Extended images in *action*

Tristan van Leeuwen, Rajiv Kumar & Felix Herrmann

SLIM
University of British Columbia

Motivation

Computation of *full*-subsurface offset volumes is computationally *prohibitively* expensive (storage & computation time)

Full-subsurface *offset* volumes allow us to conduct

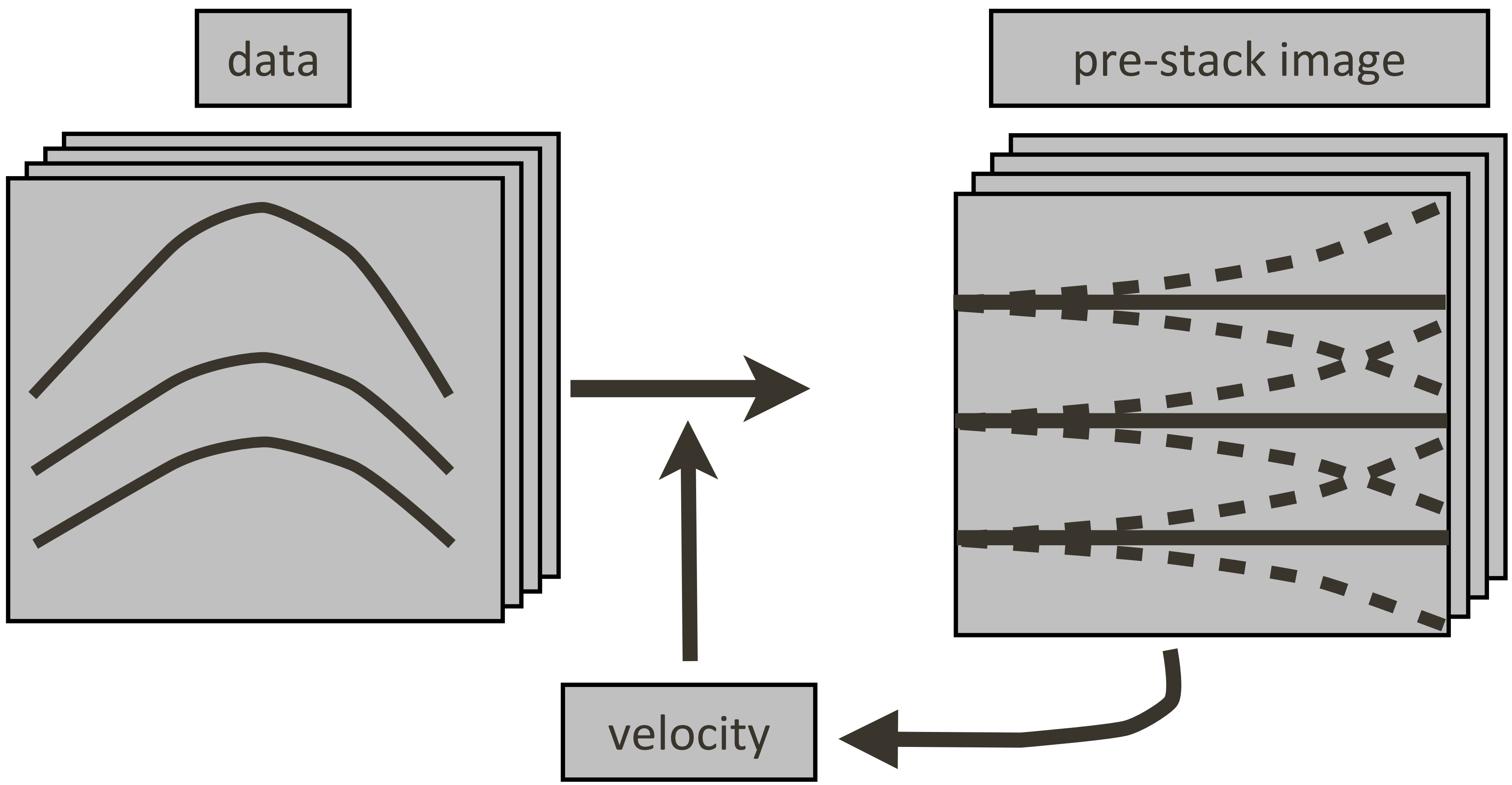
- MVA
- AVA w/ geologic *dip* corrections

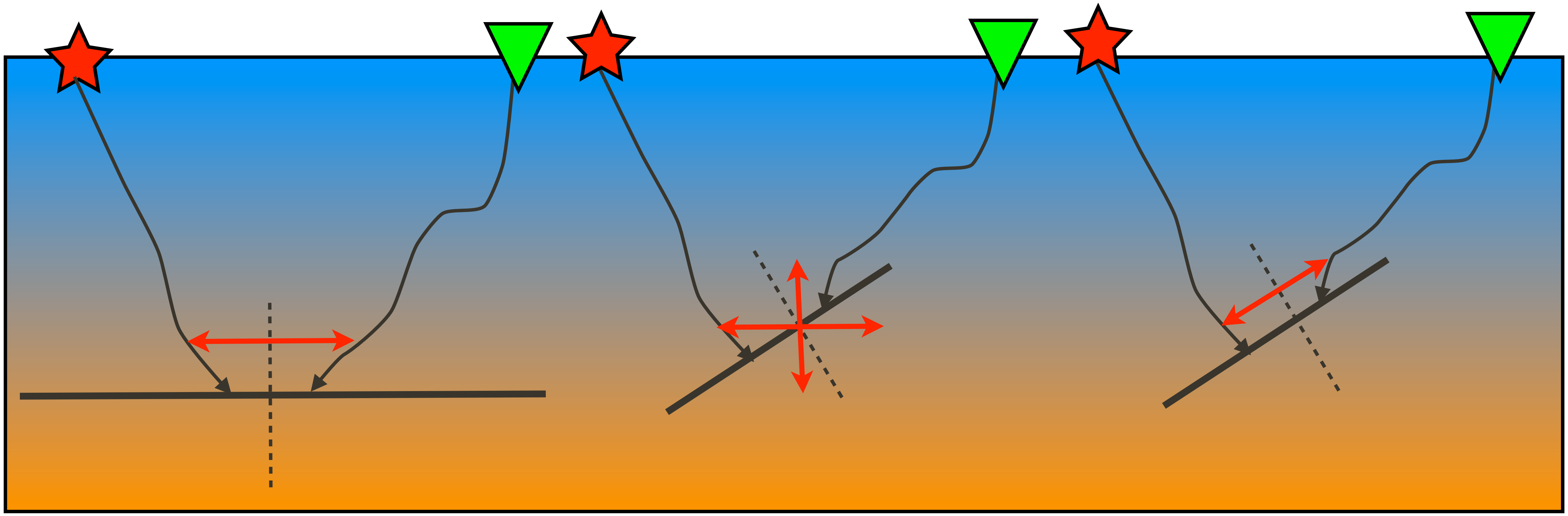
using information from *all* directions.

Use *probing* techniques we used *successfully* in *FWI*...

Migration-velocity analysis

- find *starting* model for FWI?
- *invert* kinematic errors in *image* volumes





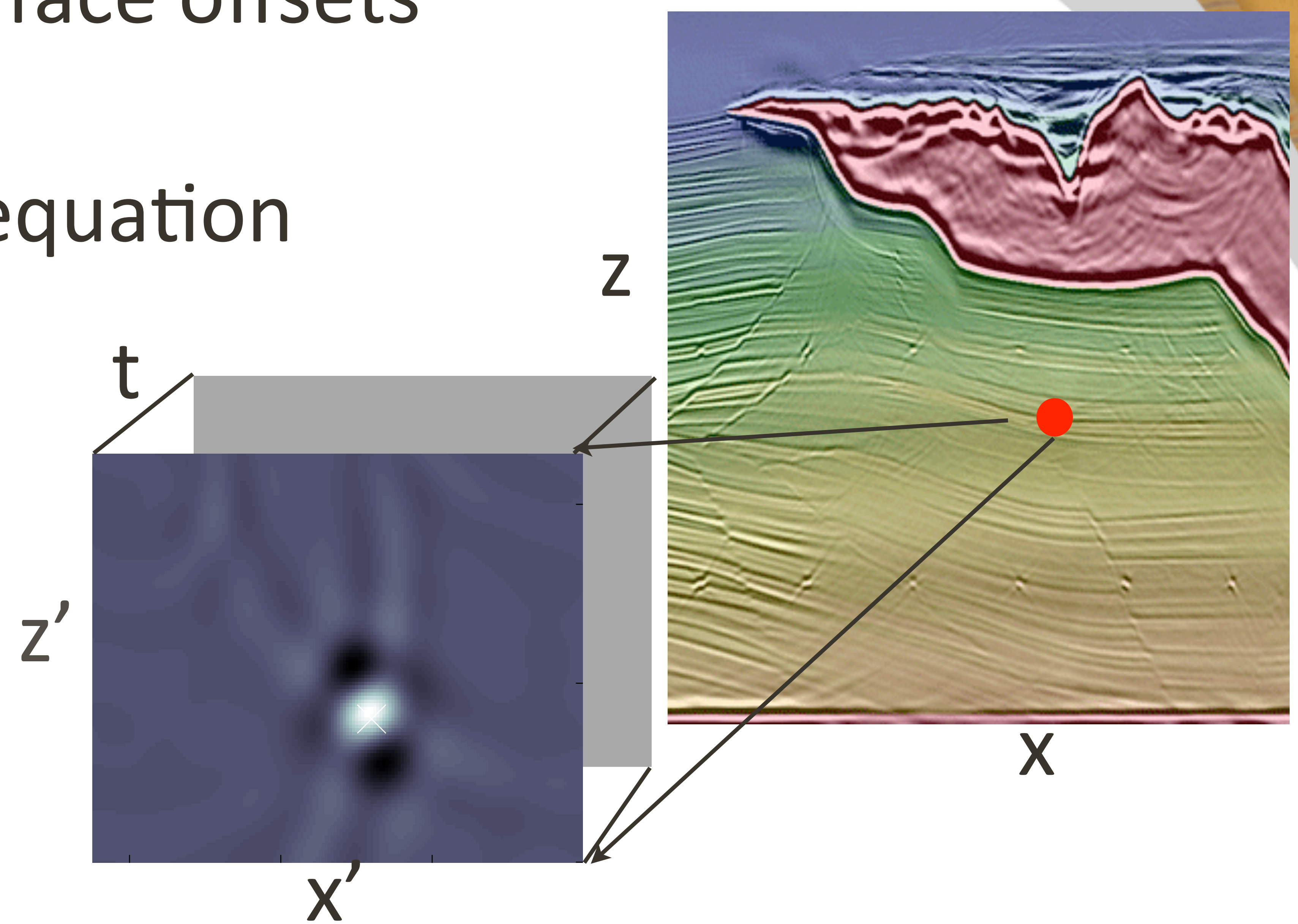
horizontal offset

horizontal +vertical offset

all offsets

[Biondo & Symes, ’04 ;Sava & Vasconcelos, ’11]

- use *all* subsurface offsets (5D volume)
- 2-way wave-equation



but.... we can *never* hope to *compute* or *store* such an *extended* image volume!

Can we work with the *extended* volume *implicitly* ?

SLIM

Overview

- Anatomy
- Physics
- Computation
- Applications:
    1. AVA
    2. MVA
- Conclusions

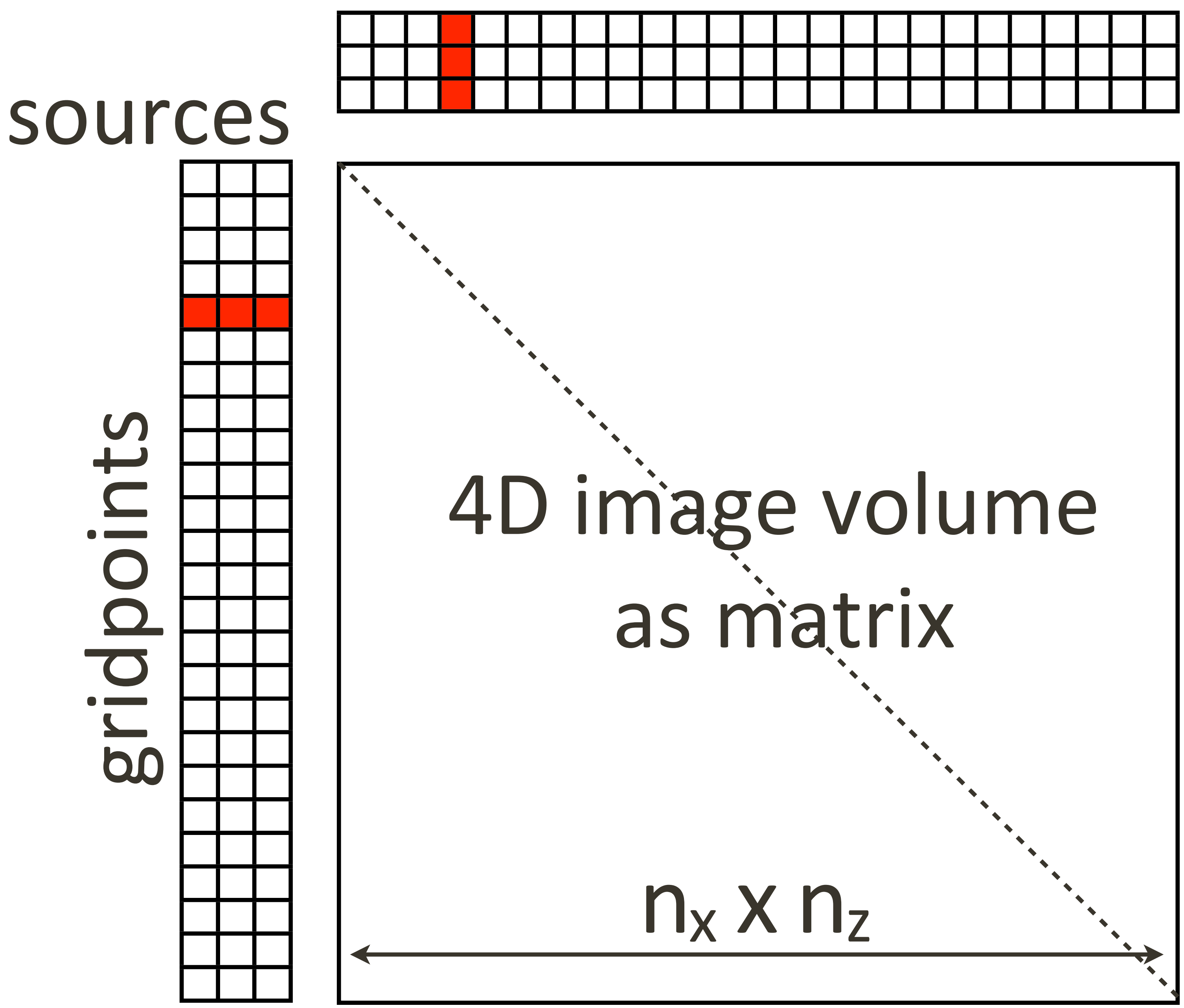
Anatomy

$$e(\omega, \mathbf{x}, \mathbf{x}') = \sum_i u_i(\omega, \mathbf{x}) v_i(\omega, \mathbf{x}')^*$$

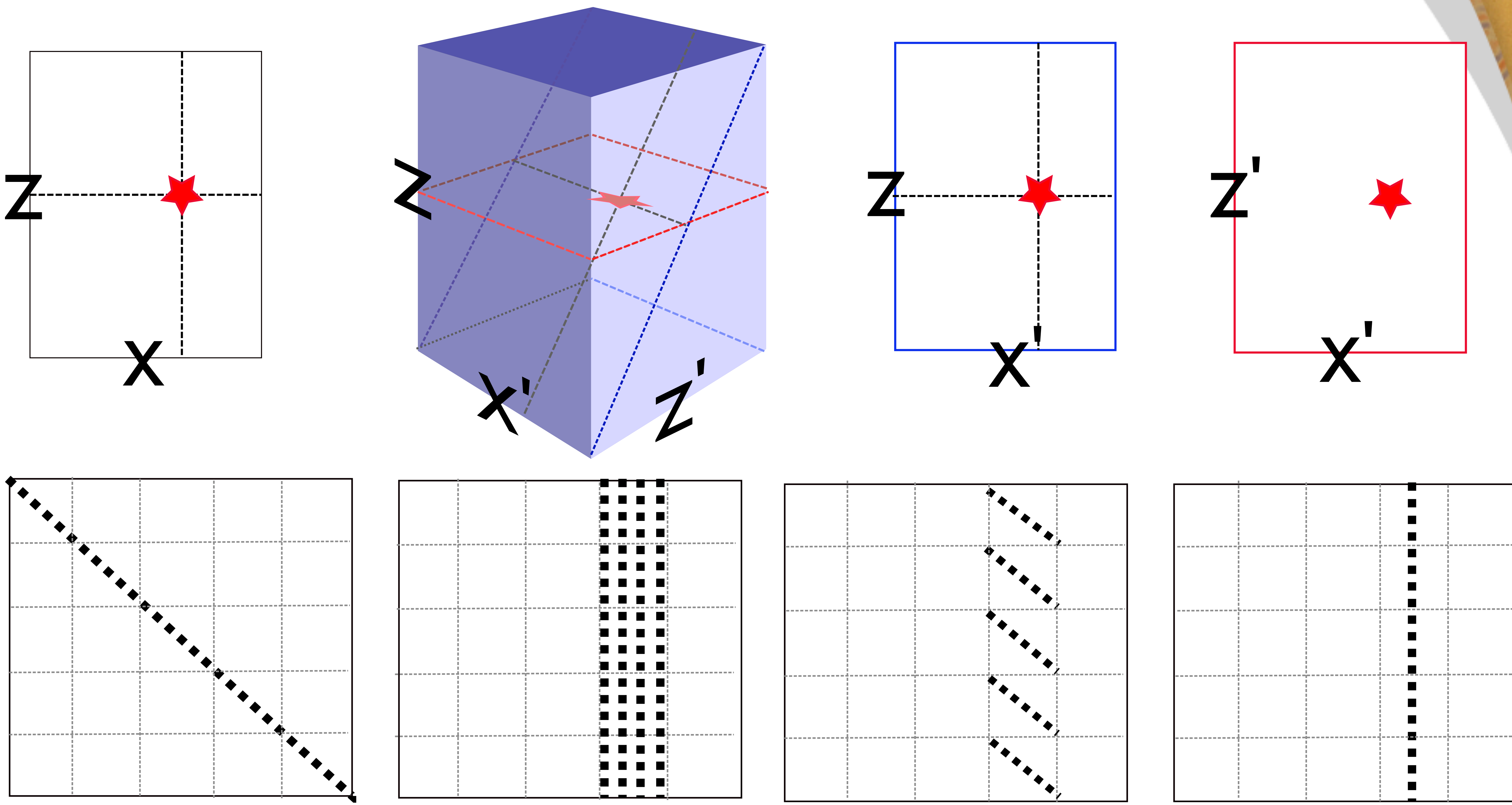
- Organize wavefields in monochromatic *data matrices*
- *Express* image volume *tensor* as *matrix*

$$E = UV^*$$

Extended images

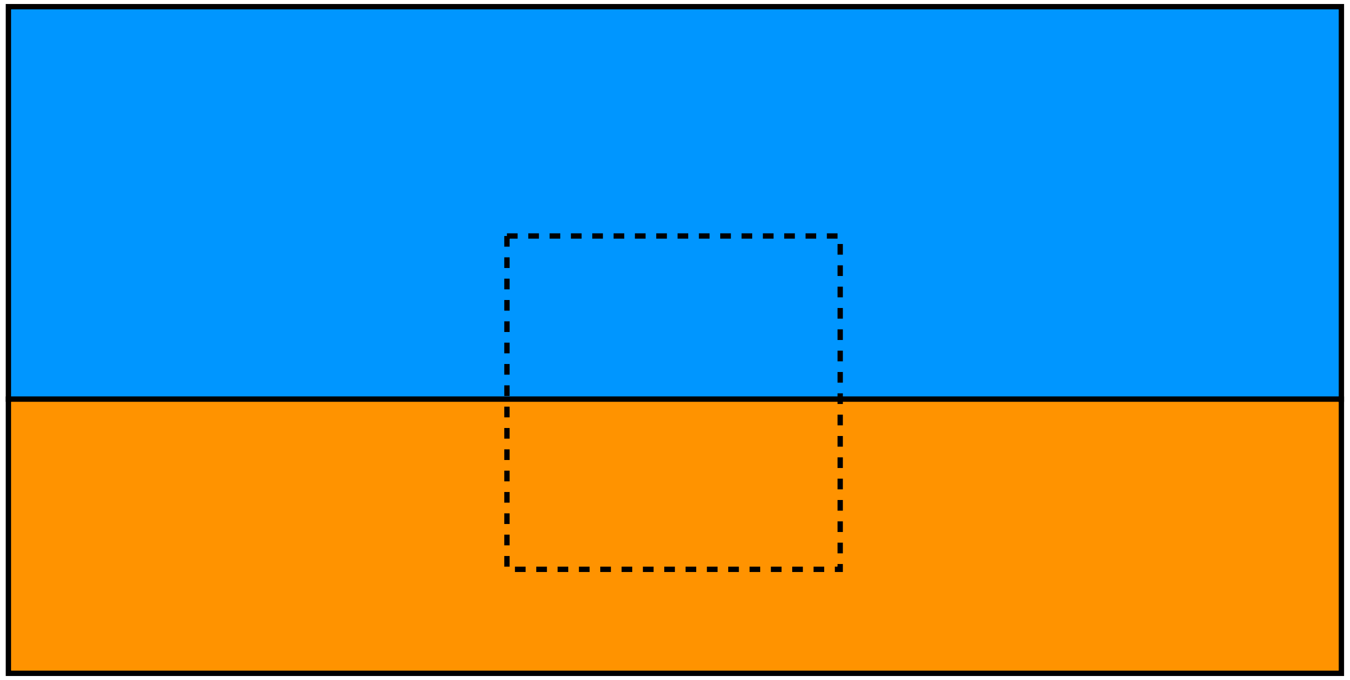


Extended images



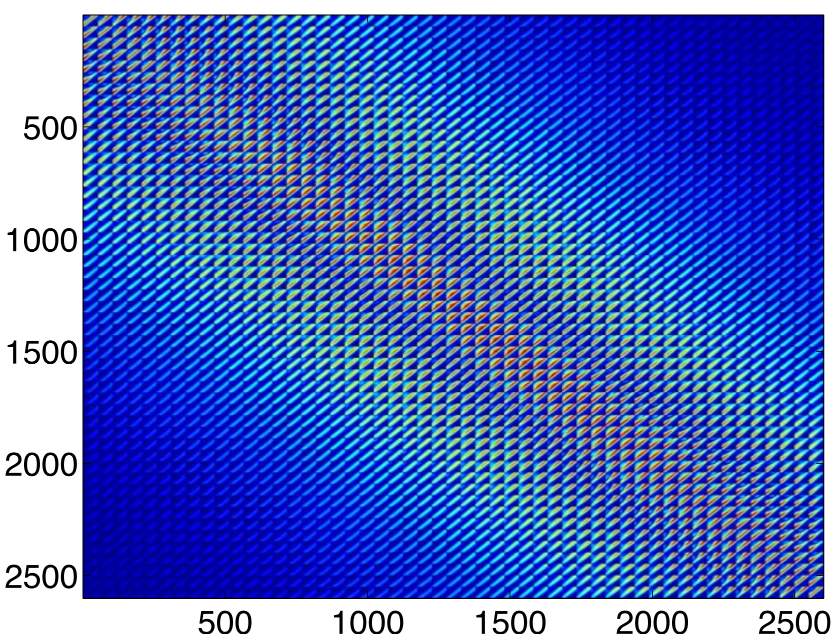
Extended images

example for *one* layer

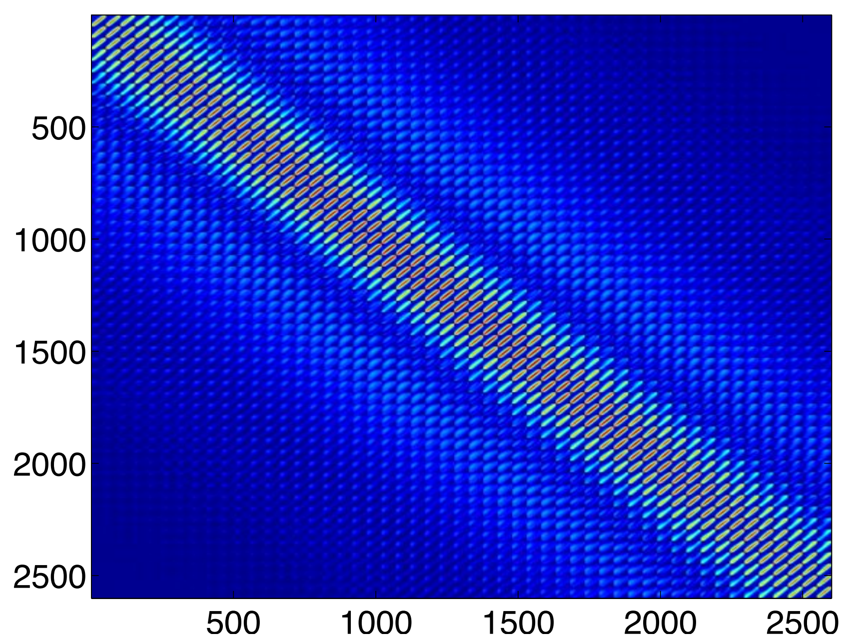


Extended images

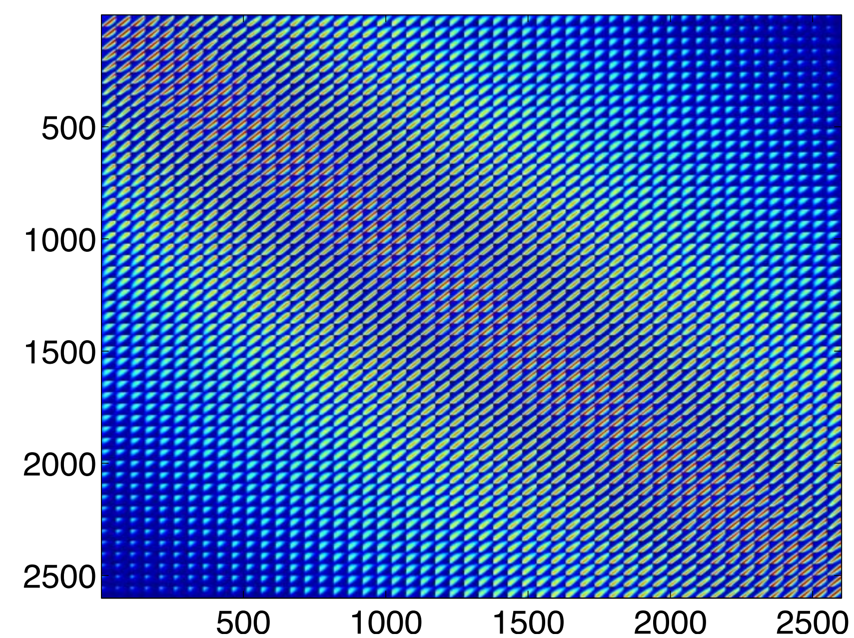
full matrix



low velocity



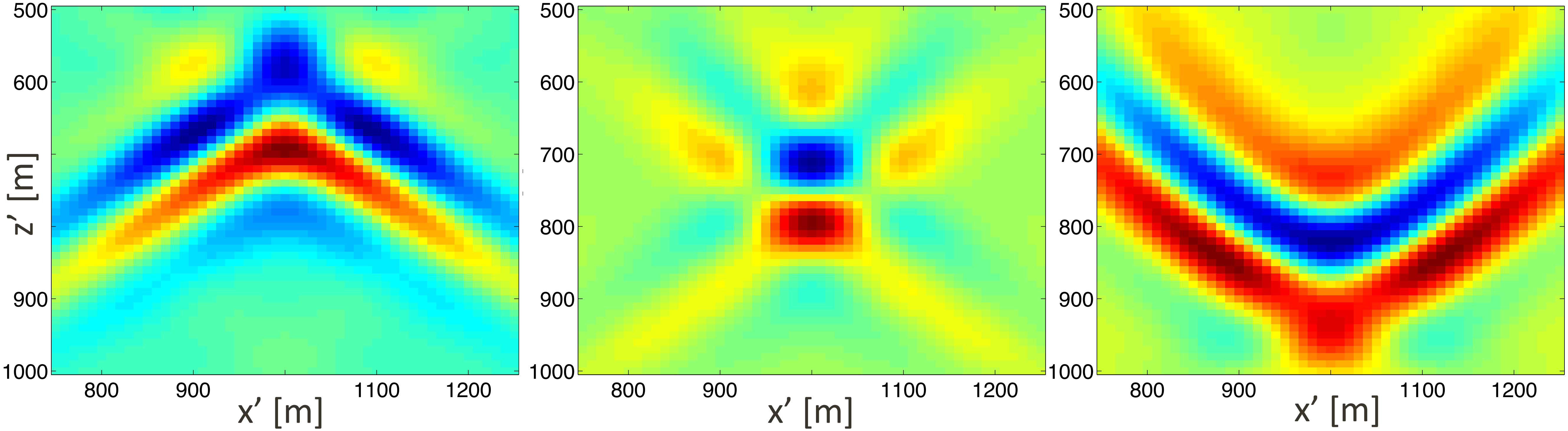
correct velocity



high velocity

Extended images

one column



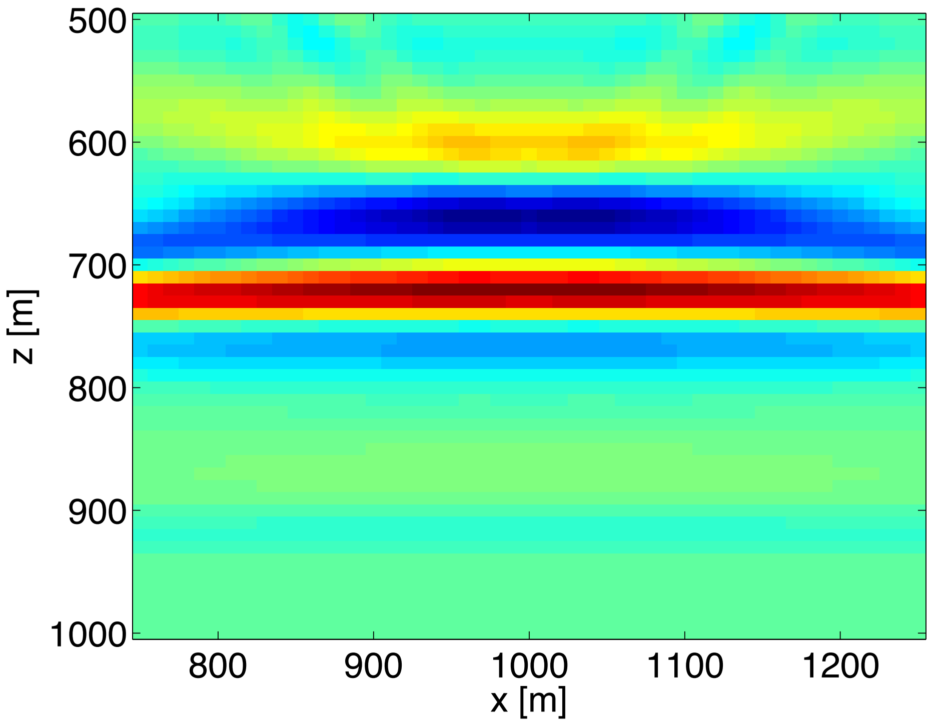
low velocity

correct velocity

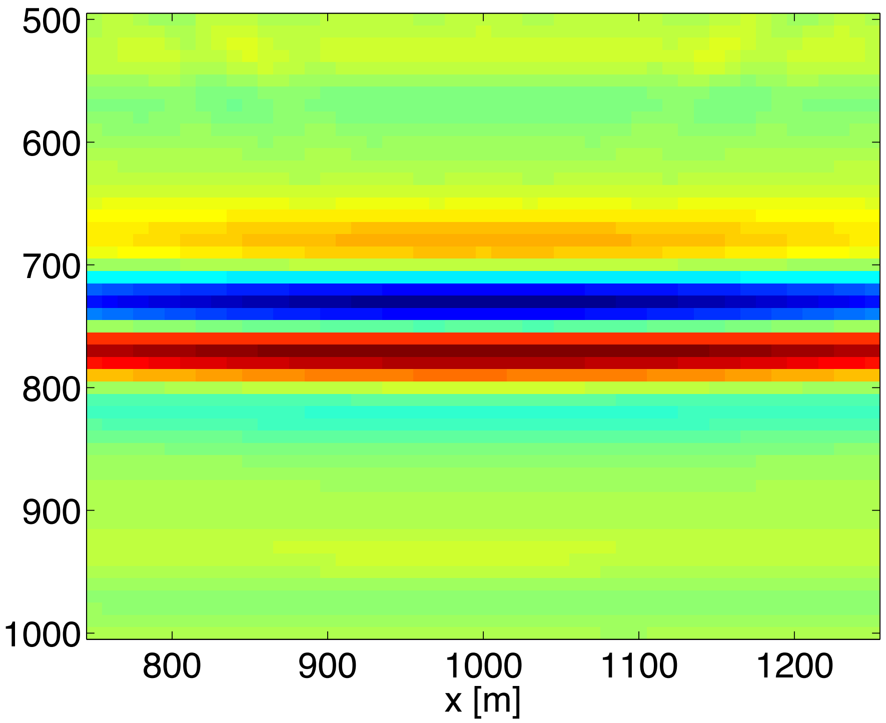
high velocity

Extended images

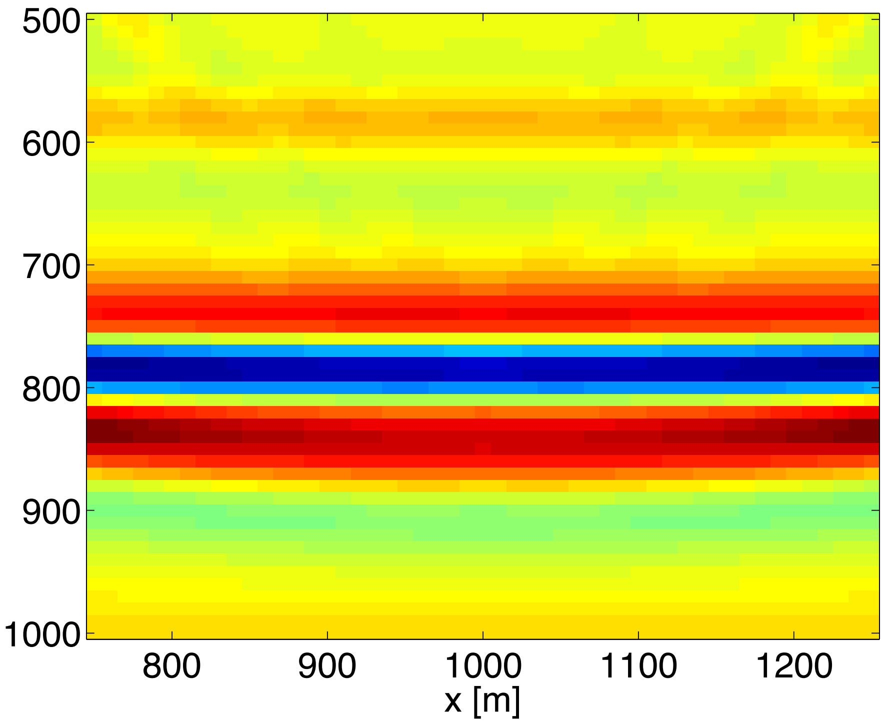
diagonal



low velocity



correct velocity



high velocity

Double wave-equation

Helmholtz operator: $H = \omega^2 \text{diag}(\mathbf{m}) + \nabla^2$

source/receiver wavefields:

$$HU = P_s^T Q \qquad H^* V = P_r^T D$$

RTM extended image: $E = UV^*$

yields: $HEH = P_s^T Q D^* P_r$

Double wave-equation

$$Le(\omega,\mathbf{x},\mathbf{x}') = \int \mathrm{d}\mathbf{s} \int \mathrm{d}\mathbf{r}\, d(\omega,\mathbf{s},\mathbf{r})\delta(\mathbf{x}-\mathbf{s})\delta(\mathbf{x}'-\mathbf{r})$$

two-way:

$$L = \left[\omega^2/c(z,x)^2 + \partial_x^2 + \partial_z^2\right]\left[\omega^2/c(z',x')^2 + \partial_{x'}^2 + \partial_{z'}^2\right]$$

one-way (DSR):

$$L = \left[\partial_z - \imath\sqrt{\omega^2/c(z,x)^2 + \partial_x^2} - \imath\sqrt{\omega^2/c(z,x')^2 + \partial_{x'}^2}\right]$$

[Claerbout, ’84; Stolk & de Hoop ’01]

Computation

- *complete* image volume too *large* to form: $(n_x \times n_z)^2$
- instead, *probe* volume for information via *mat-vecs* $E\mathbf{y}$
- $\mathbf{y}$ can be interpreted as subsurface (sim.) *source* function

SLIM

Computation

mat-vec with *extended* image:

$$\mathbf{e} = E\mathbf{y} = H^{-1}P_s^TQD^*P_rH^{-1}\mathbf{y}$$

- $\widetilde{\mathbf{d}} = P_rH^{-1}\mathbf{y}$ *(one subsurface source)*
- $\widetilde{\mathbf{w}} = D^*\widetilde{\mathbf{d}}$ *(source weights)*
- $\mathbf{e} = H^{-1}P_s^TQ\widetilde{\mathbf{w}}$ *(one source)*

Computation

Are able to compute *full*-subsurface image gathers

- w/o *looping* over *all* sources
- derives from *reciprocity* principle
- probe image space w/ arbitrary *test* functions
  - *point* scatterers (one at location of subsurface point)
  - Gaussian weights (*simultaneous* source)

Computation

computation of an *image point gather*

	# of PDE solves	"flops for correlations"
conventional	$2N_s$	$N_s \times N_h$
mat-vecs	2	$N_s \times N_r$

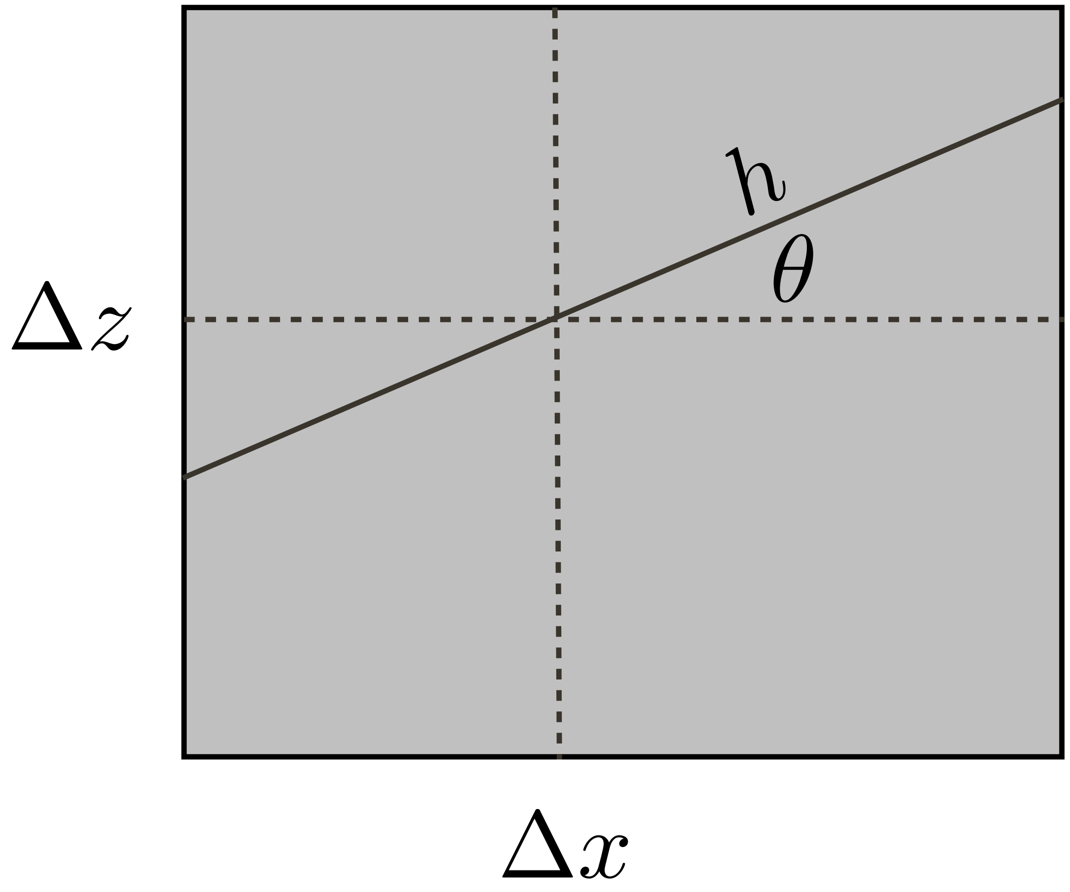
N_s - # of sources

N_r - # of receivers

N_h - # of subsurface offsets

Dip-angle gathers

align subsurface *offset* with local *dip*

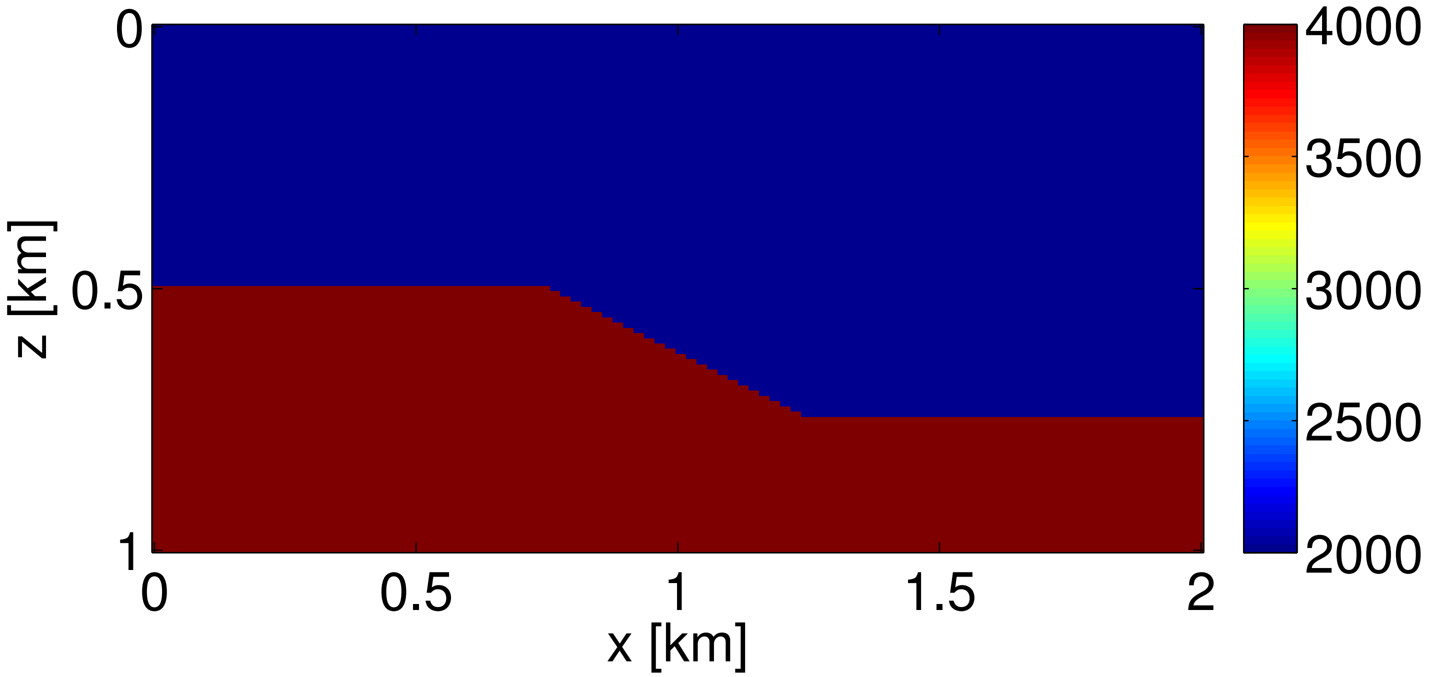


Dip-angle gathers

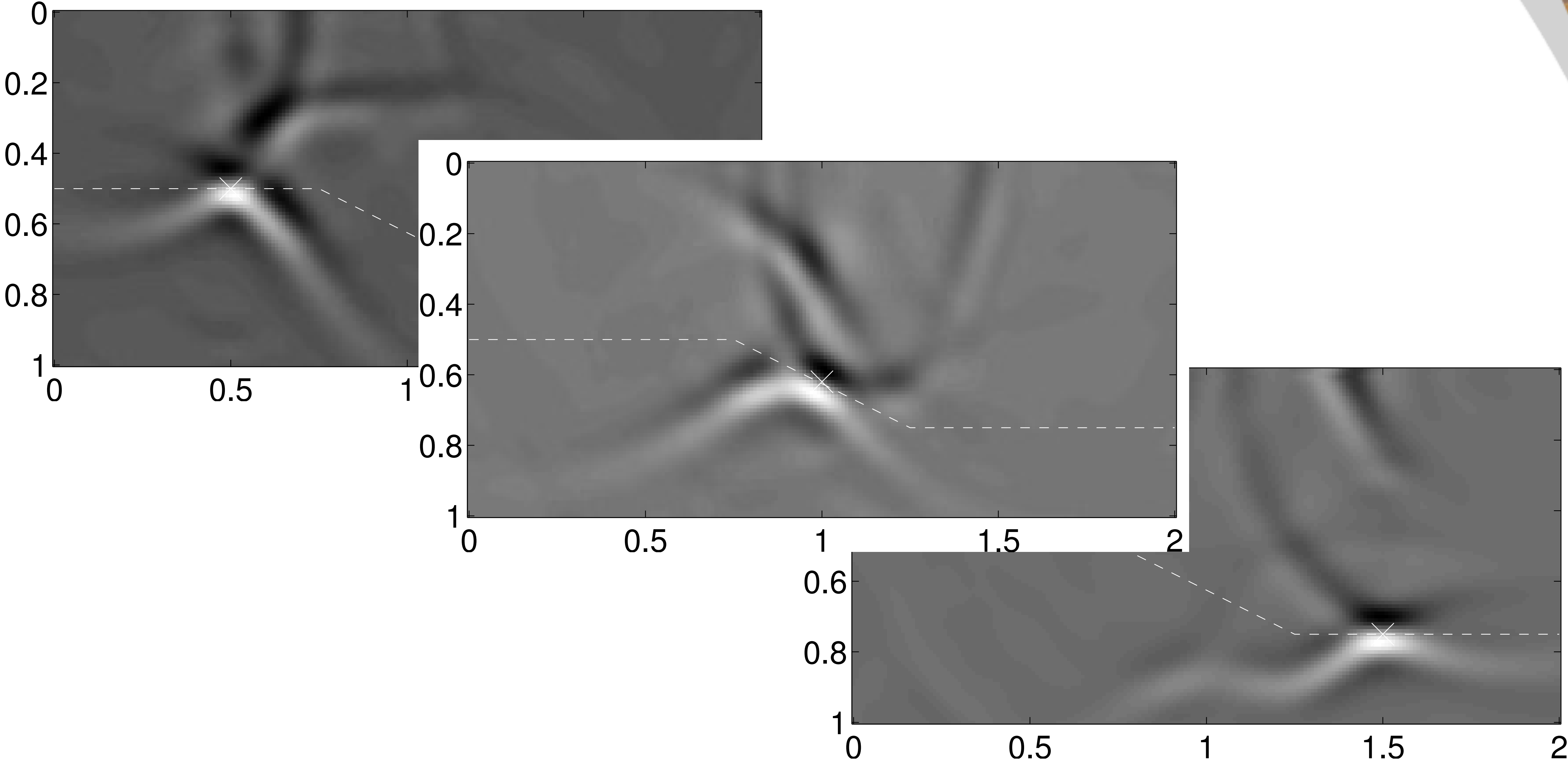
1. compute *image*-point gather
2. determine dip
3. extract *offset* along *dip*
4. *Radon* transform to compute *angle* gather

Dip-angle gathers

example



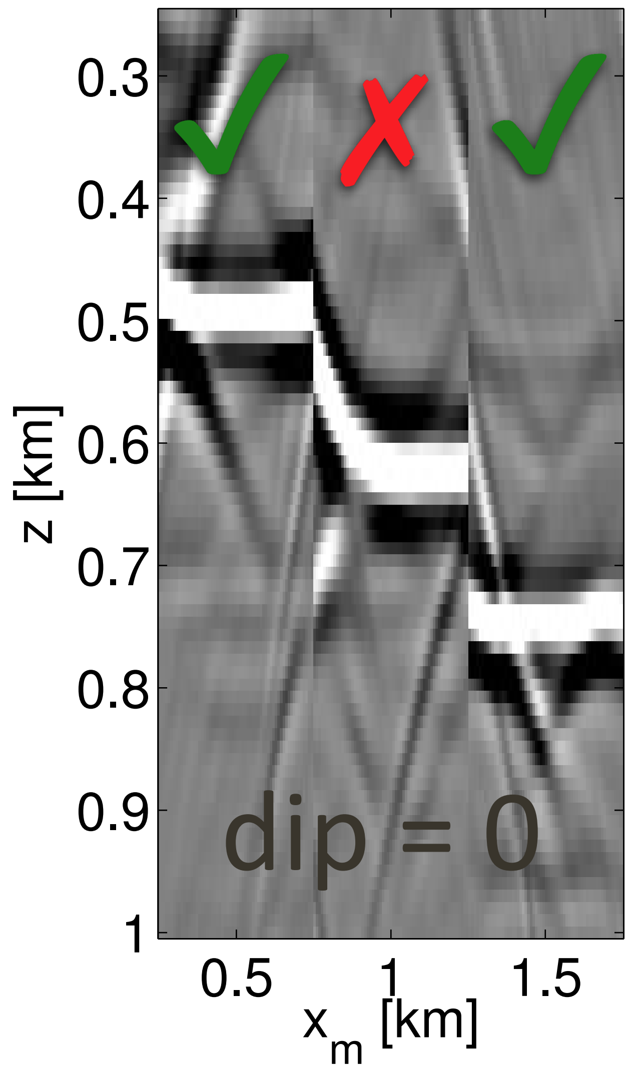
Dip-angle gathers

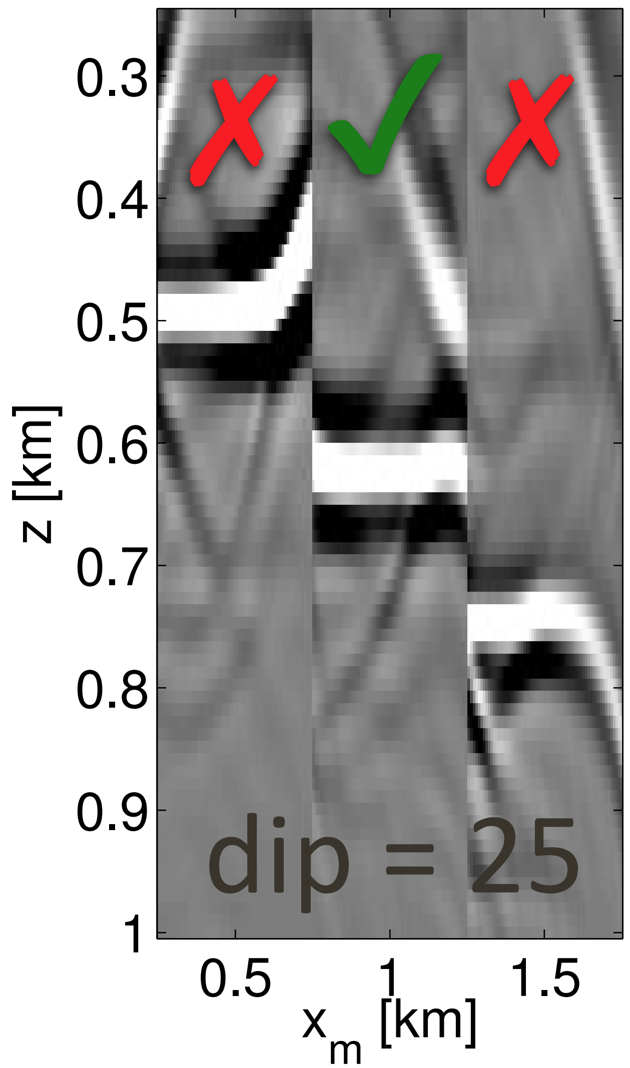


SLIM

Dip-angle gathers

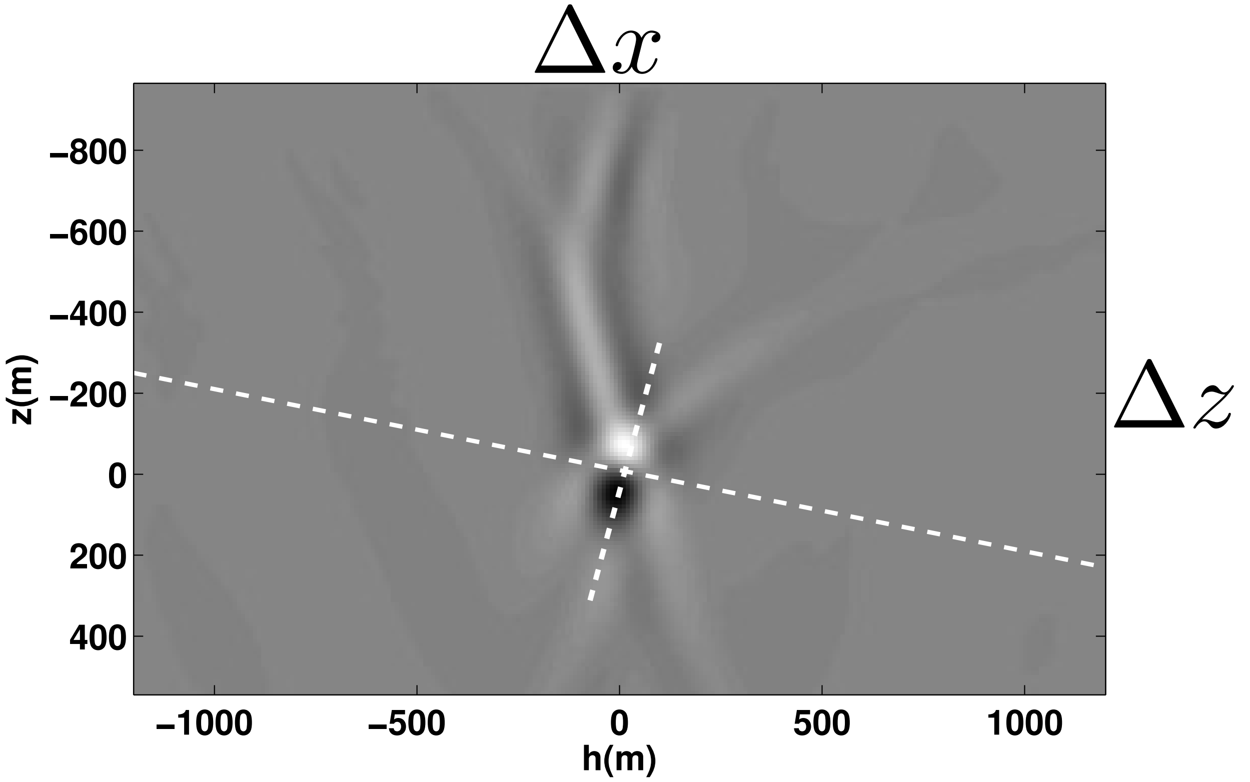
angle gathers for *correct* velocity, should all be *flat*

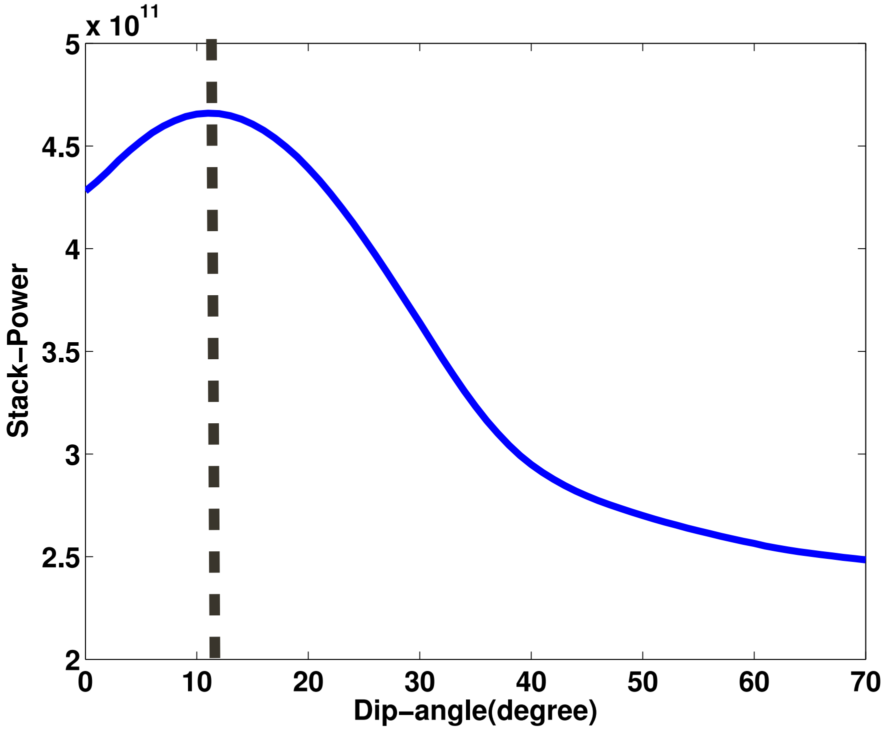




Dip-angle gathers

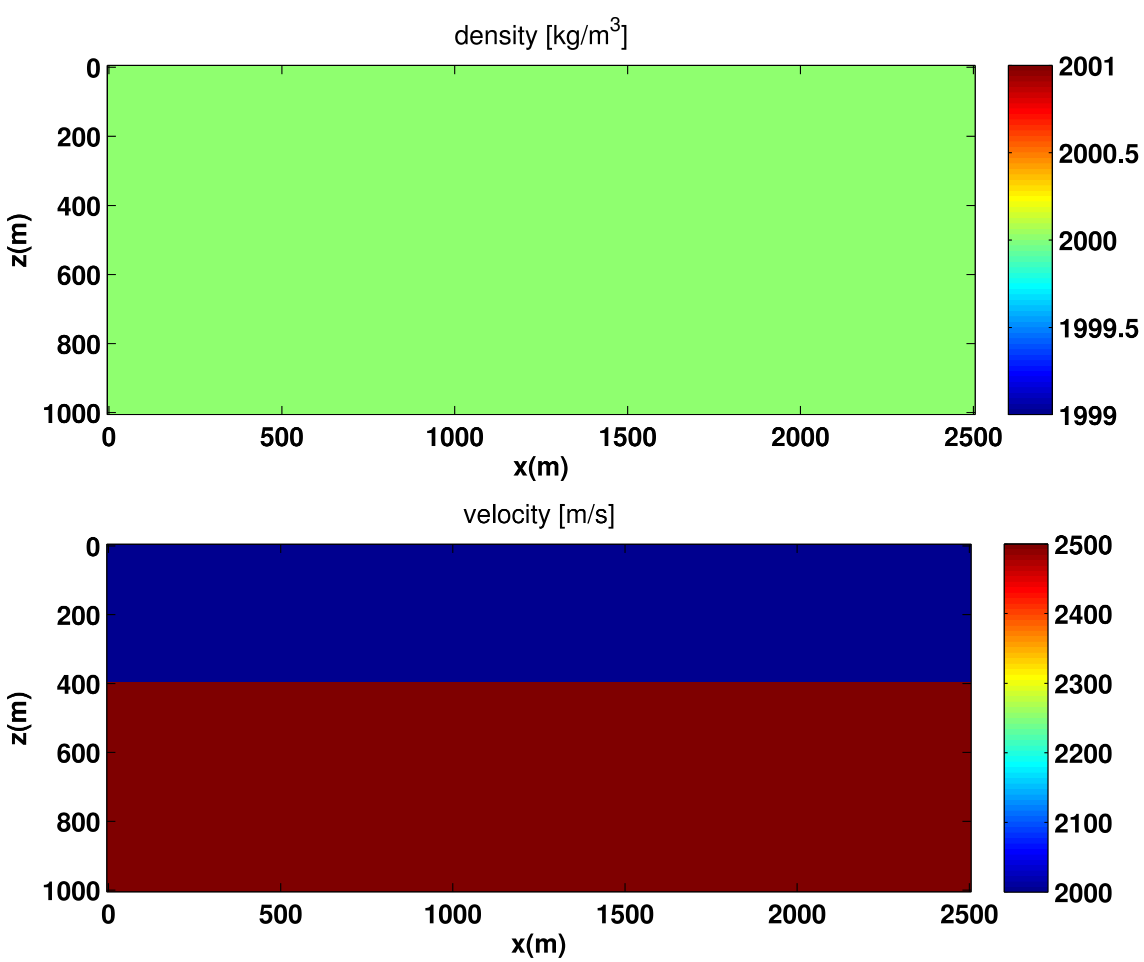
the *dip* can be *detected* by measuring the *stackpower* normal to the *dip*

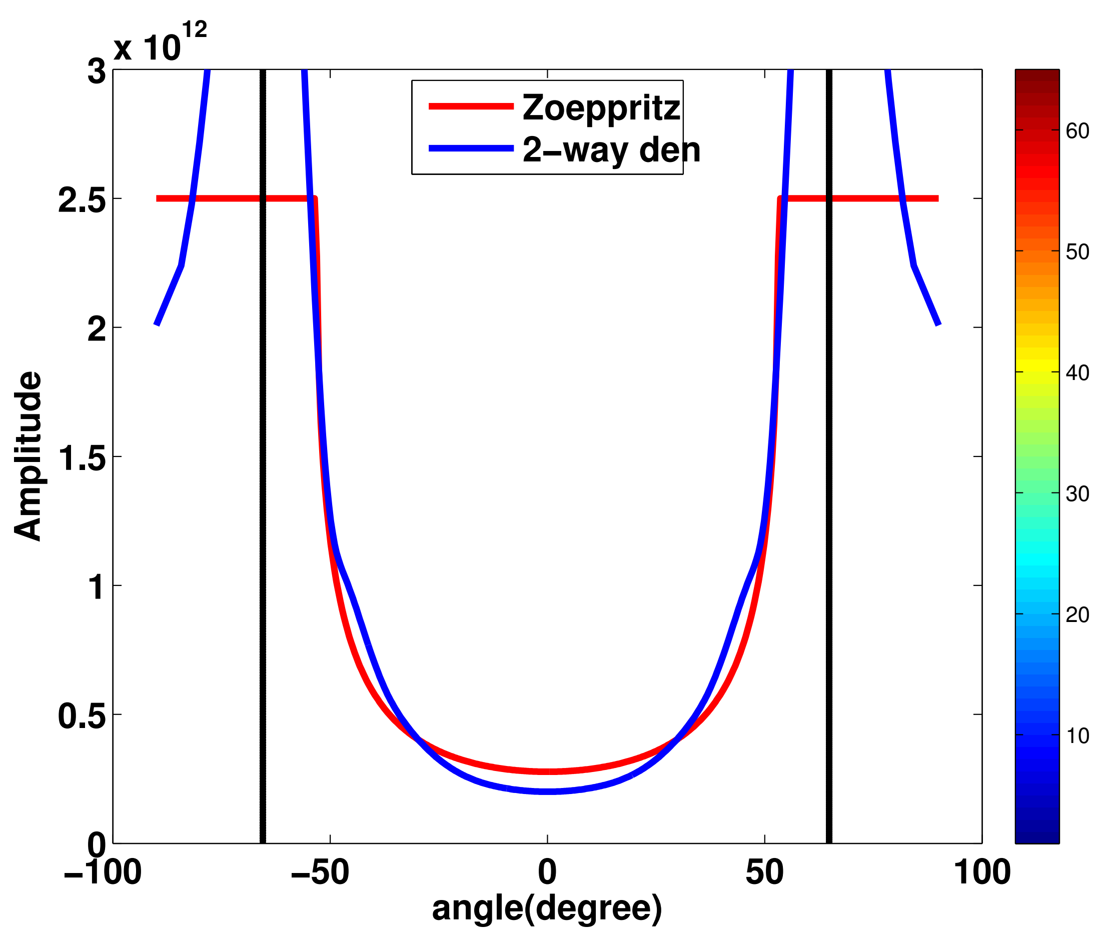




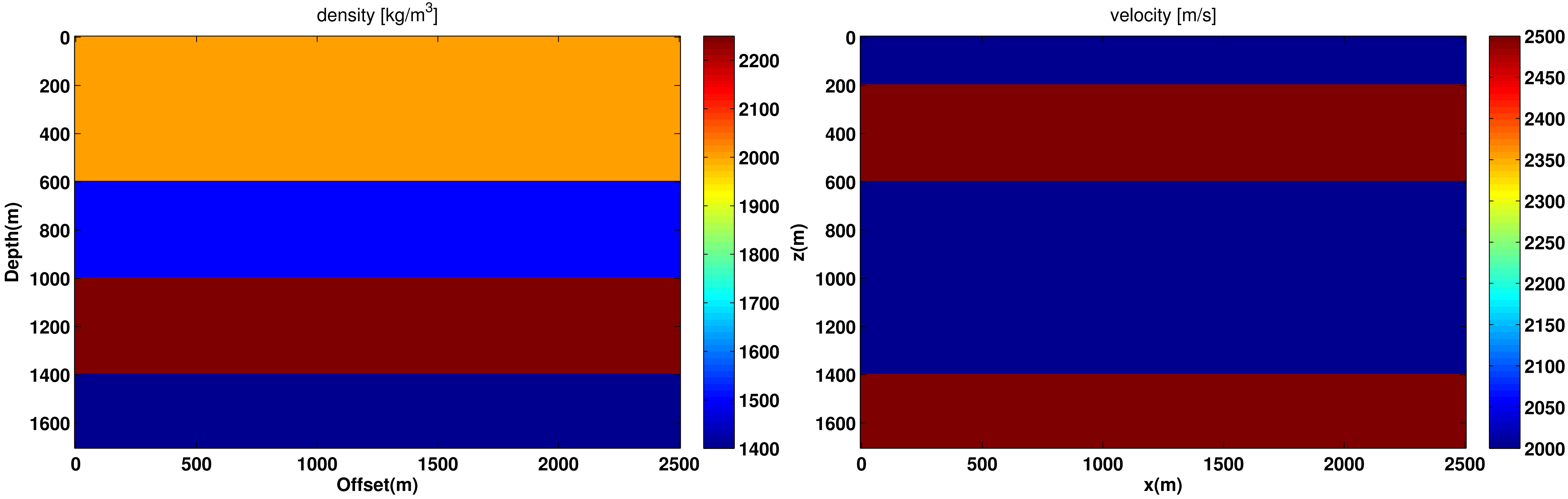
AVA

amplitude behavior of *angle* gathers can be used for *AVA*

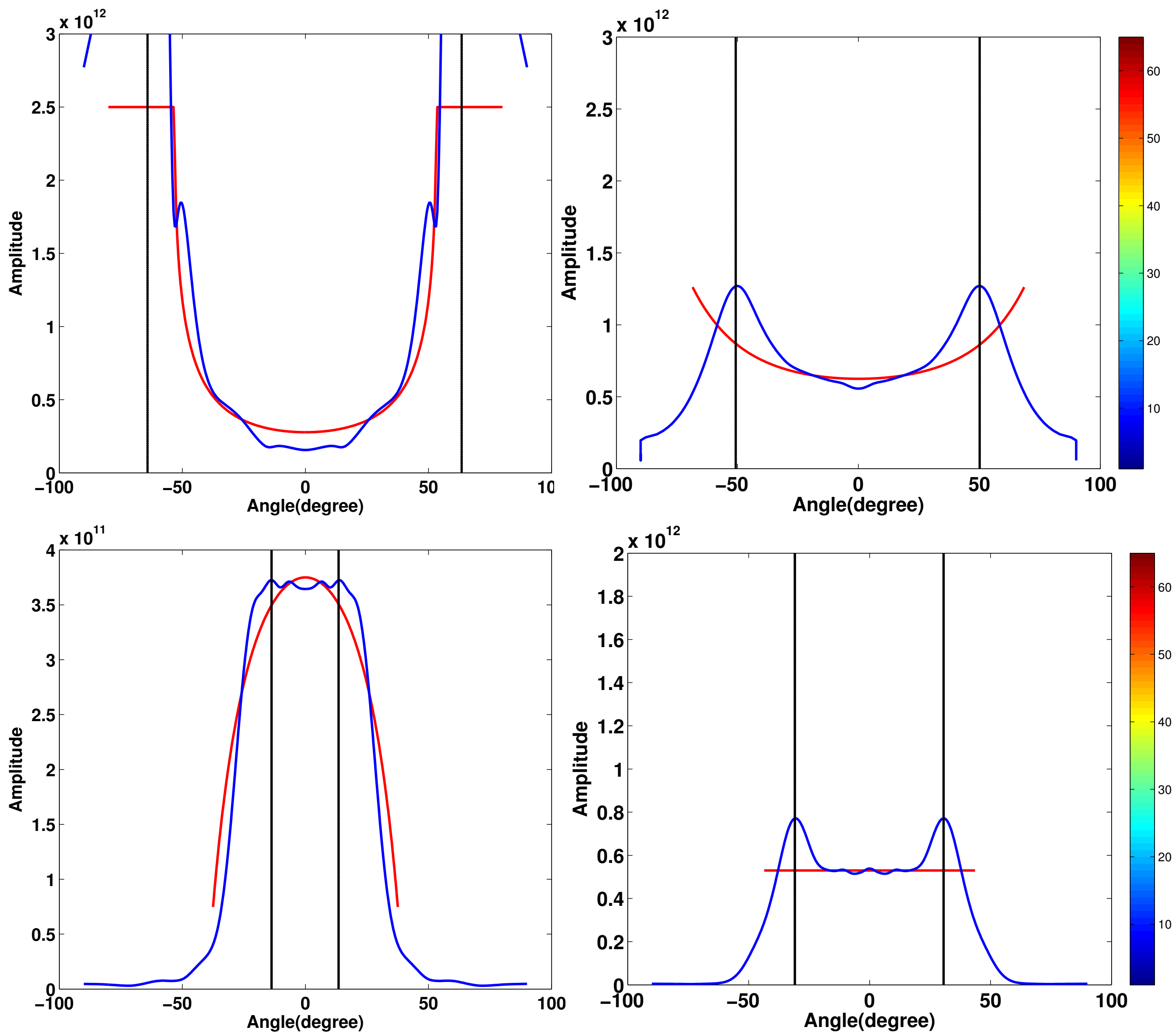




AVA


density [kg/m3]
Depth(m)
Offset(m)
0
200
400
600
800
1000
1200
1400
1600
0
500
1000
1500
2000
2500
2200
2100
2000
1900
1800
1700
1600
1500
1400
velocity [m/s]
z(m)
x(m)
2500
2450
2400
2350
2300
2250
2200
2150
2100
2050
2000

AVA



x 10^12
Amplitude
Angle(degree)
x 10^12
Amplitude
Angle(degree)
x 10^11
Amplitude
Angle(degree)
x 10^12
Amplitude
Angle(degree)

MVA

Focusing in Δx implies a *commutation* relation: $x \cdot f(x, x') = x' \cdot f(x, x')$

or

$$E\text{diag}(\mathbf{x}) = \text{diag}(\mathbf{x})E$$

Measure the error in some norm

$$||E\text{diag}(\mathbf{x}) - \text{diag}(\mathbf{x})E||_?^2$$

[Symes '08]

MVA

The *Frobenius* norm can be estimated via *randomized* trace estimation:

$$||A||_F^2 = \text{trace}(A^T A)$$

$$\approx \sum_{i=1}^{K} \mathbf{w}_i^T A^T A \mathbf{w}_i = \sum_{i=1}^{K} ||A\mathbf{w}_i||_2^2$$

where $\sum_{i=1}^{K} \mathbf{w}_i \mathbf{w}_i^T \approx I$

[Avron & Toledo, '11]

MVA

objective and *gradient*

$$\phi(\mathbf{m}) = \sum_k \frac{1}{2}||R(\mathbf{m})\mathbf{w}_k||_2^2$$

$$\nabla\phi(\mathbf{m}) = \sum_k DR(\mathbf{m}, \mathbf{w}_k)^* R(\mathbf{m})$$

where

$$R(\mathbf{m}) = E(\mathbf{m})\mathrm{diag}(\mathbf{x}) - \mathrm{diag}(\mathbf{x})E(\mathbf{m})$$

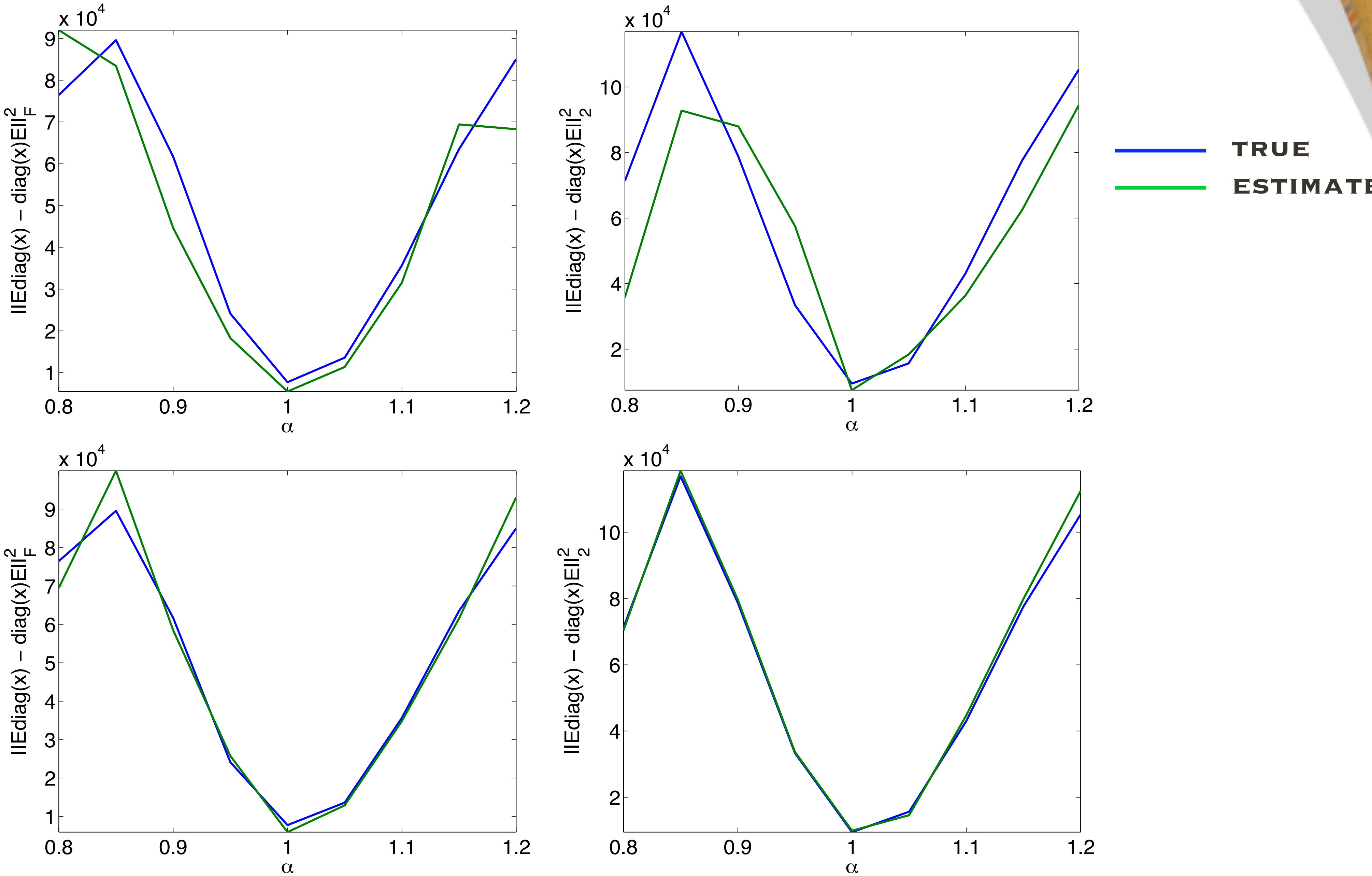
$$DR(\mathbf{m}, \mathbf{w}) = \frac{\partial R\mathbf{w}}{\partial \mathbf{m}}$$

MVA

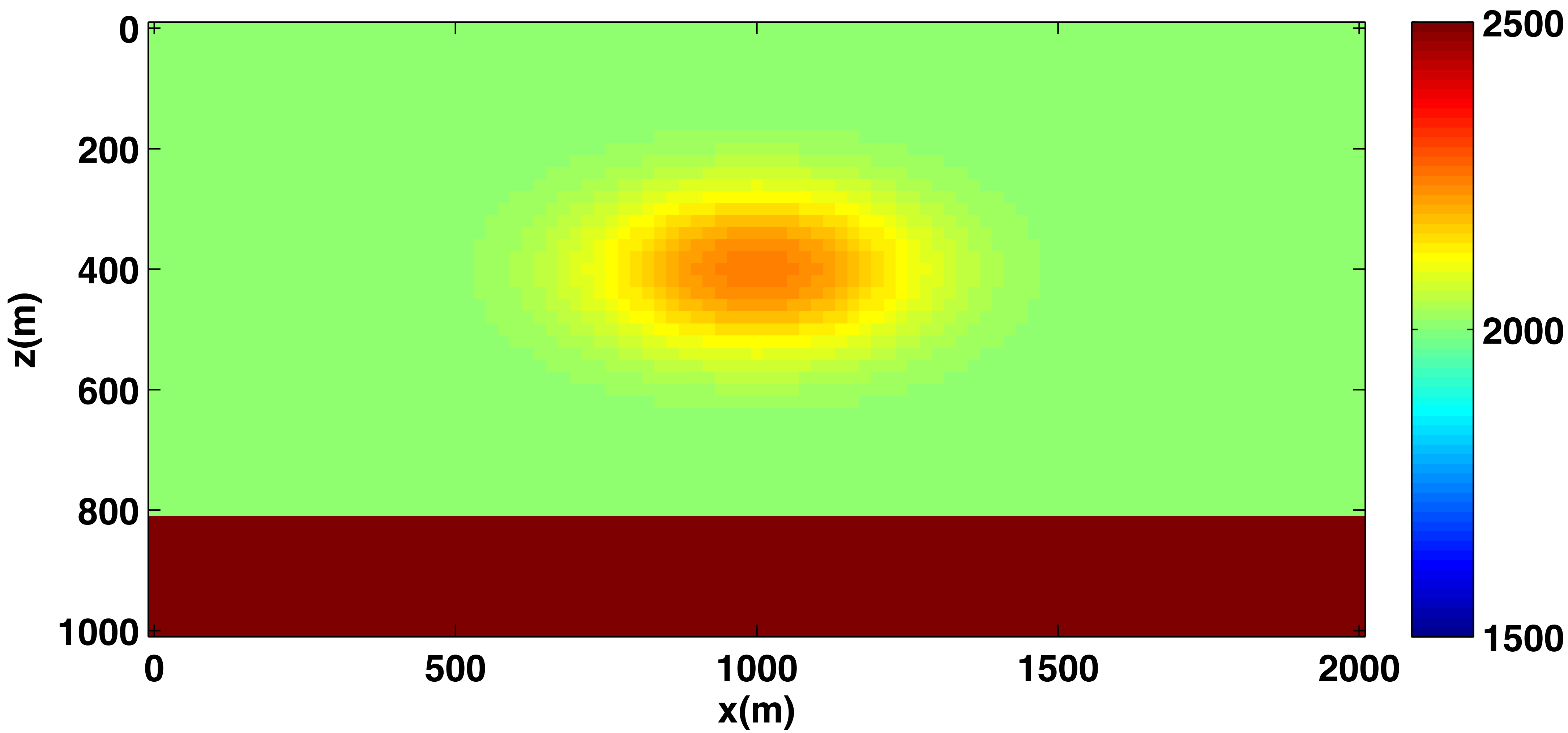
$$|| \cdot ||_F^2 \qquad\qquad || \cdot ||_2^2$$

$K = 2$

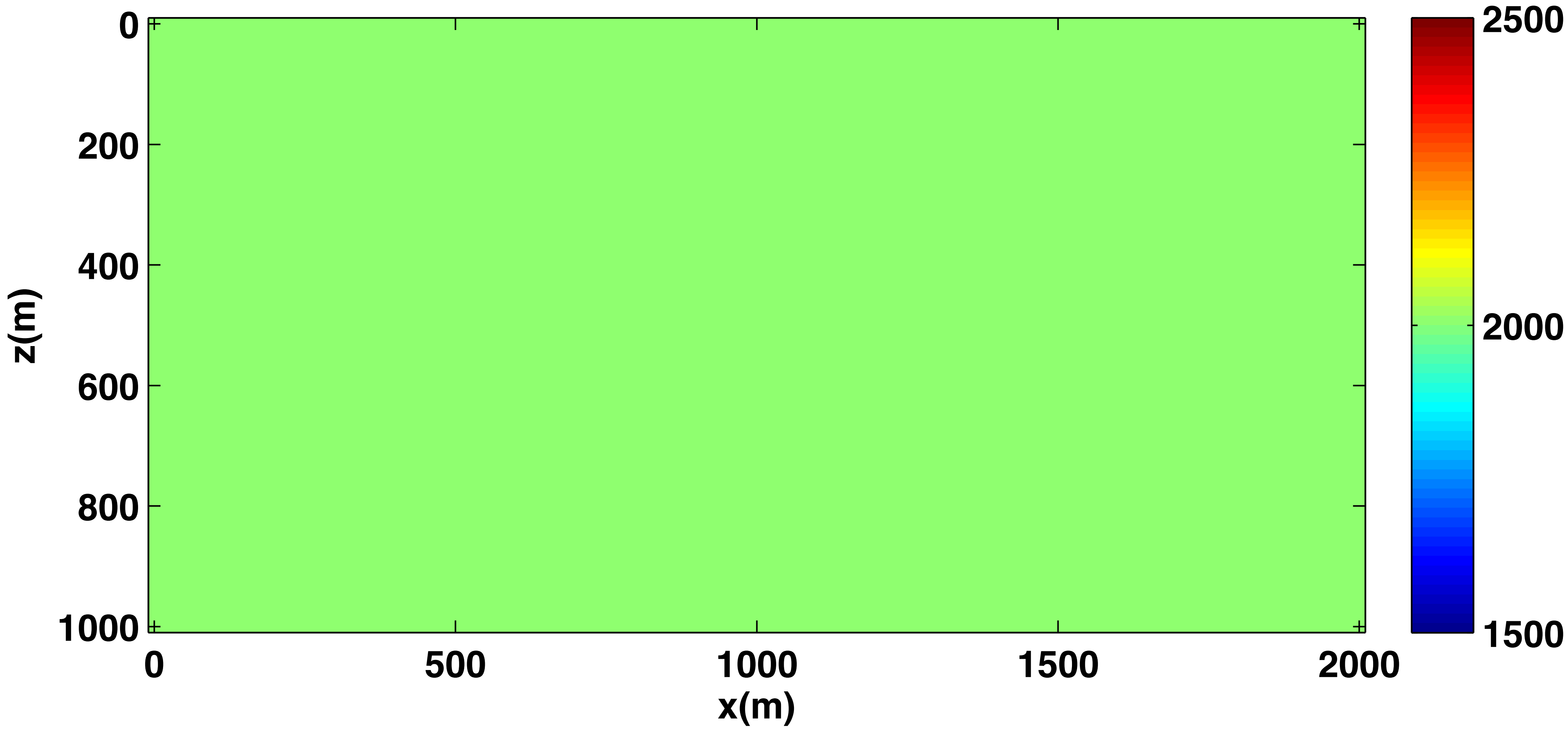
$K = 10$



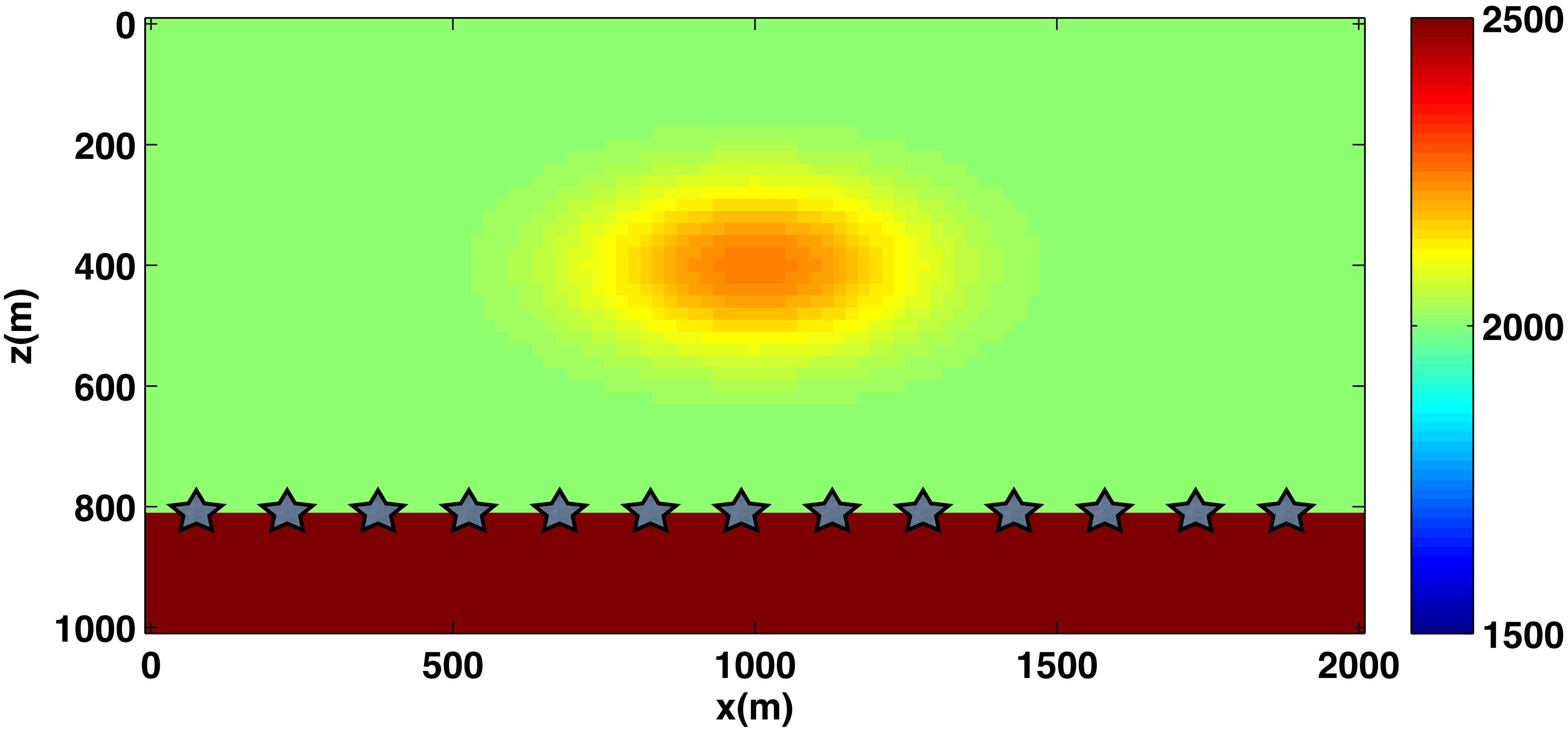
Lens Model



Initial Model

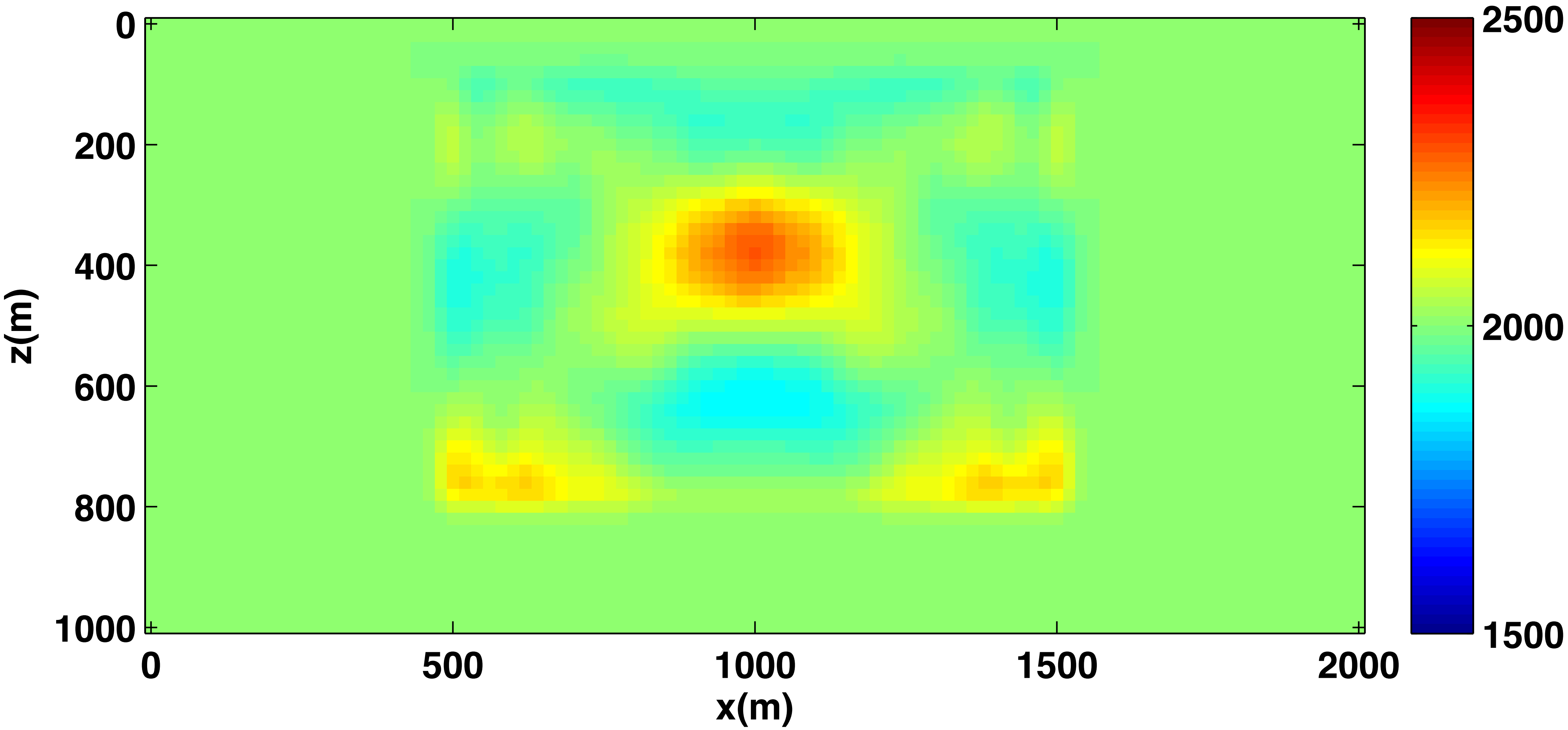


Lens Model

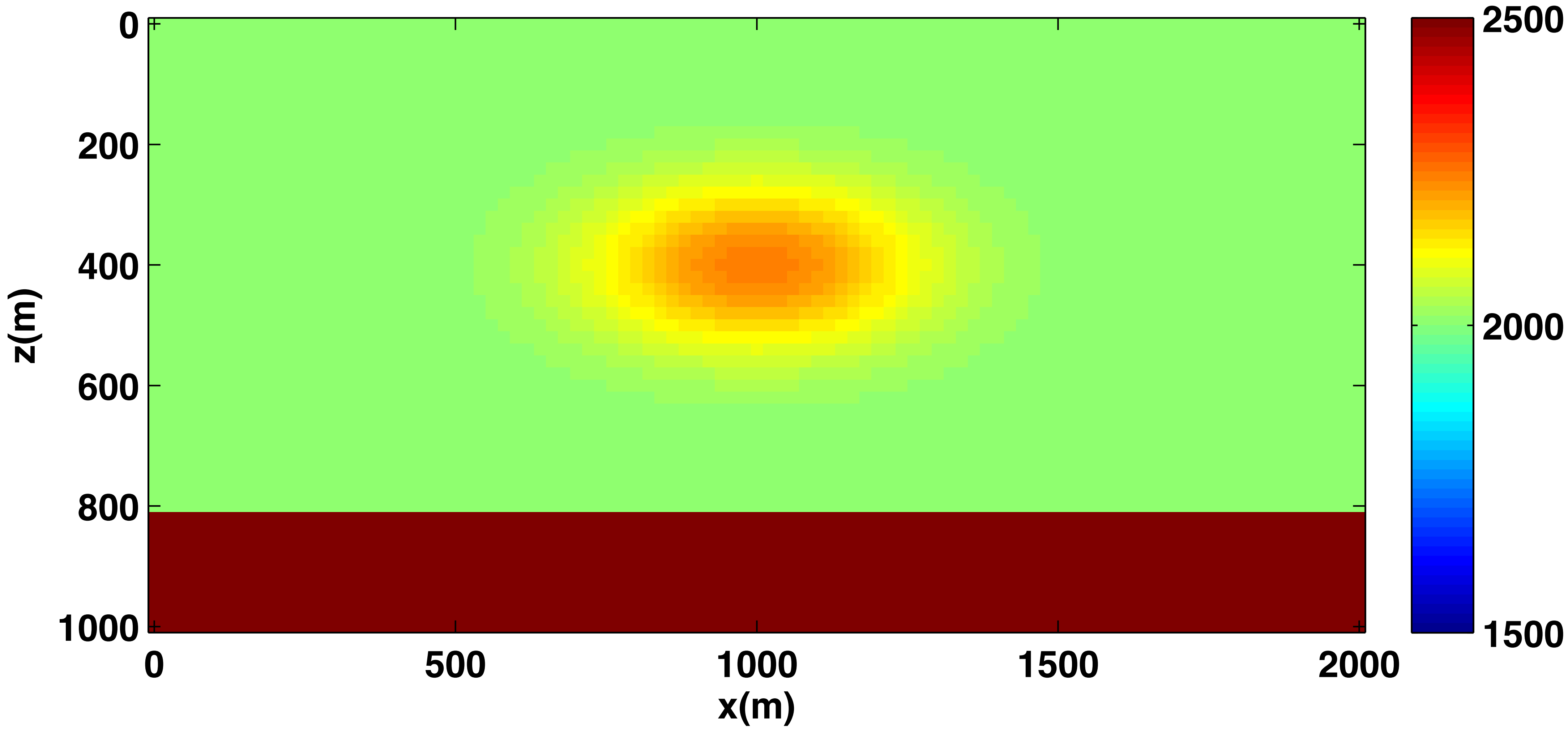


★ **represents common image points**

MVA

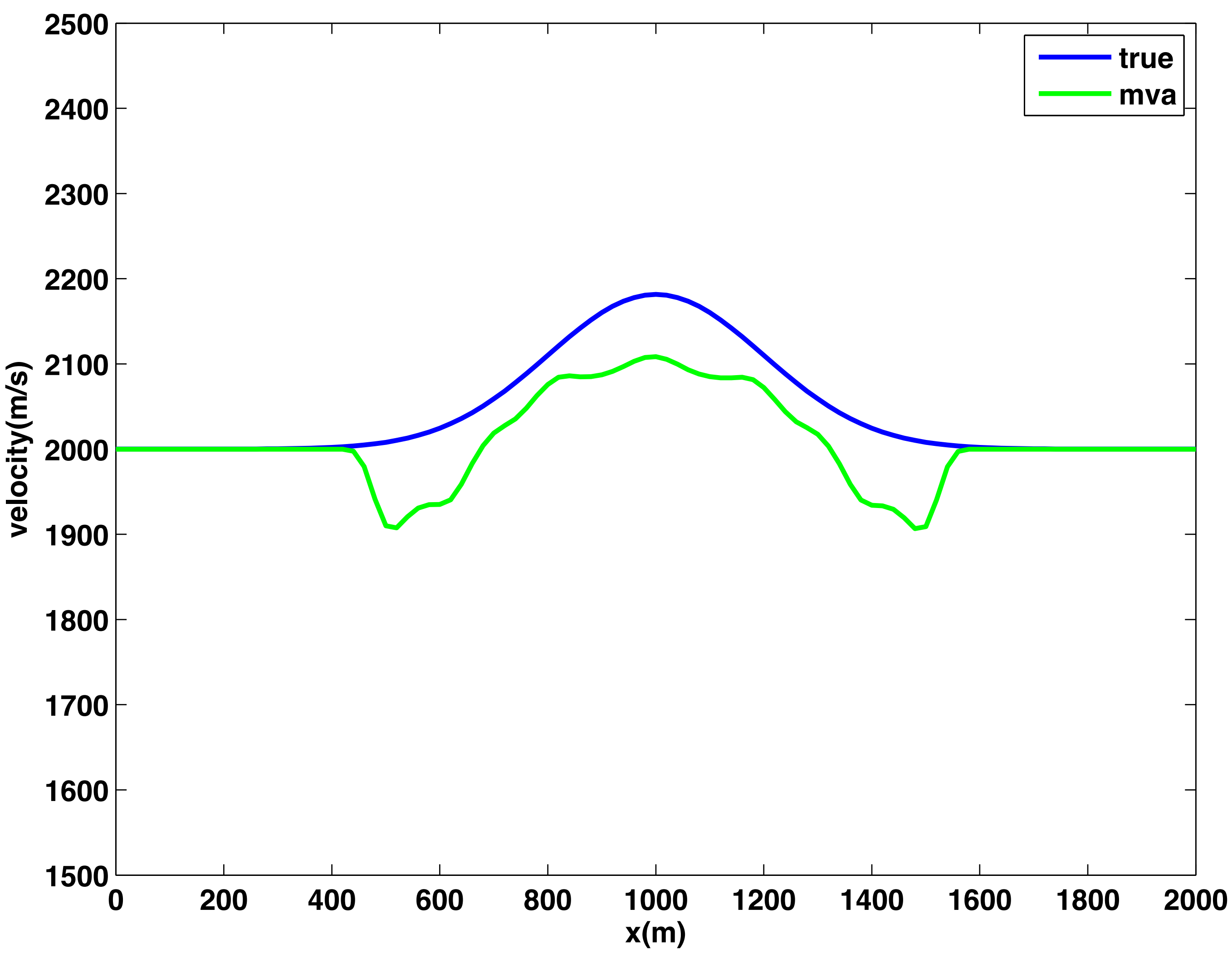


Lens Model

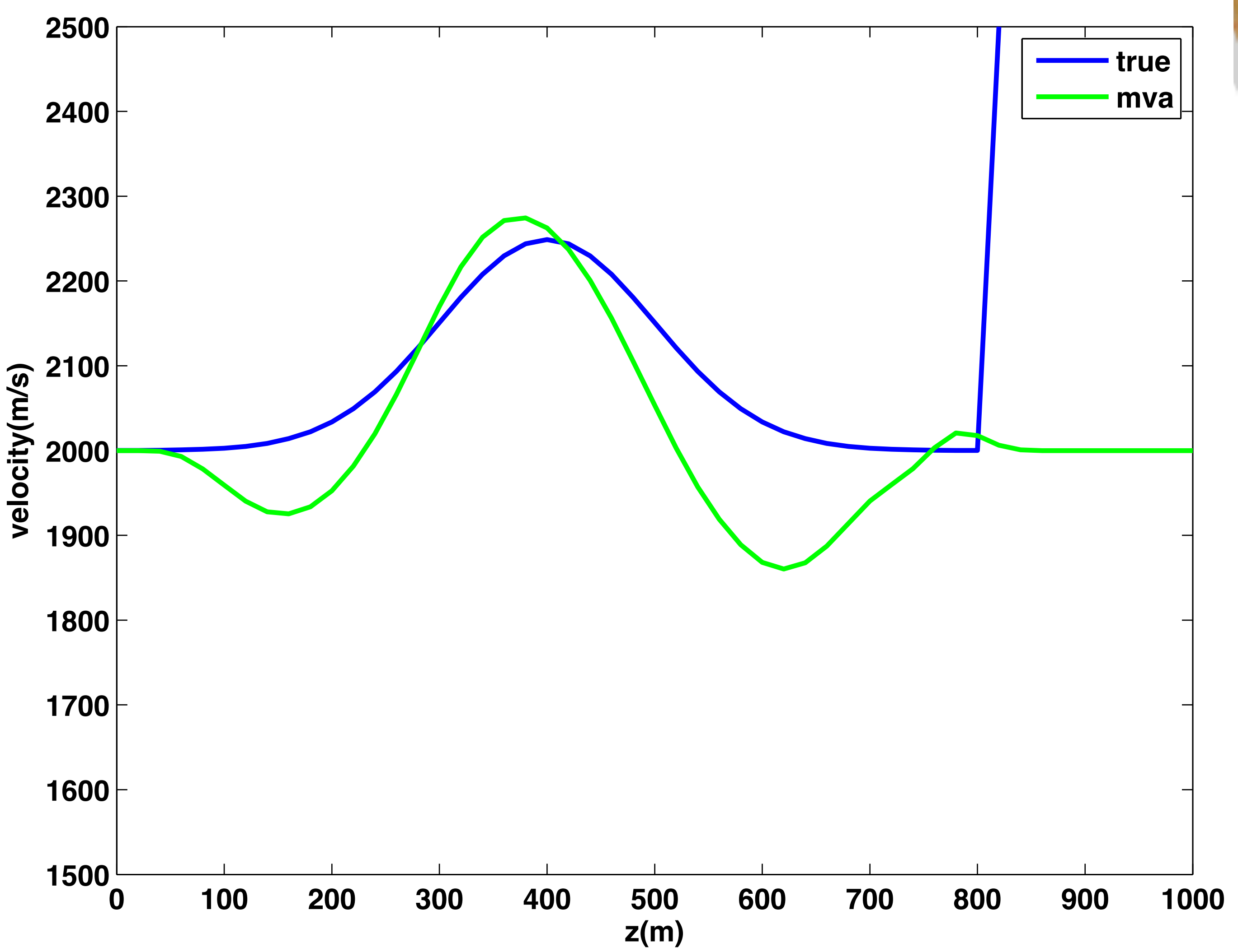


MVA

horizontal trace



vertical trace



SLIM

Summary

- Use *full* subsurface offsets, no need to estimate *dips* a *priori*
- *Probe* image volume with *mat-vecs*
- estimate dip *automatically*
- *Suitable* for AVA
- Use techniques form *randomized* trace estimation to compute MVA penalty