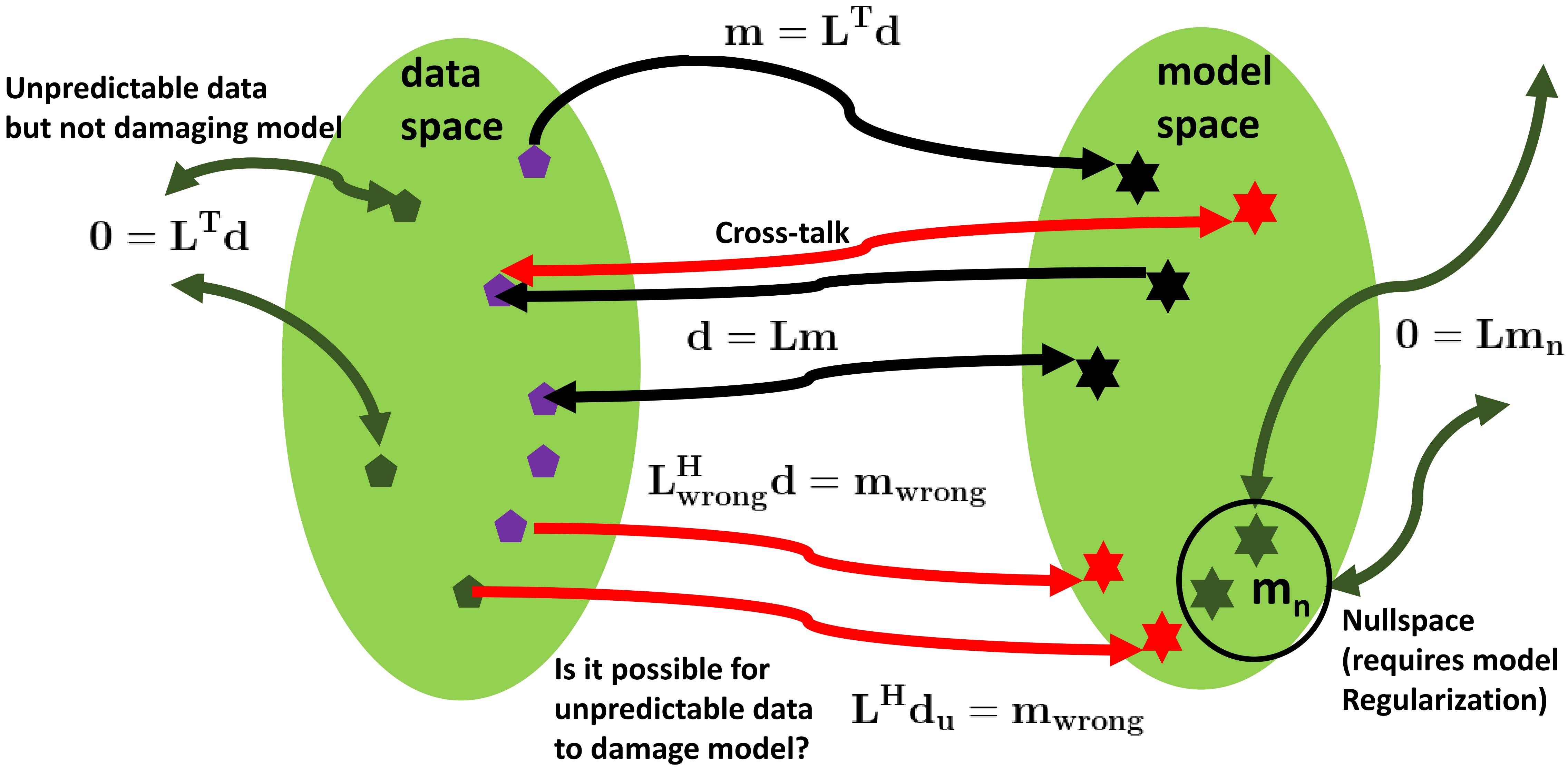
$R_2 = -\Delta\mathbf{L}\mathbf{m}$ → This is the part we don't want because it represents the error in our operator.

The "inverse crime" is helpful because it eliminates R2, but it does not work in real life. Two possible ways to deal with this:

a) To figure out ways to either decrease R2 (make our operator better), by full-physics operators (an-elastic, anisotropic)
b) To eliminate R2 from R, by distinguishing R2 from R, for example by using filtering techniques.

Currently, we try to limit the effect of R2 by stopping inversion to prevent over fitting, using global measures of error.
We need to know when to stop inversion using localized measure of error.
This requires to detect, selectively for different parts of the data space, what we can invert and what not!

Model and Data Spaces



Model and Data Spaces

data singular vectors

singular values (mapping strength)

model singular vectors

$$
\begin{bmatrix} d1 \\ d2 \\ d3 \\ d5 \\ d6 \\ d7 \\ \cdot \\ \cdot \\ \cdot \\ d_n \end{bmatrix}
=
\underbrace{\begin{bmatrix} \vec{u_1} & \vec{u_2} & \cdots & \vec{u_n} \end{bmatrix}}_{\mathbf{U}}
\begin{bmatrix}
\sigma_1 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \sigma_2 & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\
\cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \sigma_m \cdot & \cdot & \cdot & \cdot
\end{bmatrix}
\underbrace{\begin{bmatrix} \vec{v_1} \\ \vec{v_2} \\ \cdot \\ \cdot \\ \cdot \\ \vec{v_m} \end{bmatrix}}_{\mathbf{V^T}}
\begin{bmatrix} m_1 \\ m_2 \\ m_3 \\ \cdot \\ \cdot \\ \cdot \\ m_m \end{bmatrix}
$$

$$\mathbf{L} = \mathbf{USV^T}$$

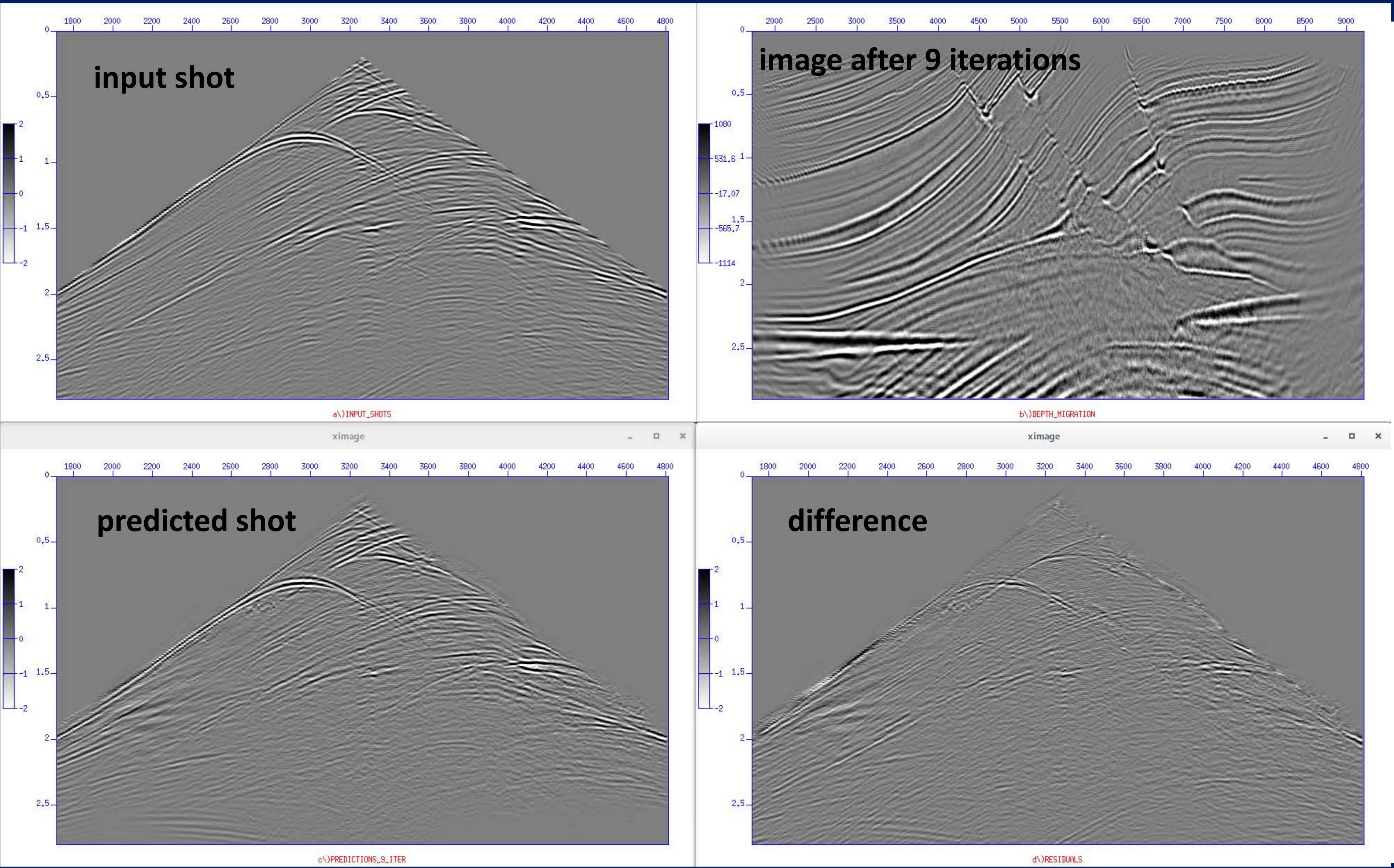
$$\mathbf{d} = \mathbf{USV^Tm}$$

$$\mathbf{m} = \sum_{\mathbf{i=1}}^{\mathbf{p}} \frac{\mathbf{u^Td}}{\sigma_\mathbf{i}} \mathbf{v_i}$$

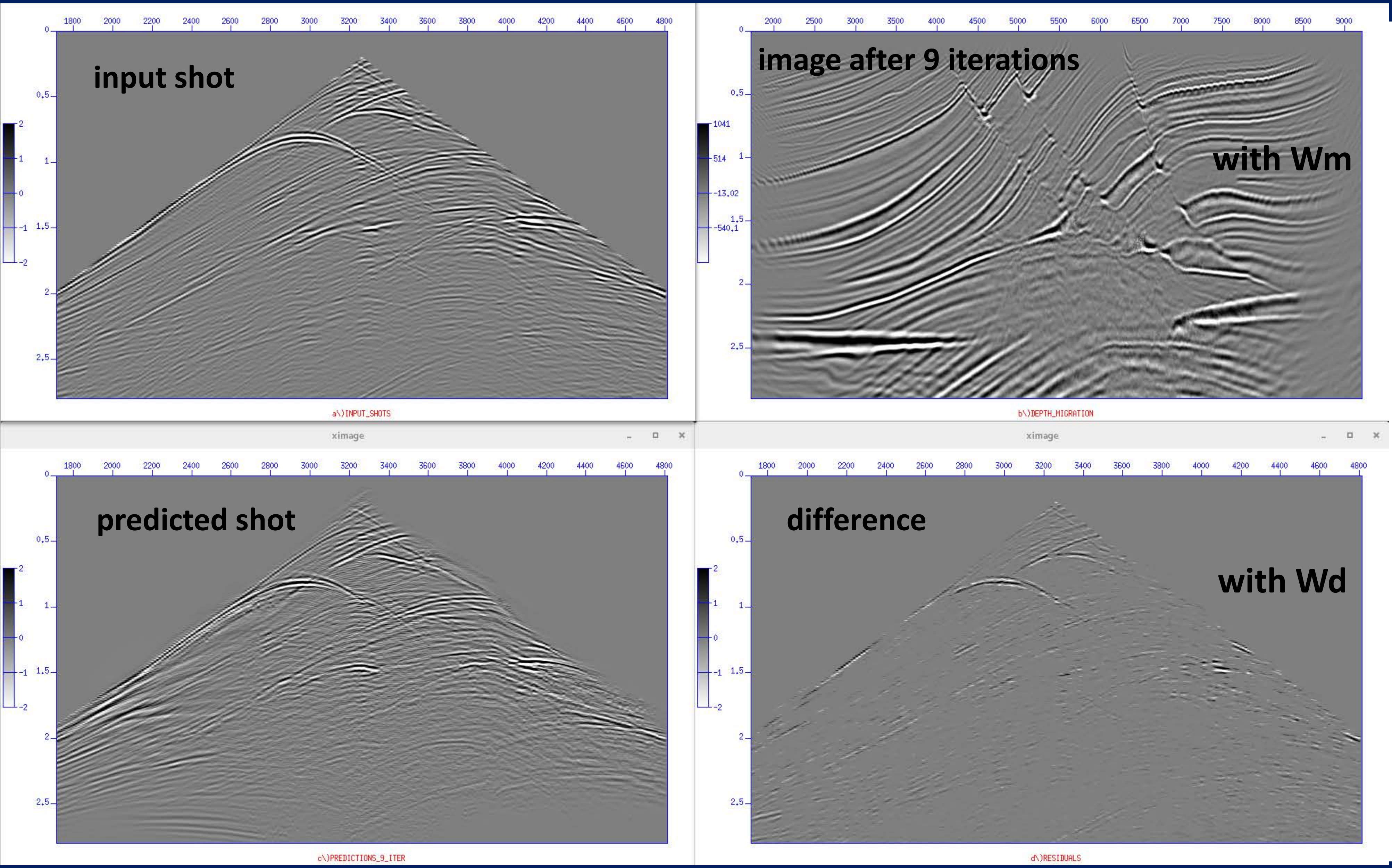
$$\mathbf{m_{adj}} = \sum_{\mathbf{i=1}}^{\mathbf{p}} (\mathbf{u^Td}) \sigma_\mathbf{i} \mathbf{v_i}$$

Inversion: delicate balance between data-U cross-correlation and mapping.

Adjoint: robust balance with wrong amplitude.

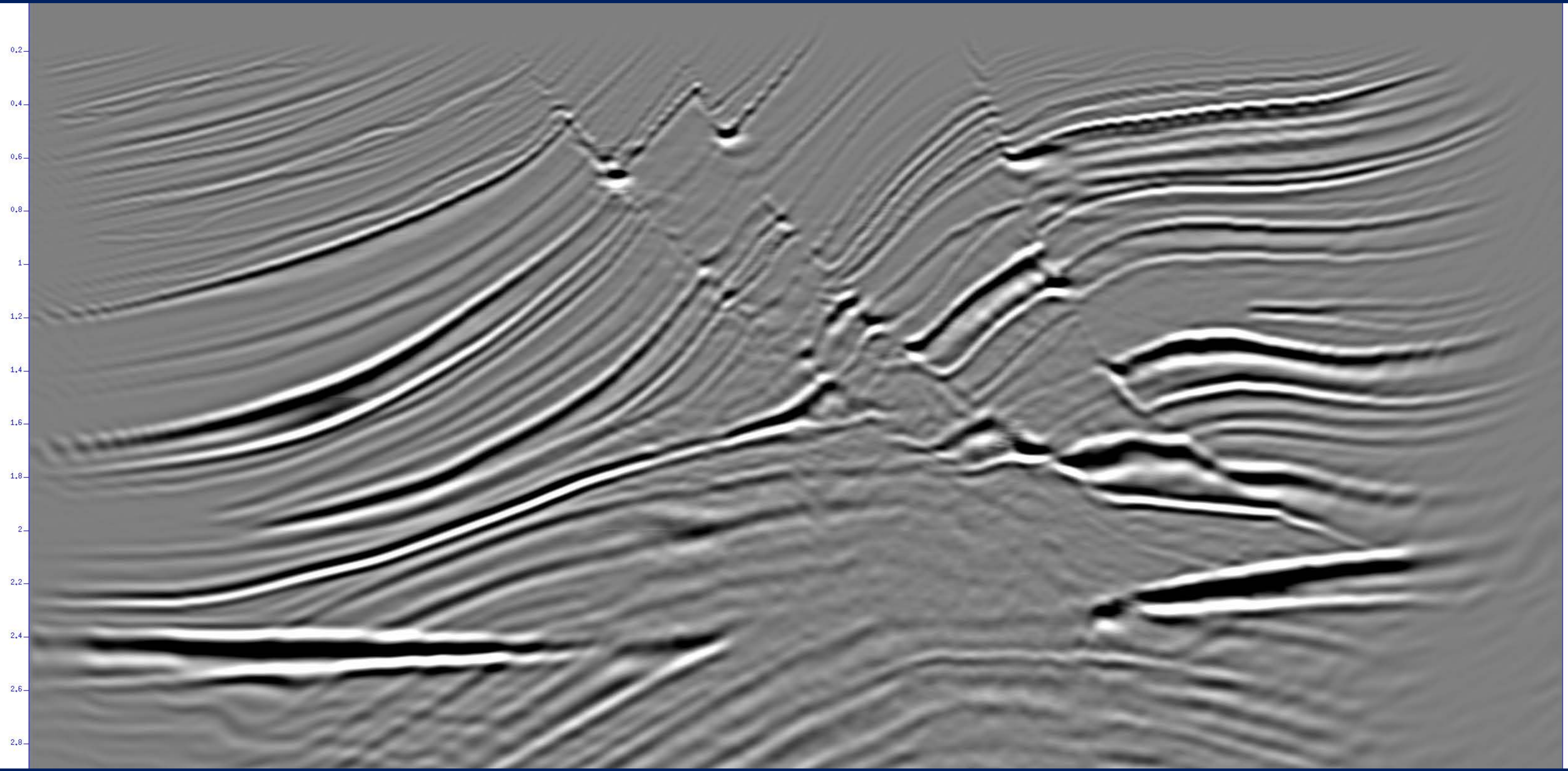


Many parts of the data are not yet predicted

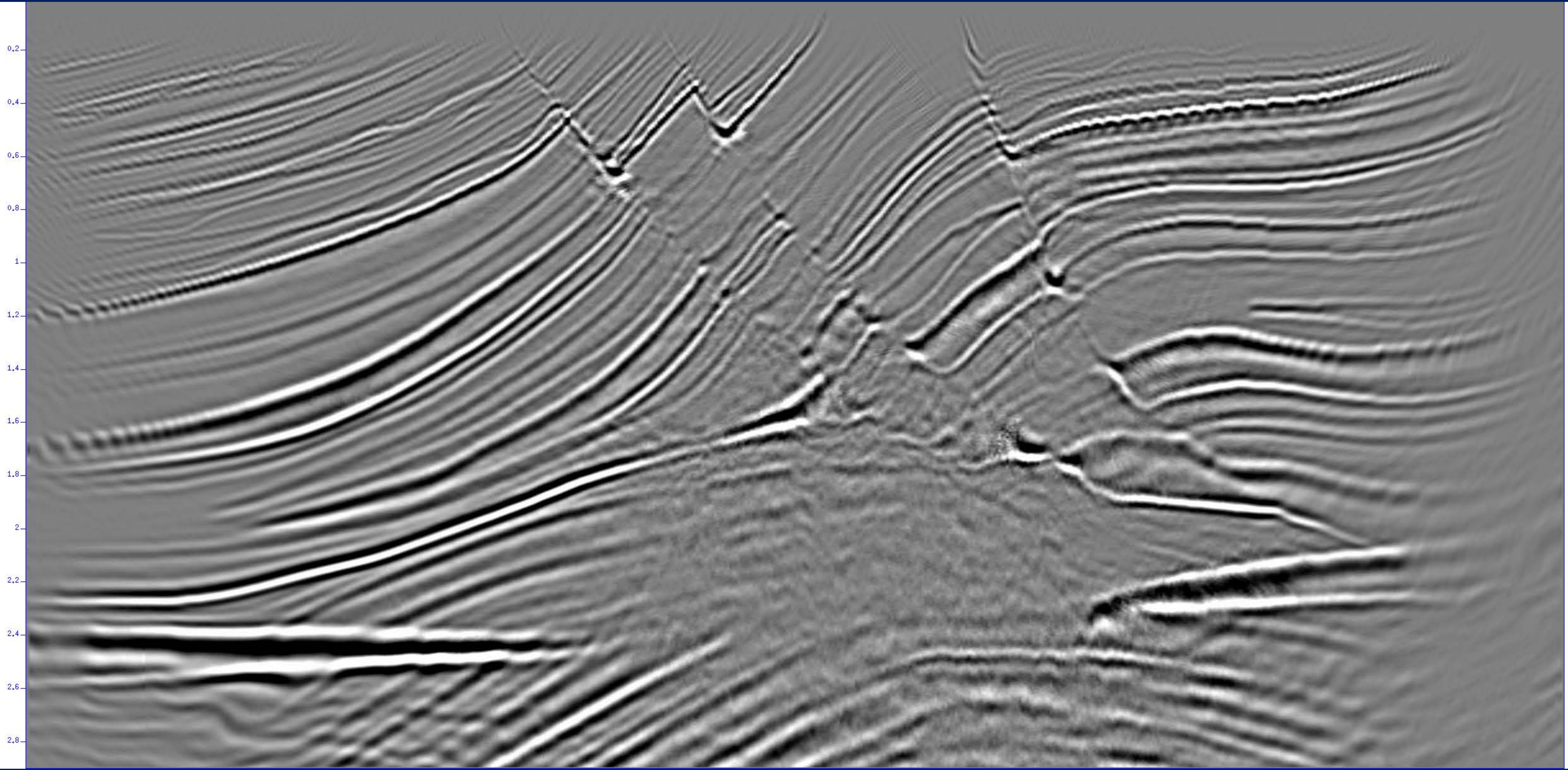


zeroed parts have increasing residuals

Kirchhoff Migration Marmousi



LS Kirchhoff Migration Marmousi



Conclusions

- Noise accumulates because inconsistencies between operator and physics
- In Kirchhoff algorithm inversion noise is more obvious than in RTM
- Often this is hidden if the data fit the operator, instead of the reverse
- Noise control can be achieved by:
  - designing better approximations to physics, either by design and/or optimization
  - filtering mapping errors from model space and data space.
  - eliminating from residuals components we can’t predict.

Acknowledgements

Sam Gray

Seismic Unix and Madagascar

CSEG

CREWES sponsors

NSERC