Lecture 5
Migration as inverse problem

M D Sacchi
UNLP Inverse Problems

Forward and Adjoint Operators

- **Two special operations (or operators?)**

$$\textbf{Forward}: \mathbf{d} = \mathbf{L}\mathbf{m}$$

$$\textbf{Conjugate transpose or adjoint}: \tilde{\mathbf{m}} = \mathbf{L}'\mathbf{d}$$

- $\mathbf{L}$ is a linear operator (Forward modelling operator).

- Why are them special?

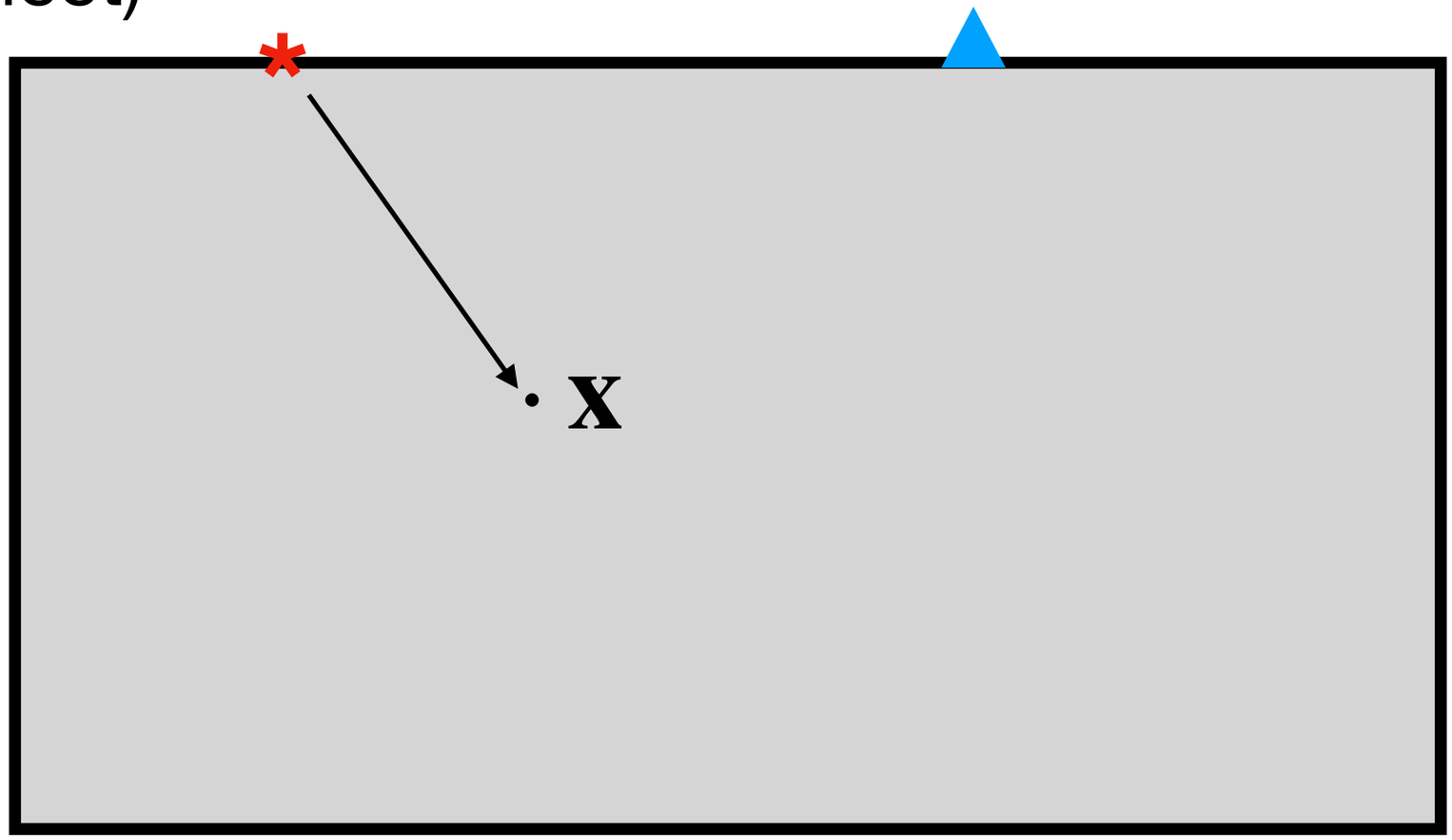
  - Iterative solvers for large inverse problems only need to know how to evaluate $\mathbf{L}[\,\cdot\,]$ and $\mathbf{L}'[\,\cdot\,]$

  - I usually interpret operators as matrices. In reality, operators are codes applied *on the flight*

  - *We will see these operators everywhere today*

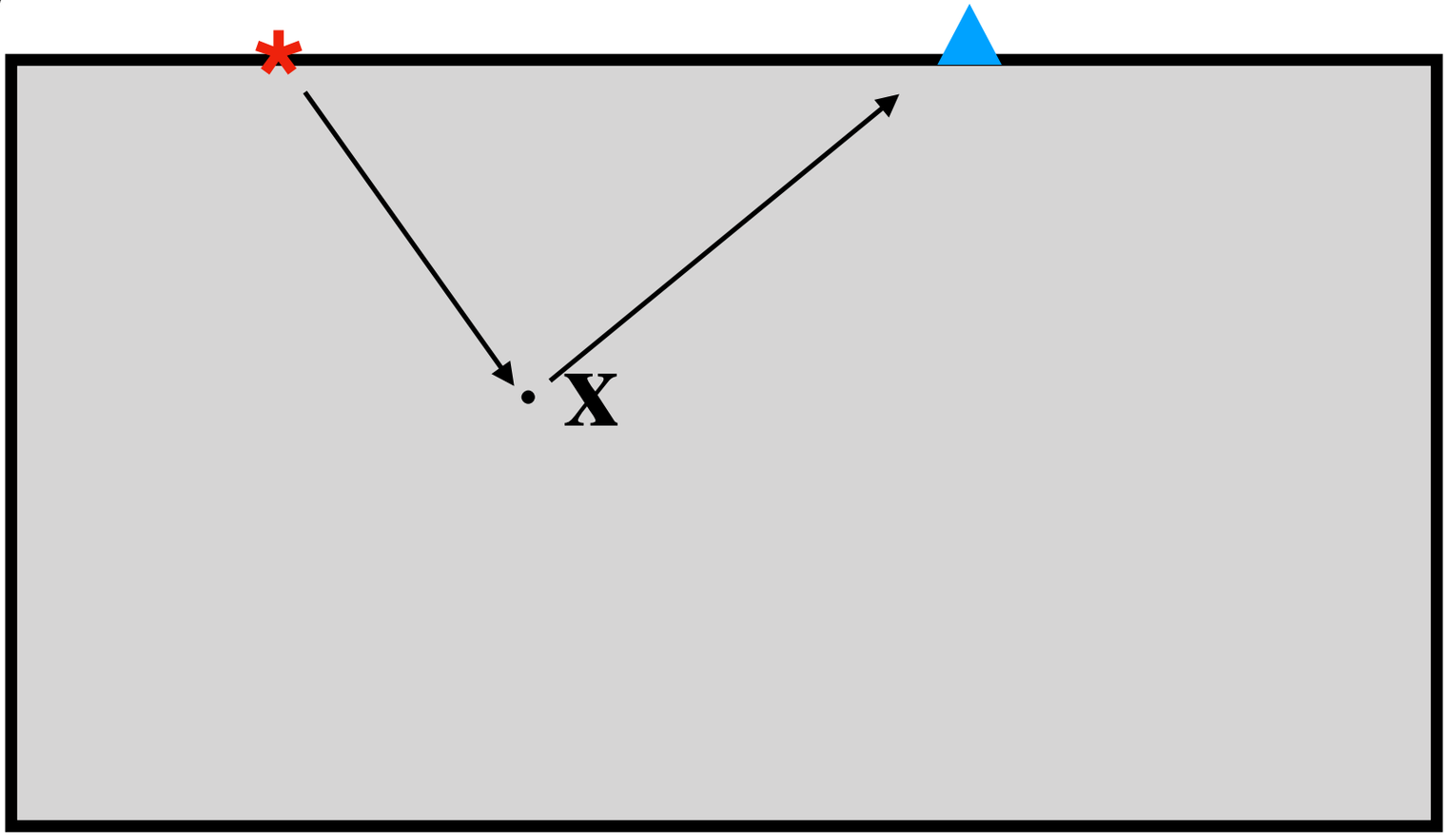
Understanding migration with simple concepts

- No reflection in point **x** (the dot does not exist, wave does not reflect)



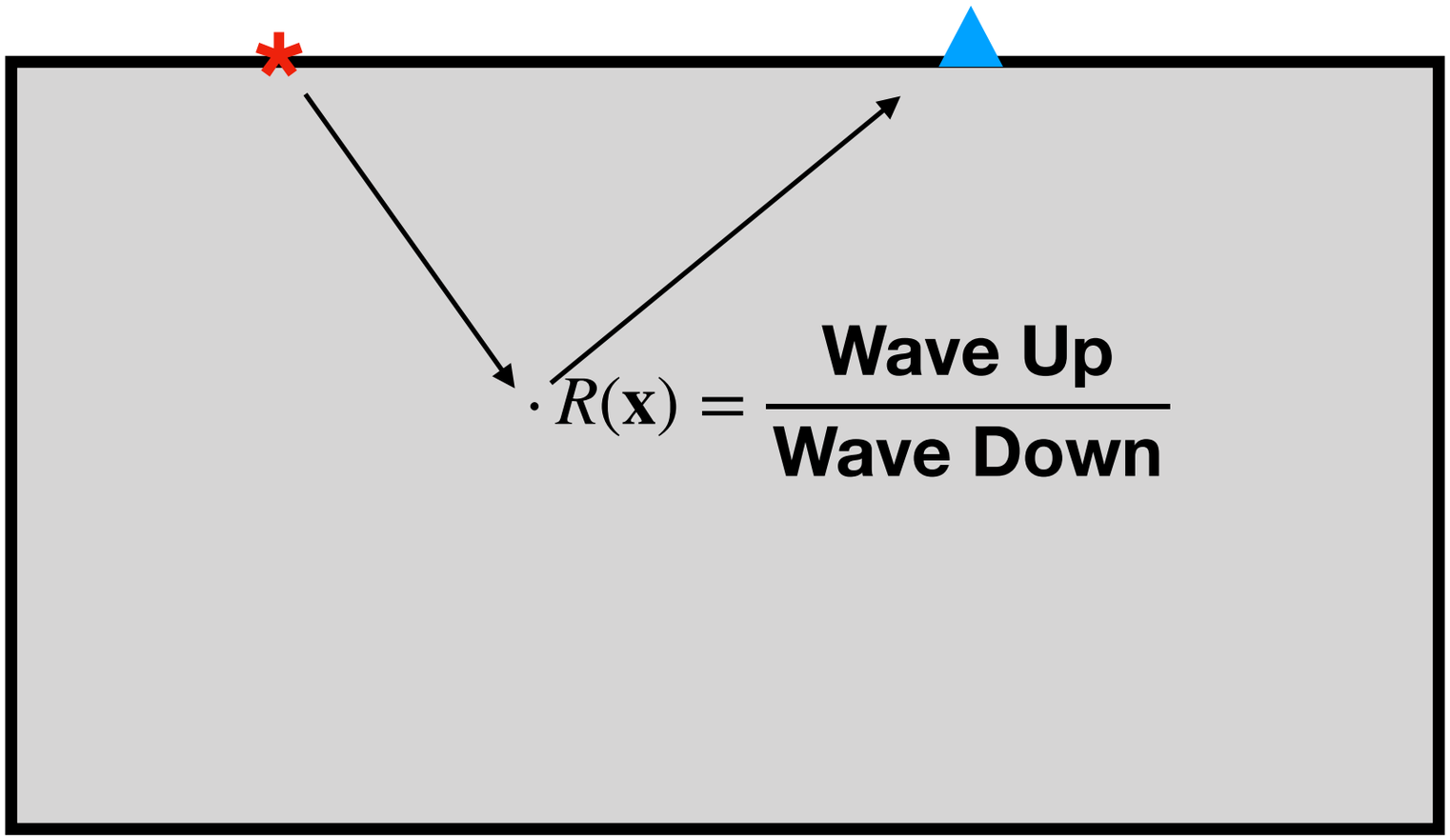
Understanding migration with simple concepts

- Reflection in point $\mathbf{x}$ (the dot does exist, wave does reflect)



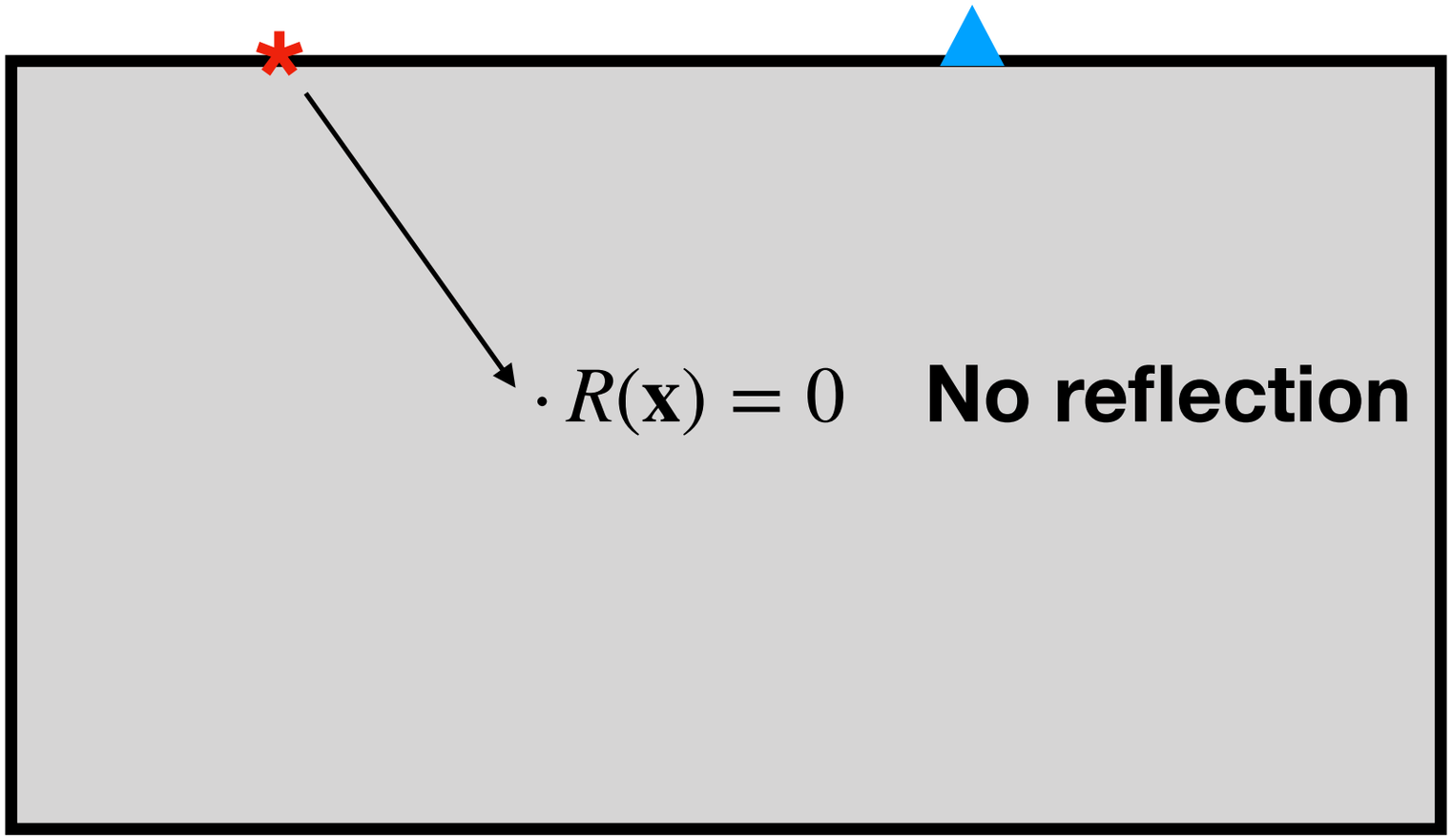
Understanding migration with simple concepts

- Put a probing function in $\mathbf{x}$



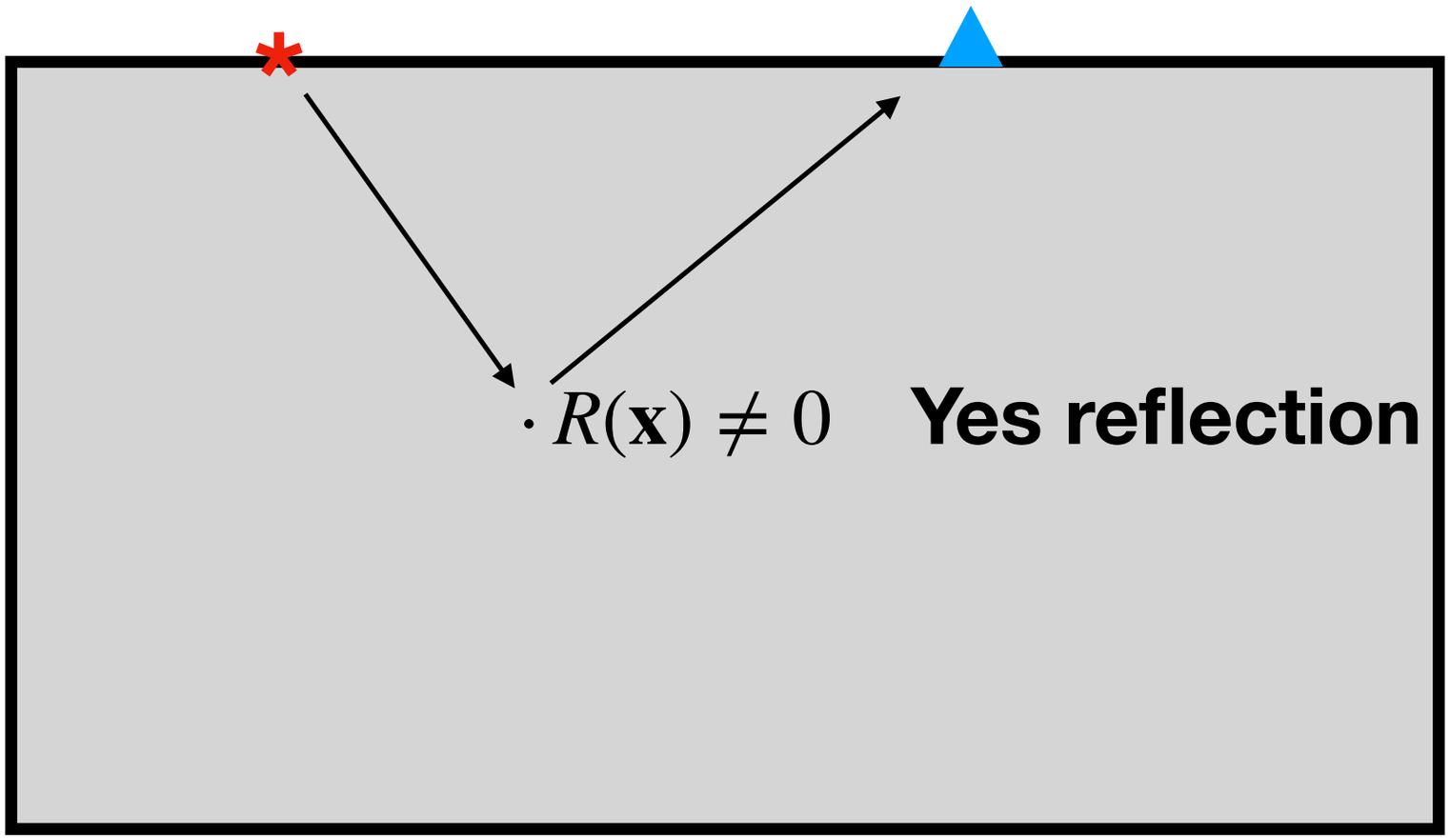
Understanding migration with simple concepts

- Put a probing function in $\mathbf{x}$



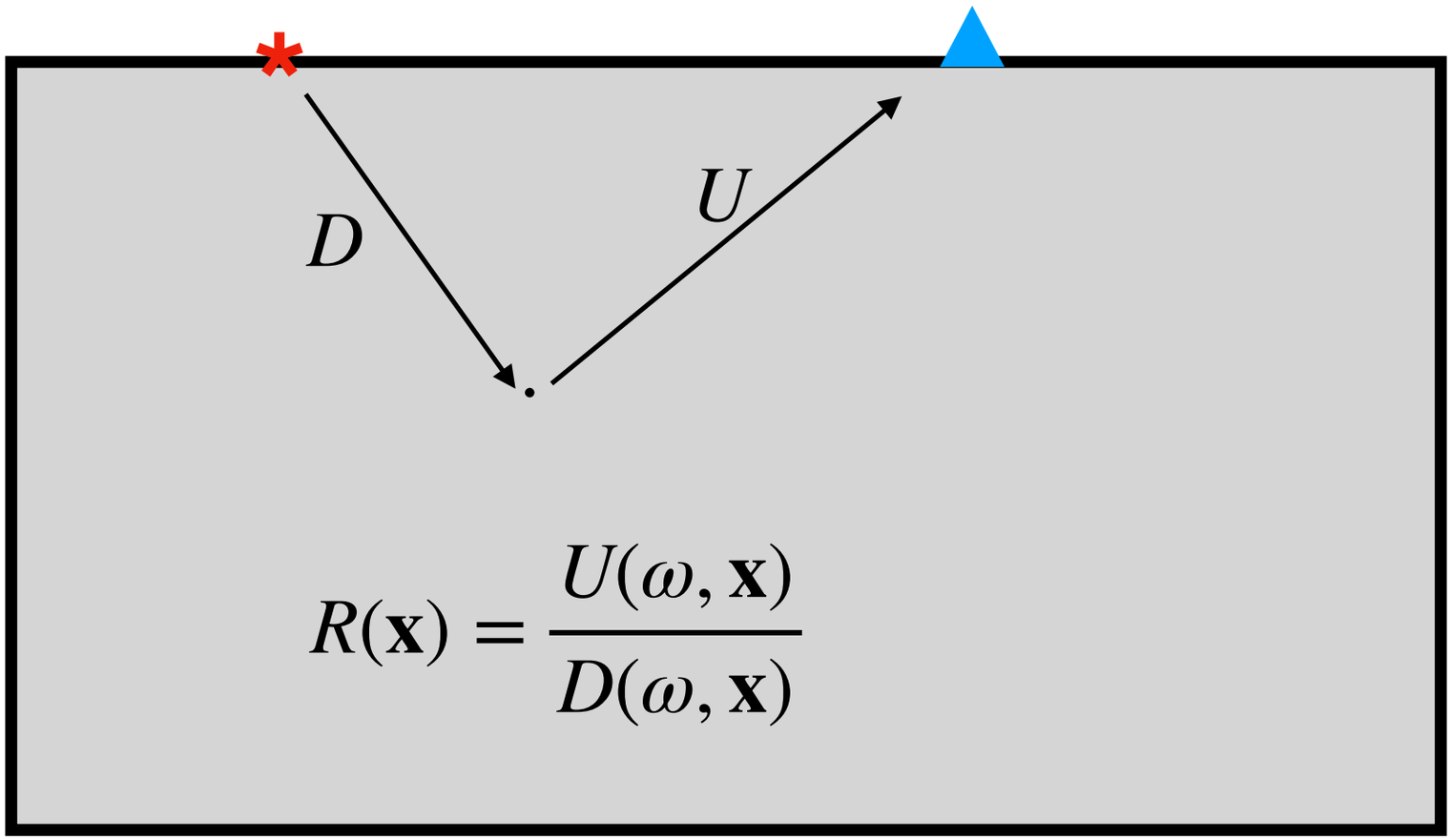
Understanding migration with simple concepts

- Put a probing function in $\mathbf{x}$



Understanding migration with simple concepts

- Let's define the probing function: "Reflectivity"



$$R(\mathbf{x}) = \frac{U(\omega, \mathbf{x})}{D(\omega, \mathbf{x})}$$

Prestack Migration

- We need to compute $\mathrm{R}(\mathbf{x})$ everywhere
- How do we do it?
- Do you remember Potential Field Methods? Upward and Downward Continuation
- Similar … but with the wave equation

Pre-stack migration

- At any point $\mathbf{x}$ by the definition of the probing function

$$U(\omega, \mathbf{x}) = D(\omega, \mathbf{x})\, R(\mathbf{x})$$

- At the surface

$$U(\omega, x = r, z = 0) = \mathbf{\Psi}\, \mathbf{P}_u U(\omega, \mathbf{x})$$

- Down-going field can be simulated by forward propagation

$$D(\omega, \mathbf{x}) \; = \mathbf{P}_d\, S(\omega, x = x_s, z = 0)$$

Pre-stack migration

Propagator

Model m

$$U(\omega, x = r, z = 0) = \boldsymbol{\Psi}\, \mathbf{P}_u U(\omega, \mathbf{x})$$

$$U(\omega, \mathbf{x}) = D(\omega, \mathbf{x})\, R(\mathbf{x})$$

Data d

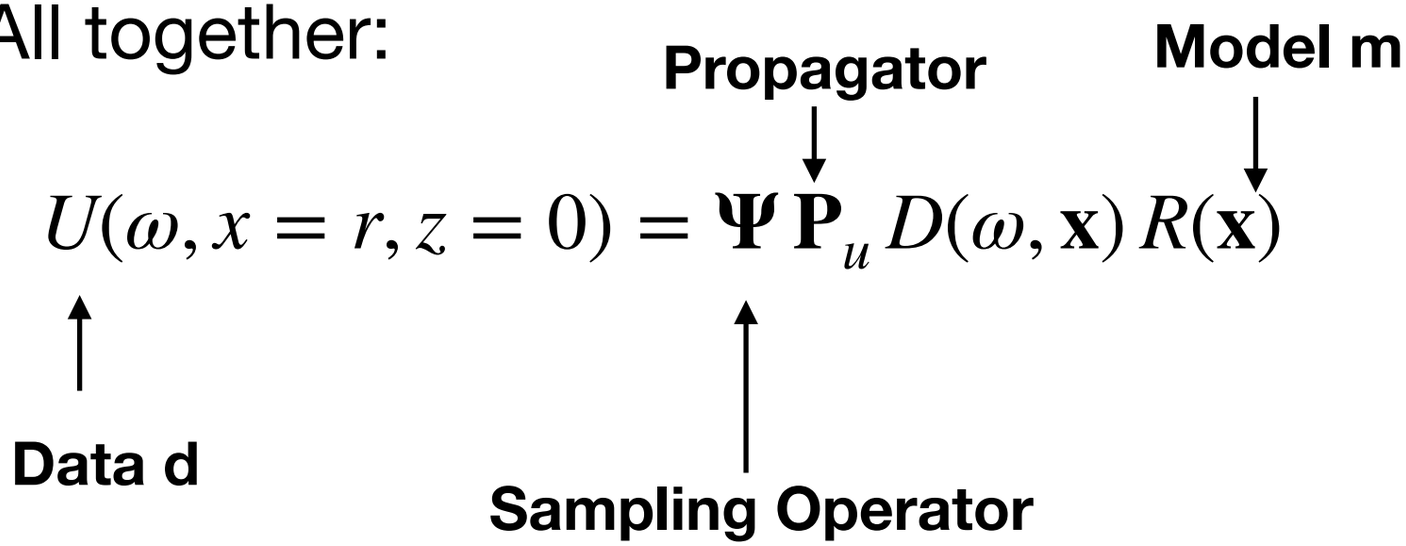
Sampling Operator

Propagator

With $D(\omega, \mathbf{x}) = \mathbf{P}_d\, S(\omega, x = x_s, z = 0)$ compute by Wavefield forward modelling

Pre-stack migration

All together:



..and finally:

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

Prestack Migration

- Clearly one has more than one source

$$\mathbf{d}_s = \mathbf{L}_s \mathbf{m}, \qquad s = 1 \ldots n_s$$

- Migration

$$\tilde{\mathbf{m}} = \sum_s \mathbf{L}'_s \mathbf{d}_s$$

- Were the operators can be synthesized as codes (not as matrices) via the procedure I abstractly denominate the **Do_It** code

- Use whiteboard to show summing / spraying trick!

Consider one source

- Two canonical operators for migration

$$\mathbf{d}_s = \mathbf{L}_s\mathbf{m}$$ **De-migration (Forward)**

$$\tilde{\mathbf{m}} = \mathbf{L}_s'\mathbf{d}_s$$ **Migration (Adjoint)**

- Least-squares migration (use iterative inversion to minimize)

$$J = \|\mathbf{d}_s - \mathbf{L}_s\mathbf{m}\|_2^2$$

Consider more sources

- Two canonical operators for migration

$$\mathbf{d}_s = \mathbf{L}_s\mathbf{m}$$ **De-migration (Forward)**

$$\tilde{\mathbf{m}} = \sum_s \mathbf{L}'_s\mathbf{d}_s$$ **Migration (Adjoint)**

- Least-squares migration (use iterative inversion to minimize)

$$J = \sum_s \|\mathbf{d}_s - \mathbf{L}_s\mathbf{m}\|_2^2$$

Pre-stack migration and imaging condition

$$U(\omega, x = r, z = 0) = \mathbf{\Psi}\,\mathbf{P}_u\, D(\omega, \mathbf{x})\, R(\mathbf{x}) \longrightarrow \mathbf{d}_s = \mathbf{L}_s\mathbf{m}$$

$$\tilde{R}(\mathbf{x}) = D^*(\omega, \mathbf{x})\mathbf{P}'_u\mathbf{\Psi}'U(\omega, x = r, z = 0) \longleftarrow \tilde{\mathbf{m}} = \mathbf{L}'_s\mathbf{d}_s$$

Famous Cross-correlation Imaging Condition

$$\tilde{R}(\mathbf{x}) = D^*(\omega, \mathbf{x})U(\omega, \mathbf{x})$$

Pre-stack migration and imaging condition

$$R(\mathbf{x}) = \frac{U(\omega, \mathbf{x})}{D(\omega, \mathbf{x})}$$ **Proposed probing function**

$$\tilde{R}(\mathbf{x}) = D^*(\omega, \mathbf{x})U(\omega, \mathbf{x})$$ **Imaging Condition (Classical Migration)**

$$R(\mathbf{x}) = \frac{U(\omega, \mathbf{x})}{|D(\omega, \mathbf{x})|^2} \cdot D^*(\omega, \mathbf{x}) \approx c \cdot D^*(\omega, \mathbf{x})U(\omega, \mathbf{x}) = c \cdot \tilde{R}(\mathbf{x})$$

Migration vs LS - migration

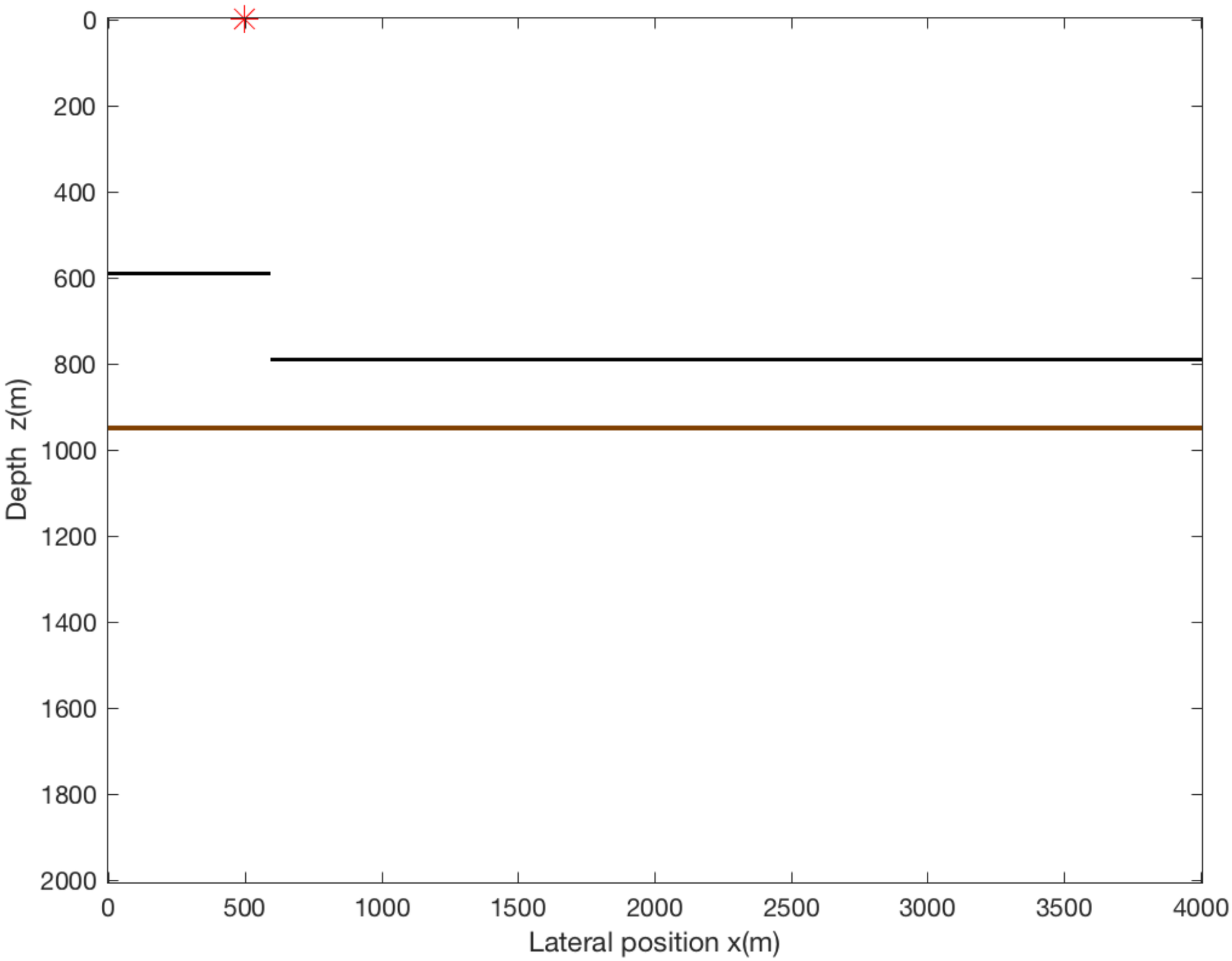
- Migration: Blurred version of $\mathbf{m}$ via adjoint
- LS - Migration: Attempts to retrieve $\mathbf{m}$ by solving an inverse problem
- What are those propagators?
  - One-way wave equation migration and LS migration: Propagators are synthesized in f-k domain (Gazdag, PSPI, Split-step etc)
  - Two-way wave equation and two-way wave equation LS migration: Synthesized by Finite Differences (Adjoint is often called RTM -> Reverse Time Migration)

Example

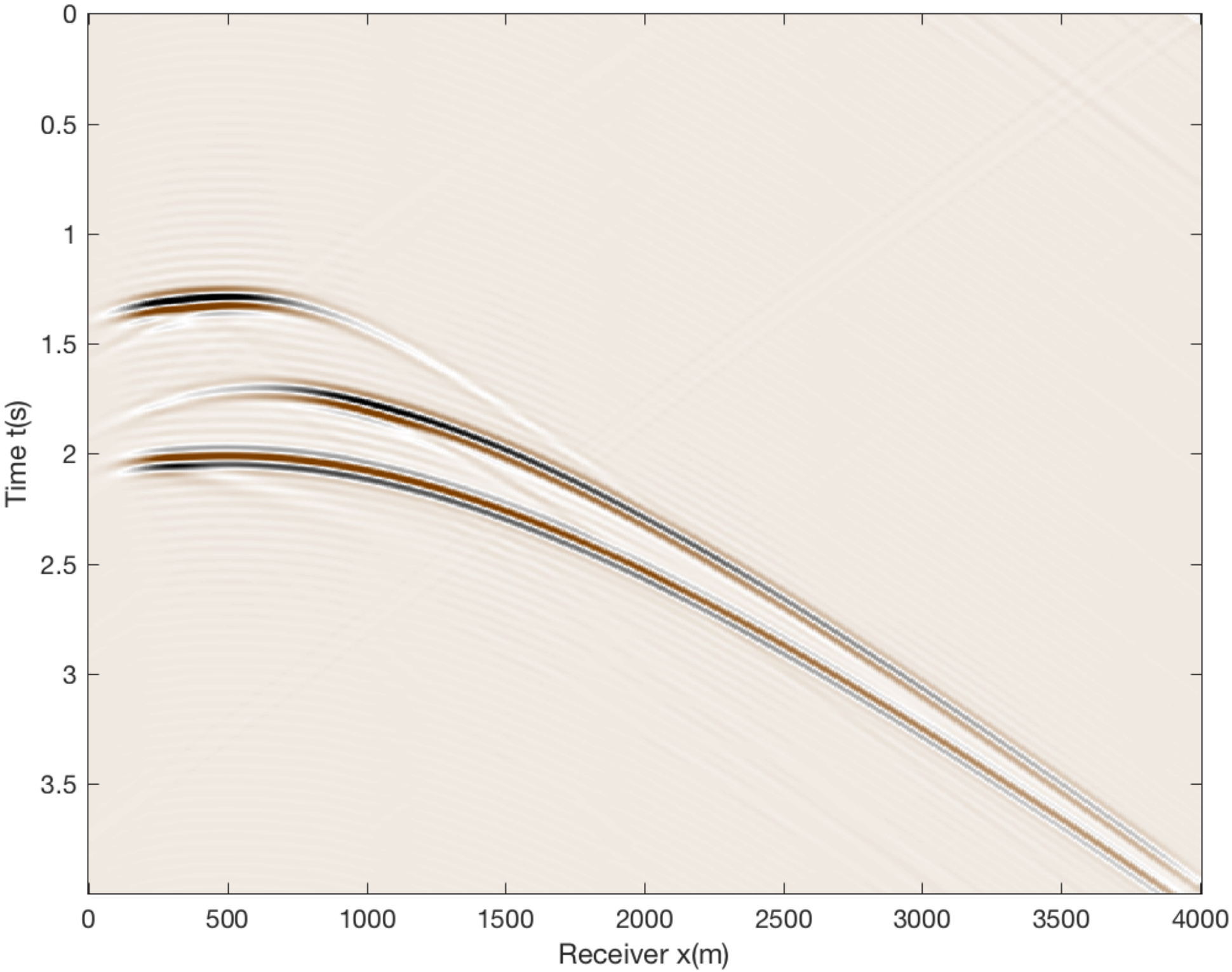
- Dot-product test gets tedious ! (code by Kevin Cheng)

```
m1 = randn(nz,nx,ns);
d2 = randn(nt,nx,ns);

  [d1] = kvssmfor(m1,vel,sx,fpeak,tshift,nt,dt,dx,dz,f1,f2);
  [m2] = kvssmadj(d2,vel,sx,fpeak,tshift,nt,dt,dx,dz,f1,f2);

dot1 = sum(sum(sum(d1.*d2)))
dot2 = sum(sum(sum(m1.*m2)))
```

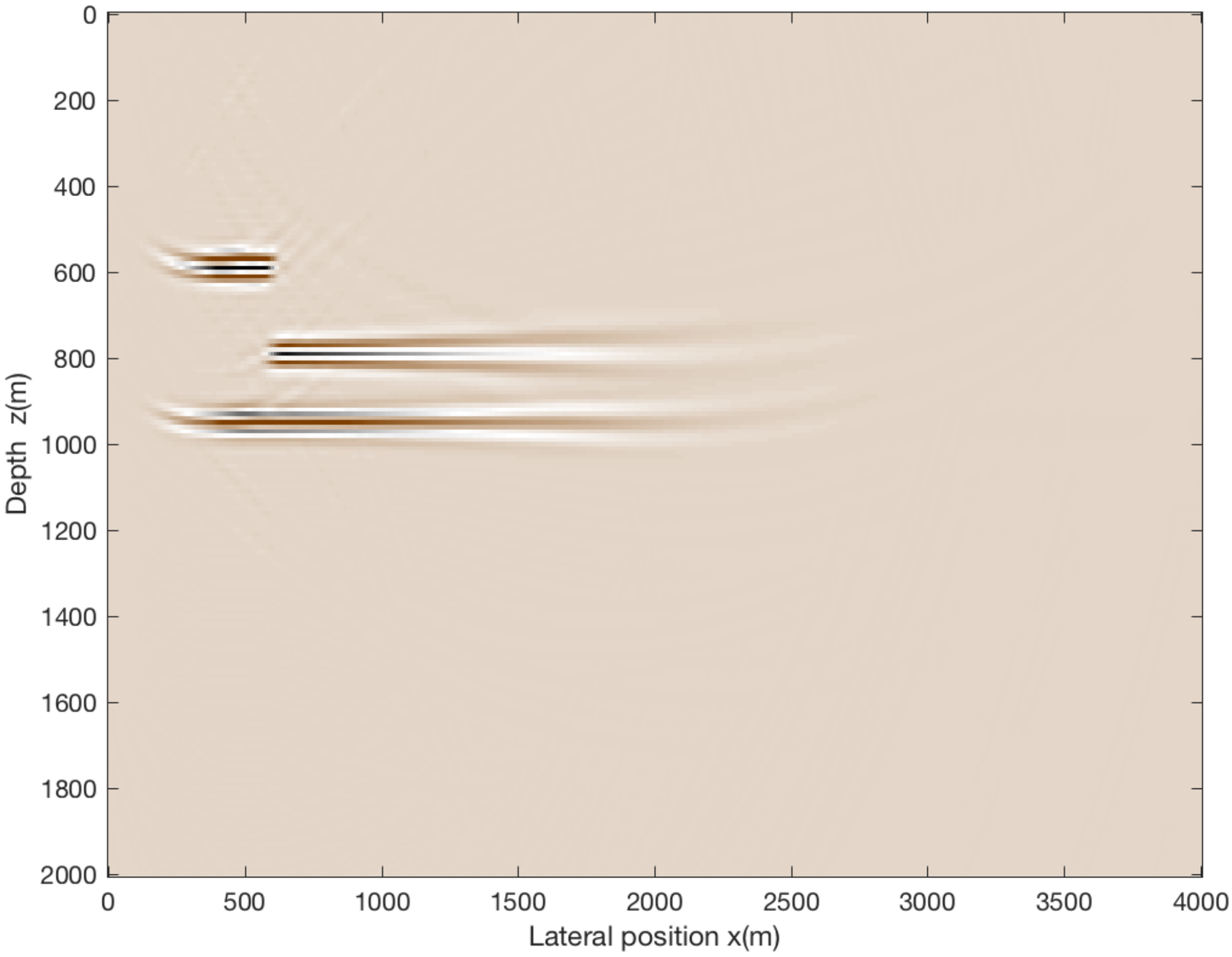
True Structure



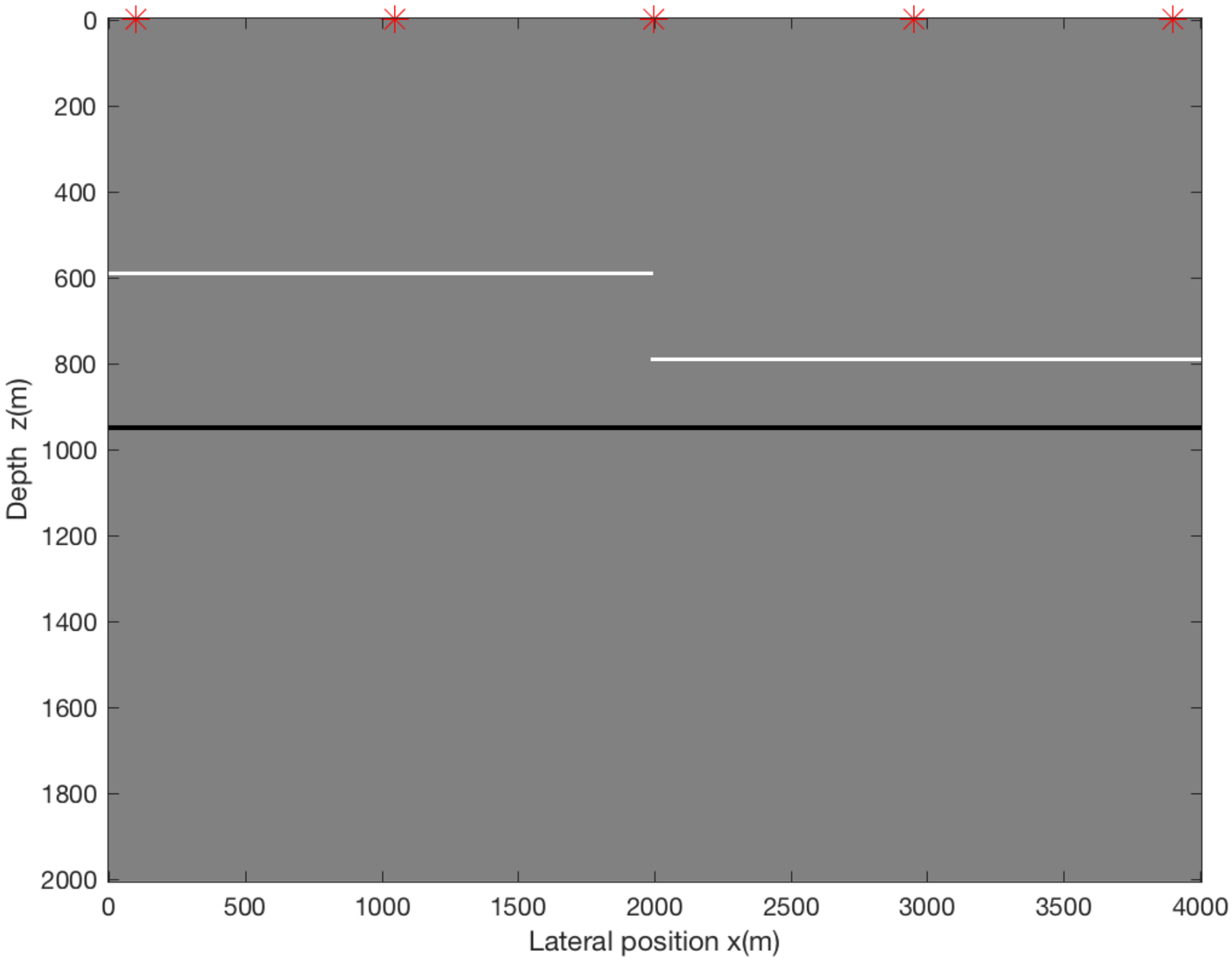
Shot


0
0.5
1
1.5
2
2.5
3
3.5
Time t(s)
0
500
1000
1500
2000
2500
3000
3500
4000
Receiver x(m)

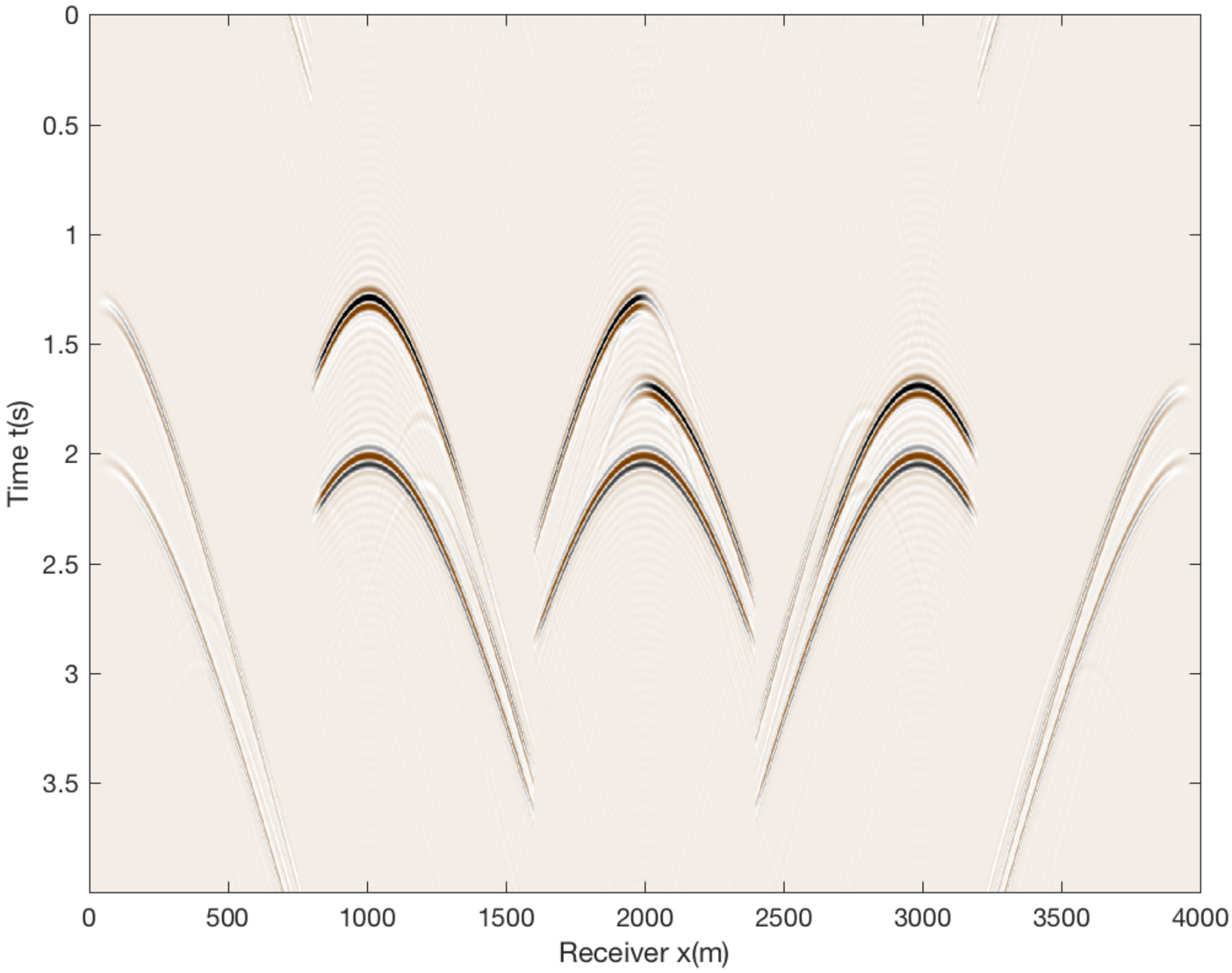
Pre-stack Migration (Adjoint)



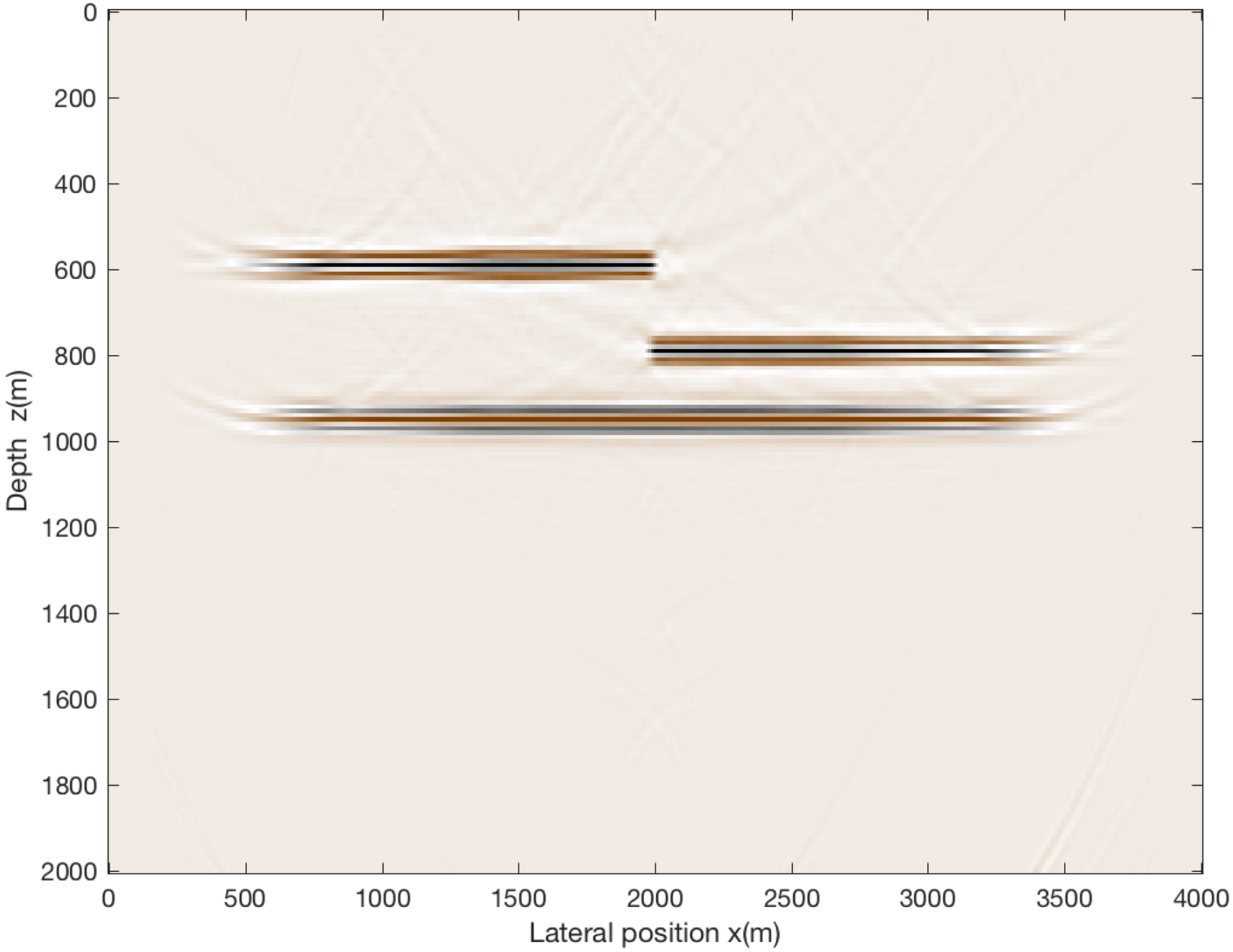
5 sources


0
200
400
600
800
1000
1200
1400
1600
1800
2000
Depth z(m)
0
500
1000
1500
2000
2500
3000
3500
4000
Lateral position x(m)

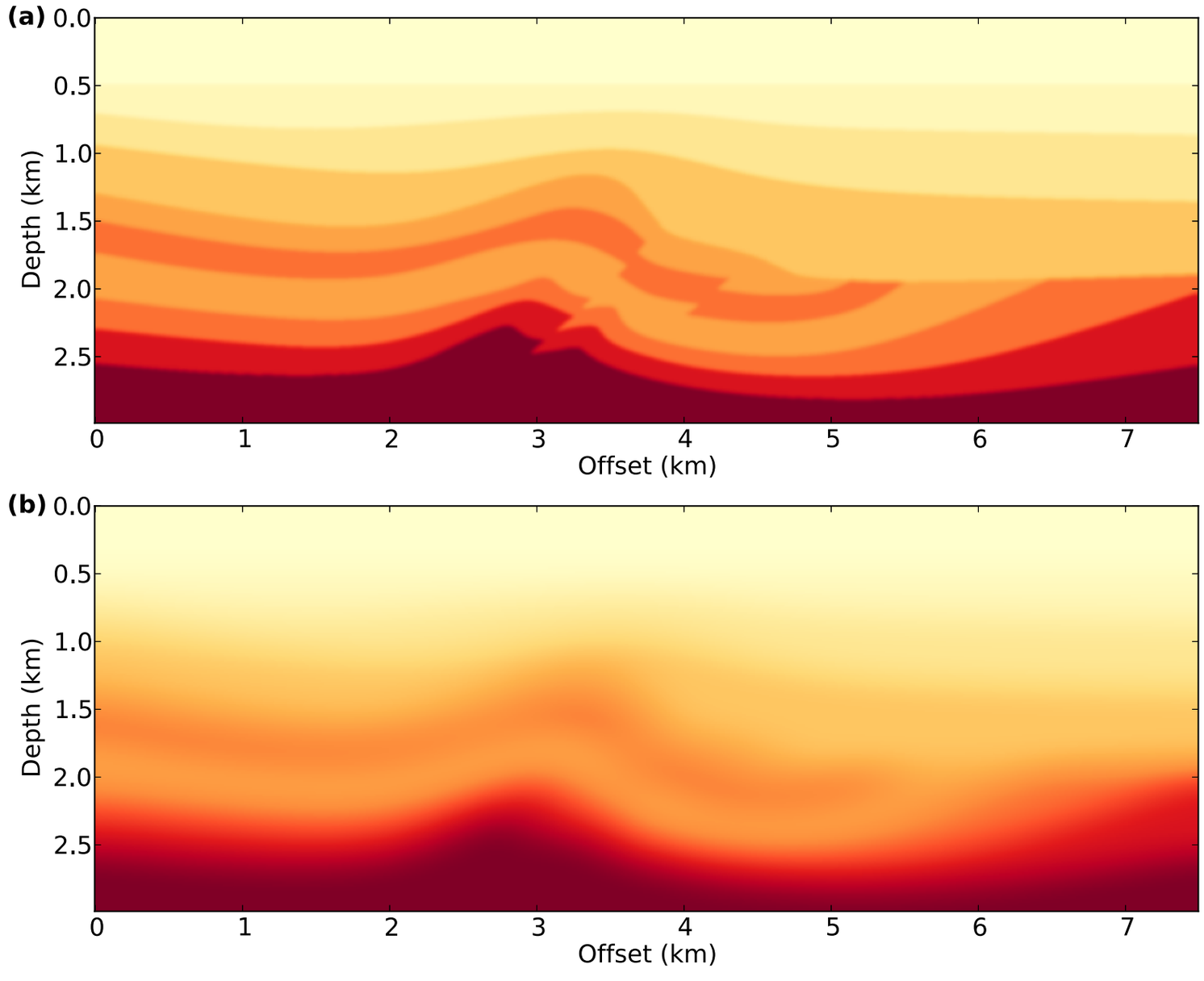
Pre-stack data for 5 sources



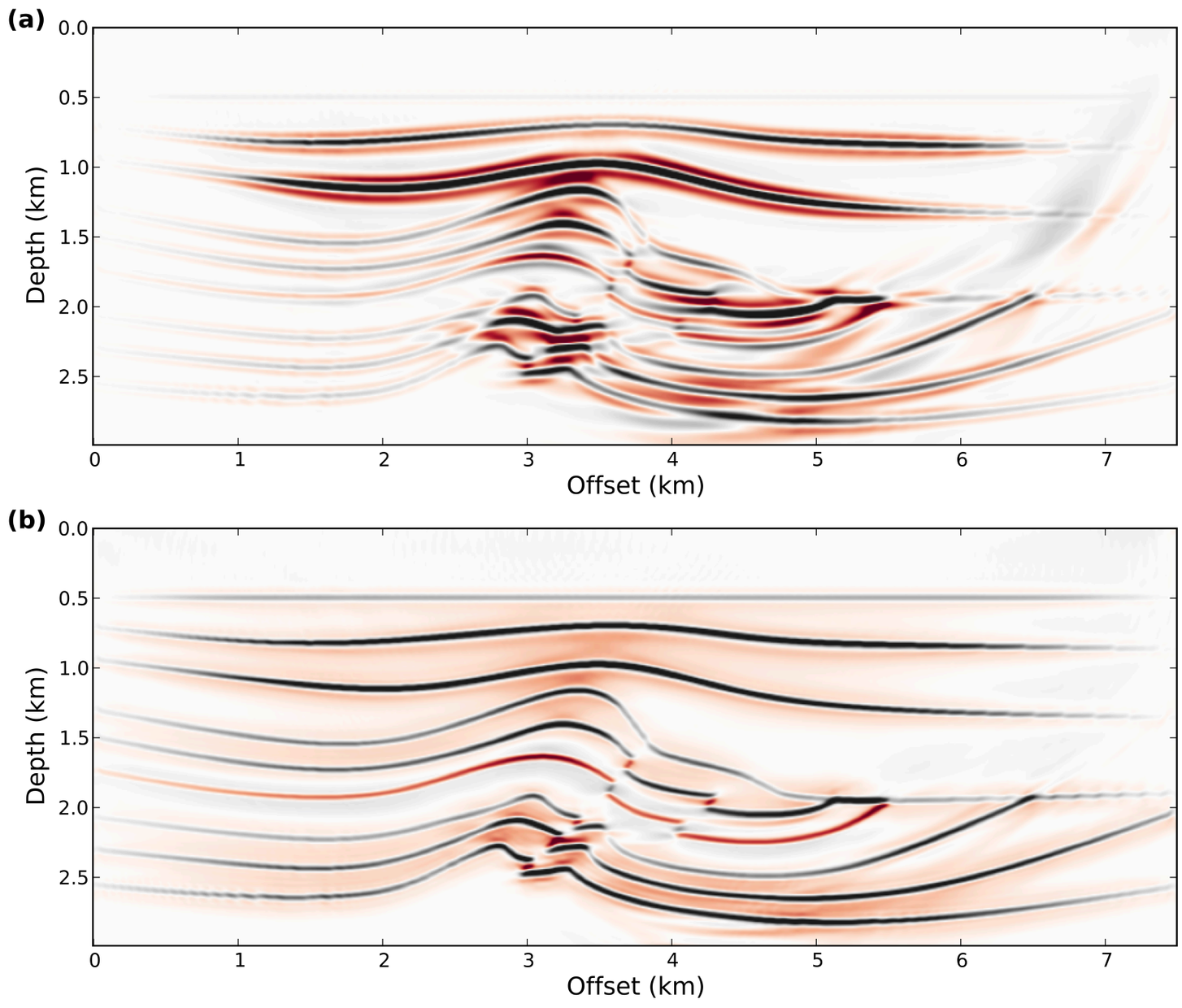
Pre-stack Migration (Adjoint)



Migration and LS Migration with two-way wave equation operators (Linan Xu and Sacchi, GEO, 2018)



Migration (top) and LS Migration (bottom)



Smoothing and LS migration

Rather than minimizing

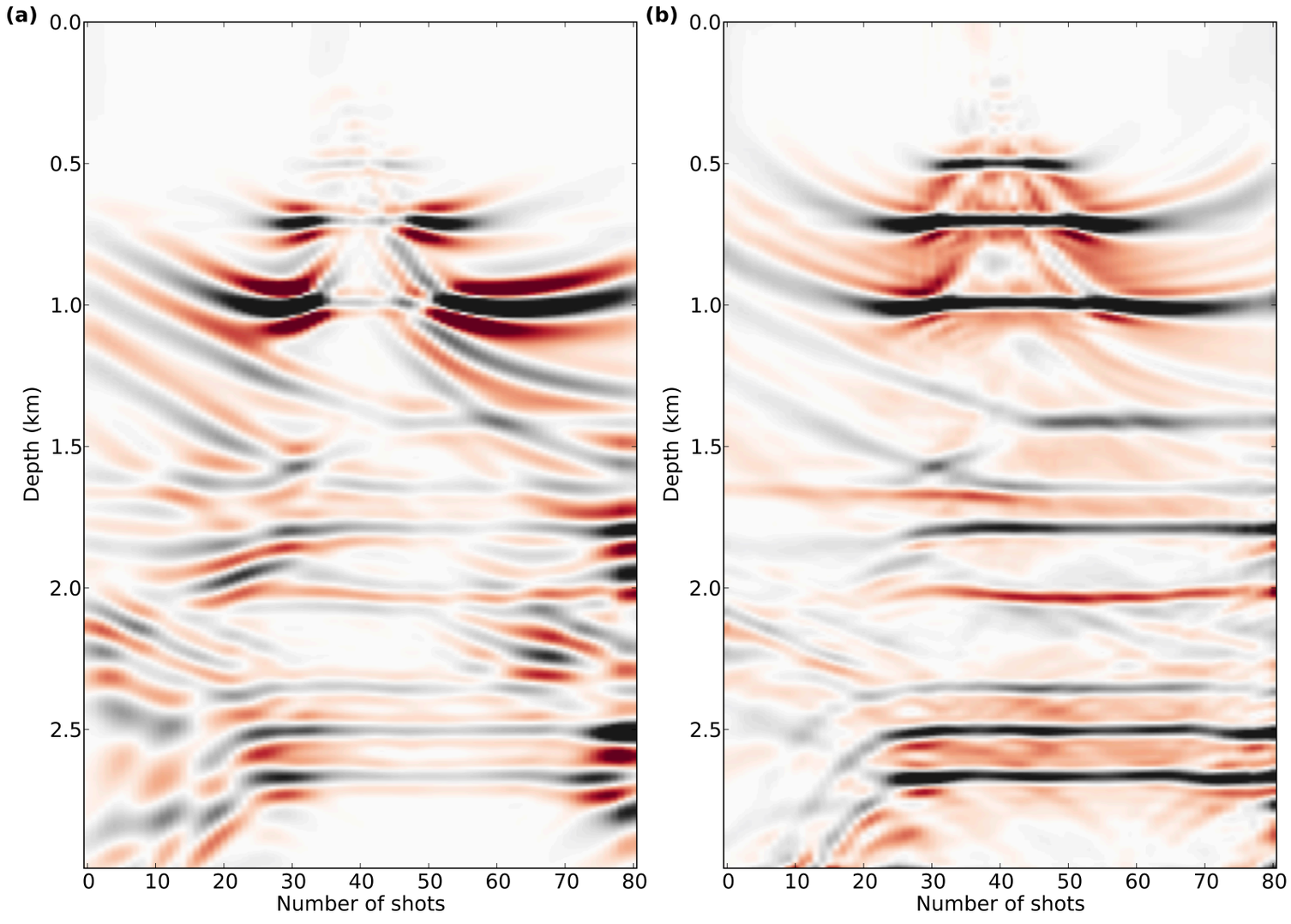
$$J = \sum_s \|\mathbf{d}_s - \mathbf{L}_s\mathbf{m}\|_2^2$$

We minimize

$$J = \sum_s \|\mathbf{d}_s - \mathbf{L}_s\mathbf{m}_s\|_2^2 + \mu \sum_s \|\mathbf{D}_s\mathbf{m}_s\|_2^2$$

Adjacent individual shot images are constraint to be similar

Index-source gather at a one lateral position (a) Migration and (b) LS migration with lateral smoothing regularization



In Linear Algebra terms: Difference between Mig and LS-Mig

- Consider de-migration and migration

$$\mathbf{d}_s = \mathbf{L}_s\mathbf{m} \qquad \textbf{(1)}$$

$$\tilde{\mathbf{m}} = \mathbf{L}_s'\mathbf{d}_s \qquad \textbf{(2)}$$

- (1) in (2) $\tilde{\mathbf{m}} = \mathbf{L}_s'\mathbf{L}_s\mathbf{m}$ **Migration yields a distorted image**

In Linear Algebra terms: Difference Mig and LS-Mig

Consider de-migration and migration

$$\tilde{\mathbf{m}} = \mathbf{L}_s'\mathbf{L}_s\mathbf{m}$$

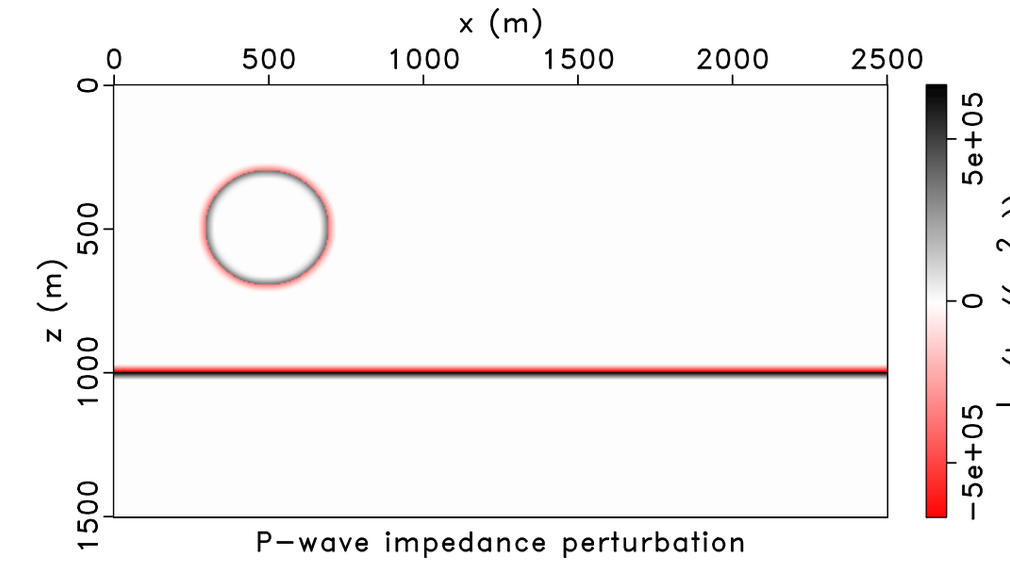
Consider the LS solution that one could have computed if the operators were matrices (they are not matrices ok!)

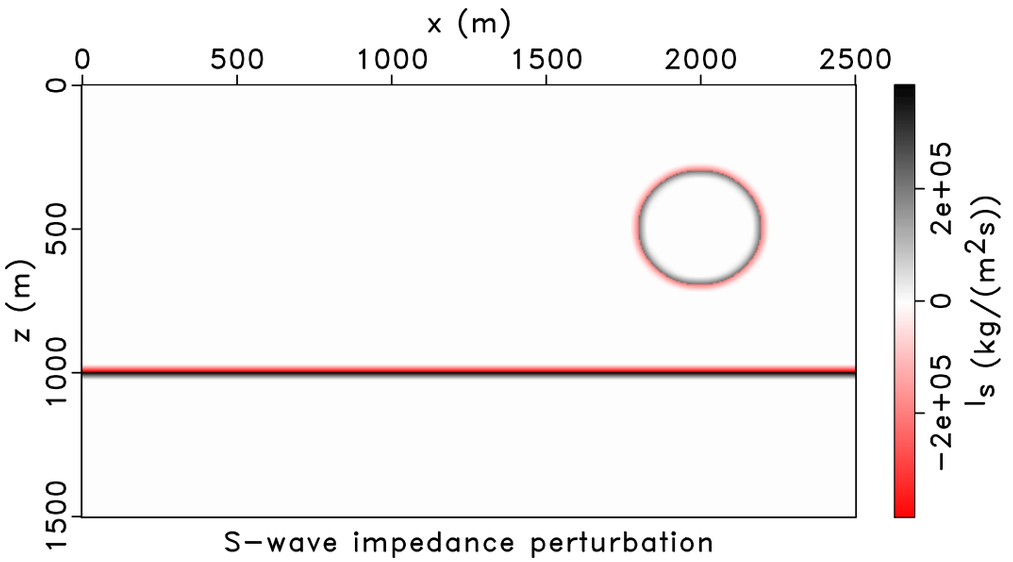
$$\mathbf{m}_{sol} = (\mathbf{L}_s'\mathbf{L}_s)^{-1}\tilde{\mathbf{m}} = (\mathbf{L}_s'\mathbf{L}_s)^{-1}\mathbf{L}_s'\mathbf{d}$$

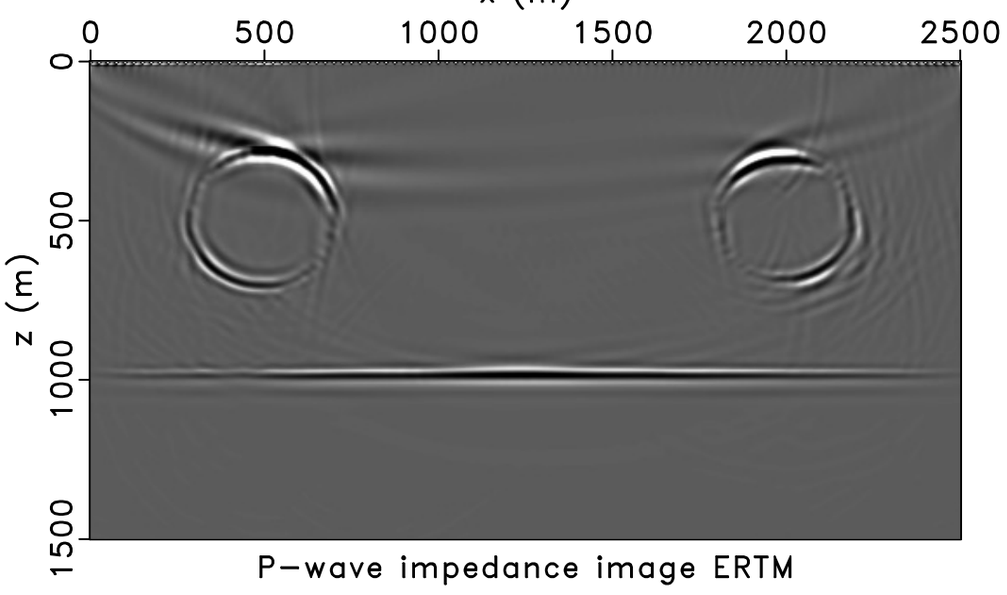
Therefore, LS-migration attempts iterative improve the migration image by removing the Hessian operator. Because the inverse cannot be physically constructed, we use iterative methods such as SD or CG that only require the operator $\mathbf{L}$ and $\mathbf{L'}$.

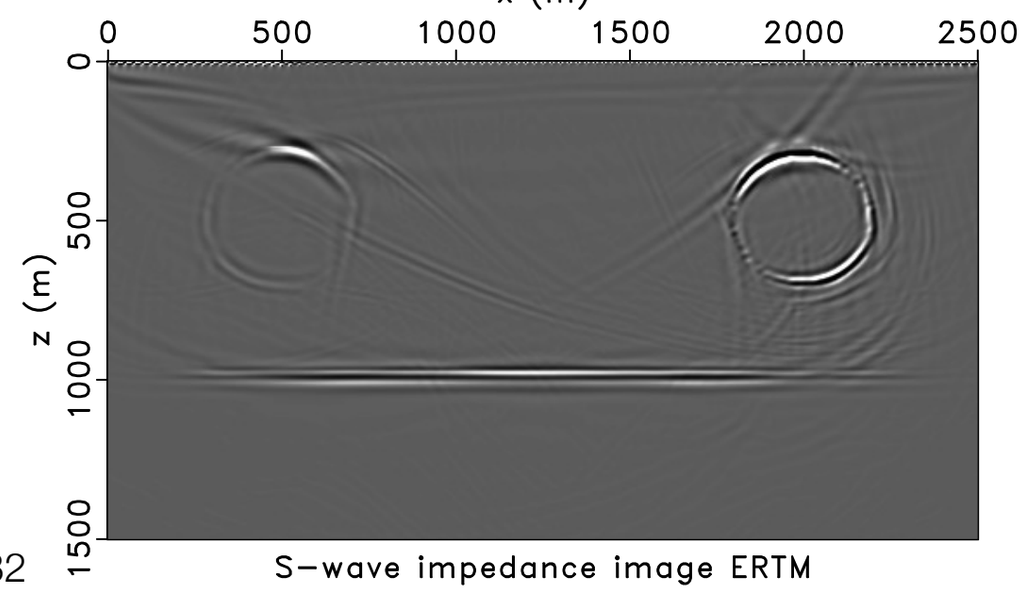
Multi-parameter LS migration (more challenging problem, Ke Chen, PhD thesis)

- Simultaneous inversion of P and S-wave velocity perturbations via LS migration





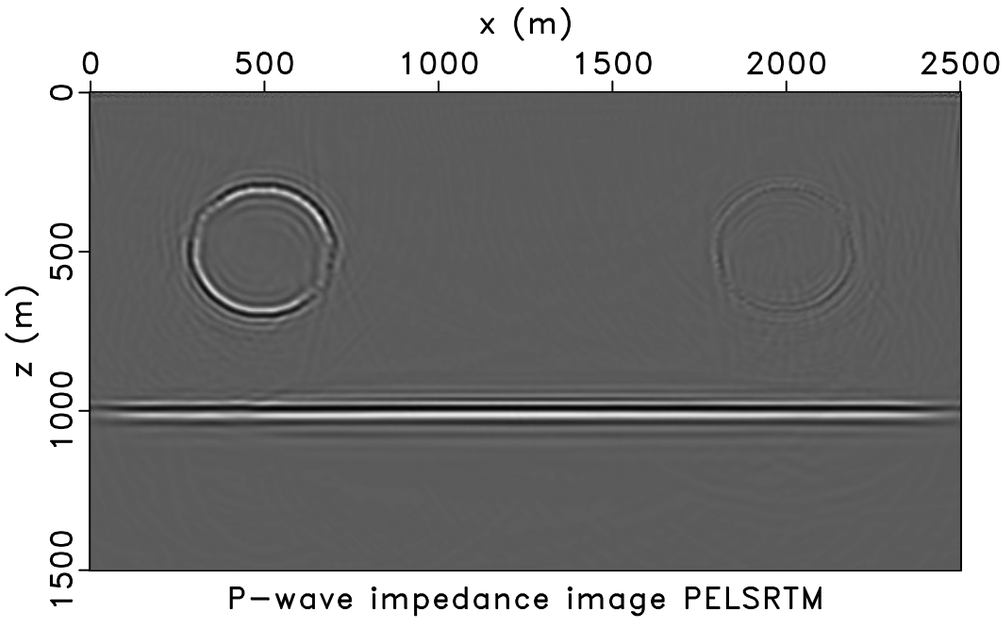


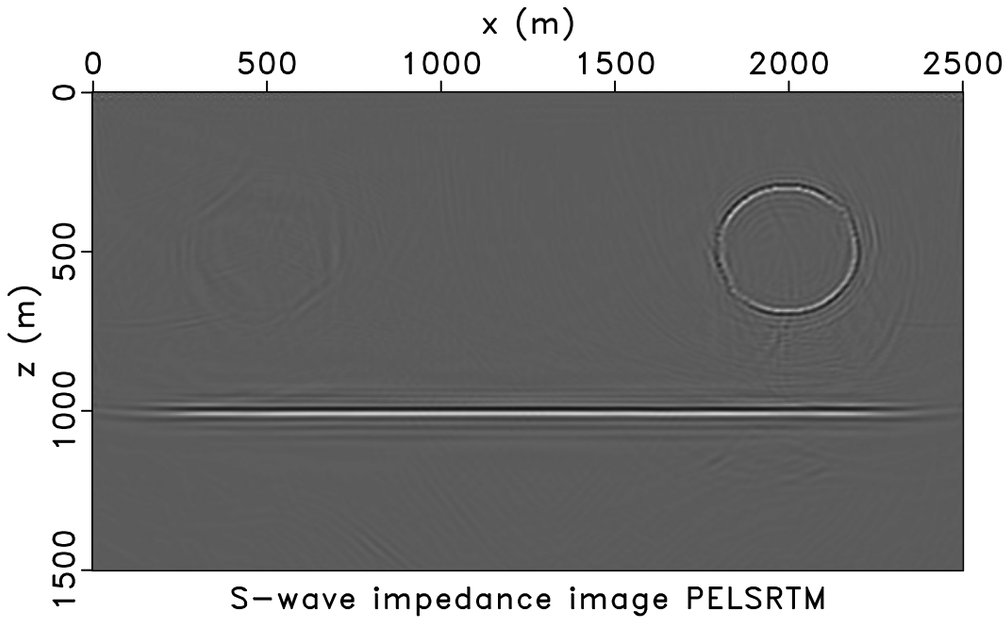


Multi-parameter LS migration (more challenging problem, Ke Chen, PhD thesis)

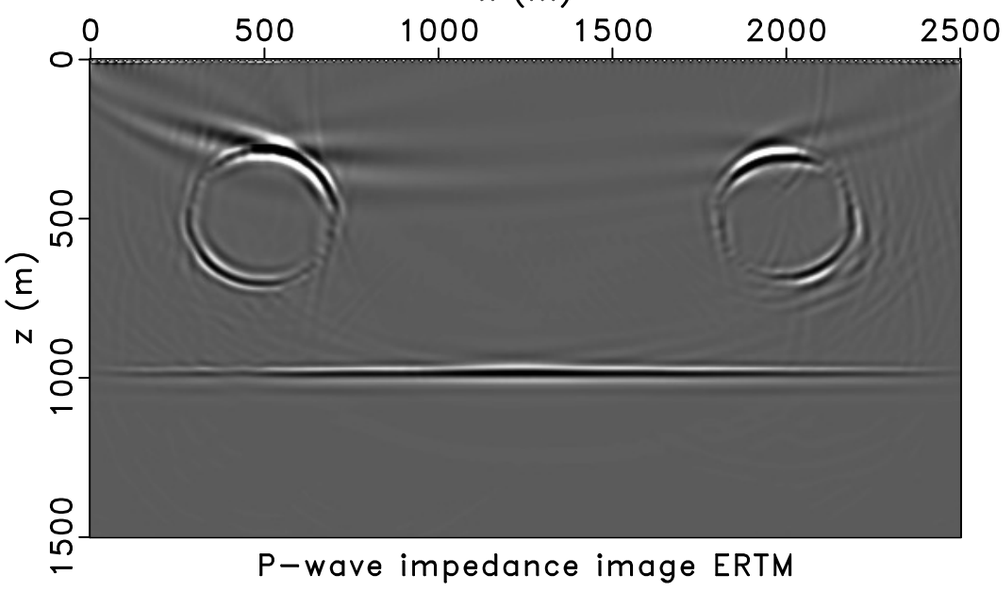
- Simultaneous inversion of P and S-wave velocity perturbations via LS migration

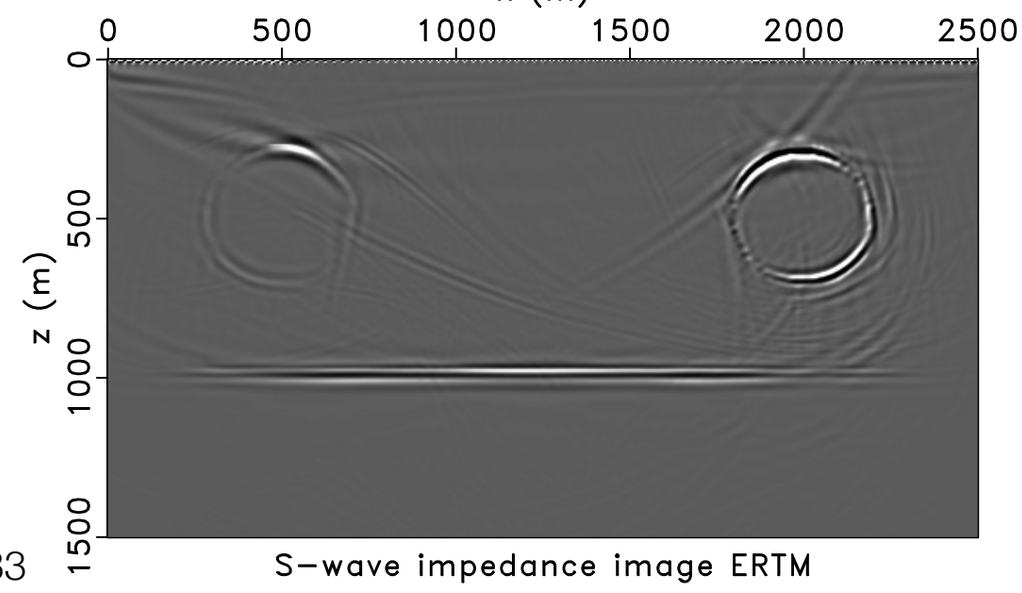
LS-Mig



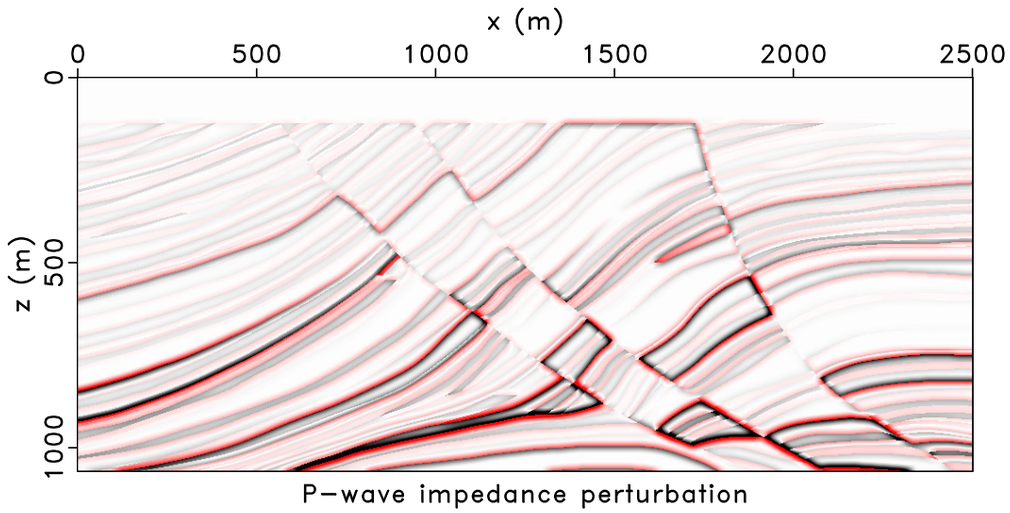


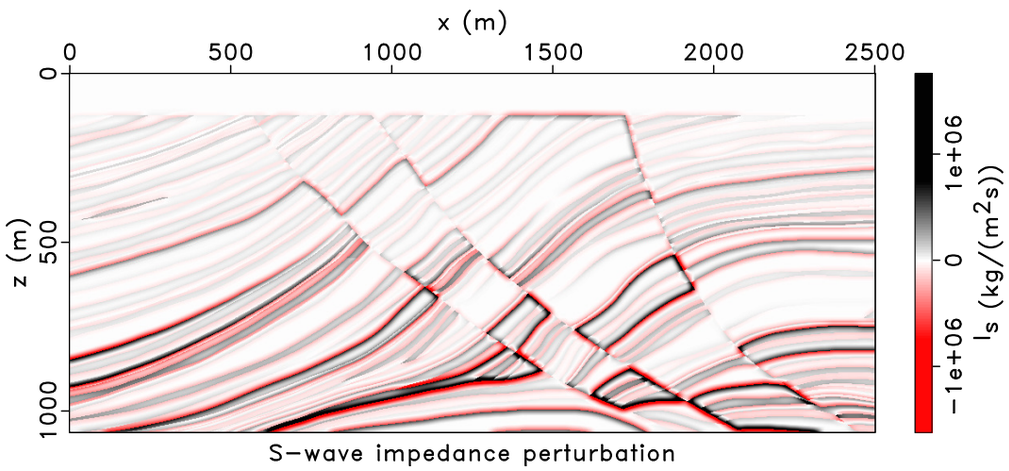
Mig





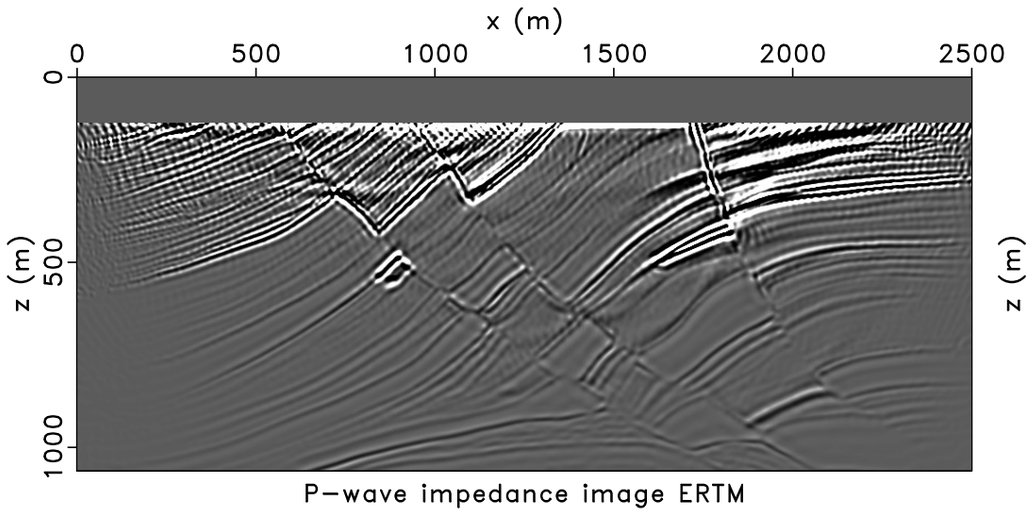
Marmousi example



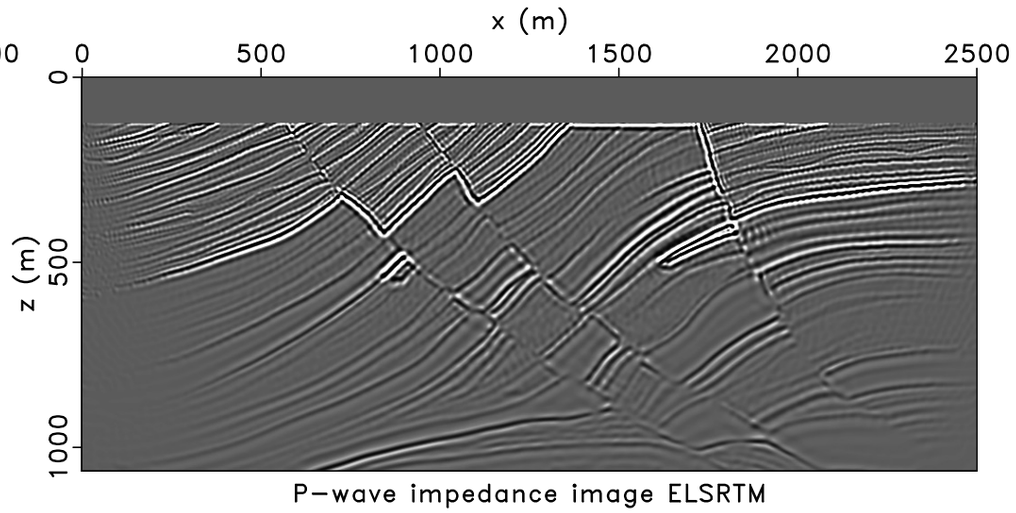


Chen and Sacchi, GJI, 2018

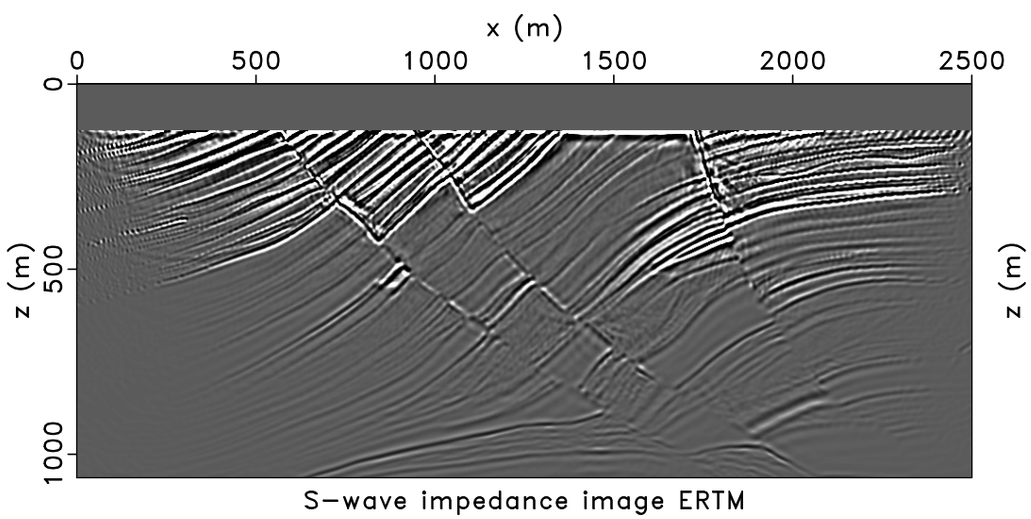
Elastic RTM Mig vs Elastic RMT-LS Mig



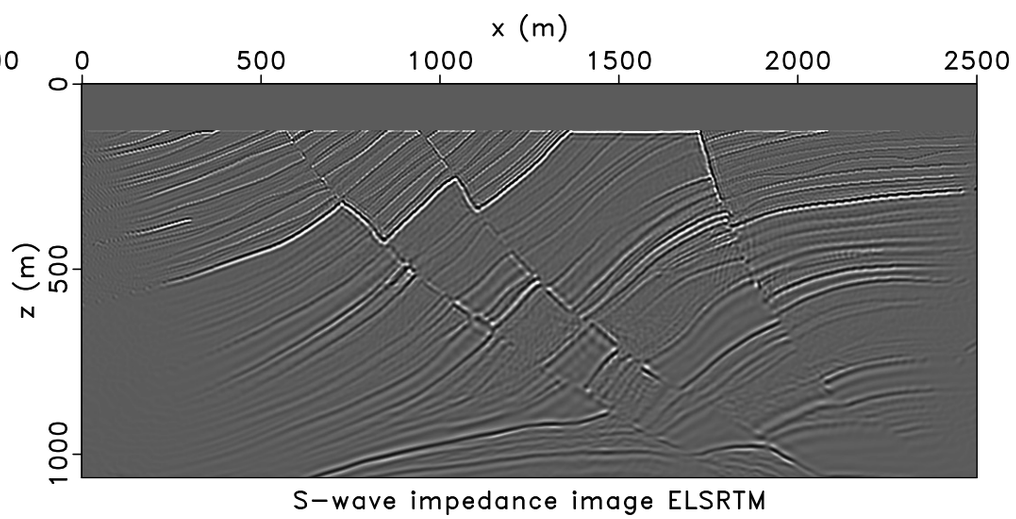
P−wave impedance image ERTM



P−wave impedance image ELSRTM



S−wave impedance image ERTM



S−wave impedance image ELSRTM