Random boundary for low-frequency wave propagation

Xukai Shen, Robert Clapp

SEP143 Pg 85 - 92
June 7th, 2011

Outline

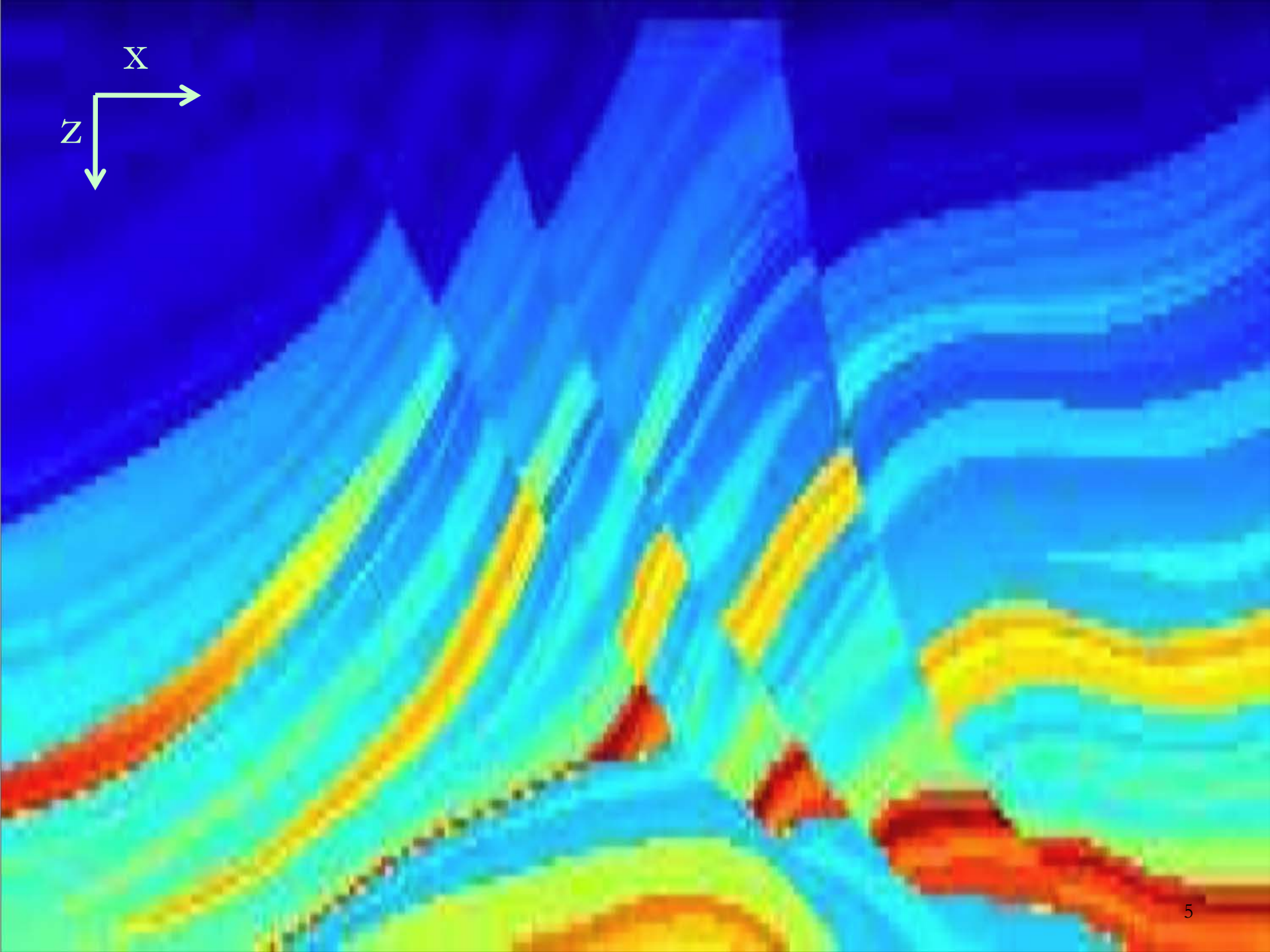
1. Motivation
2. Design of random boundary for low frequency
3. Examples

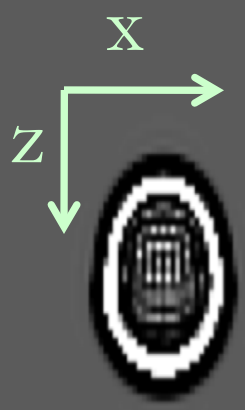
Outline

1. Motivation
2. Design of random boundary for low frequency
3. Examples

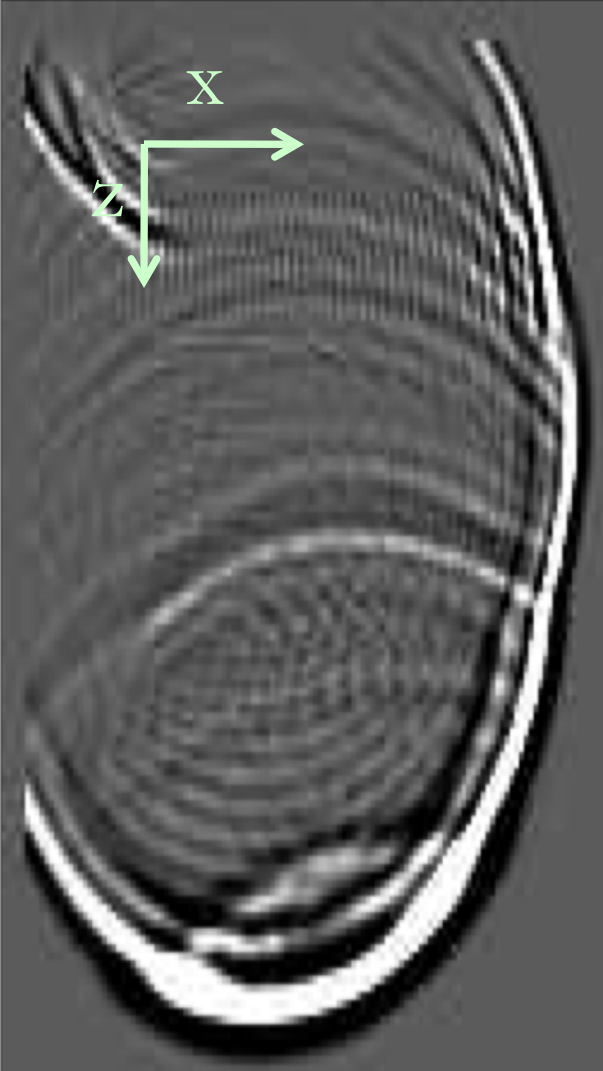
Reverse Time Migration Image calculation:

Source propagation , record source wavefield $\mathbf{U}_s$

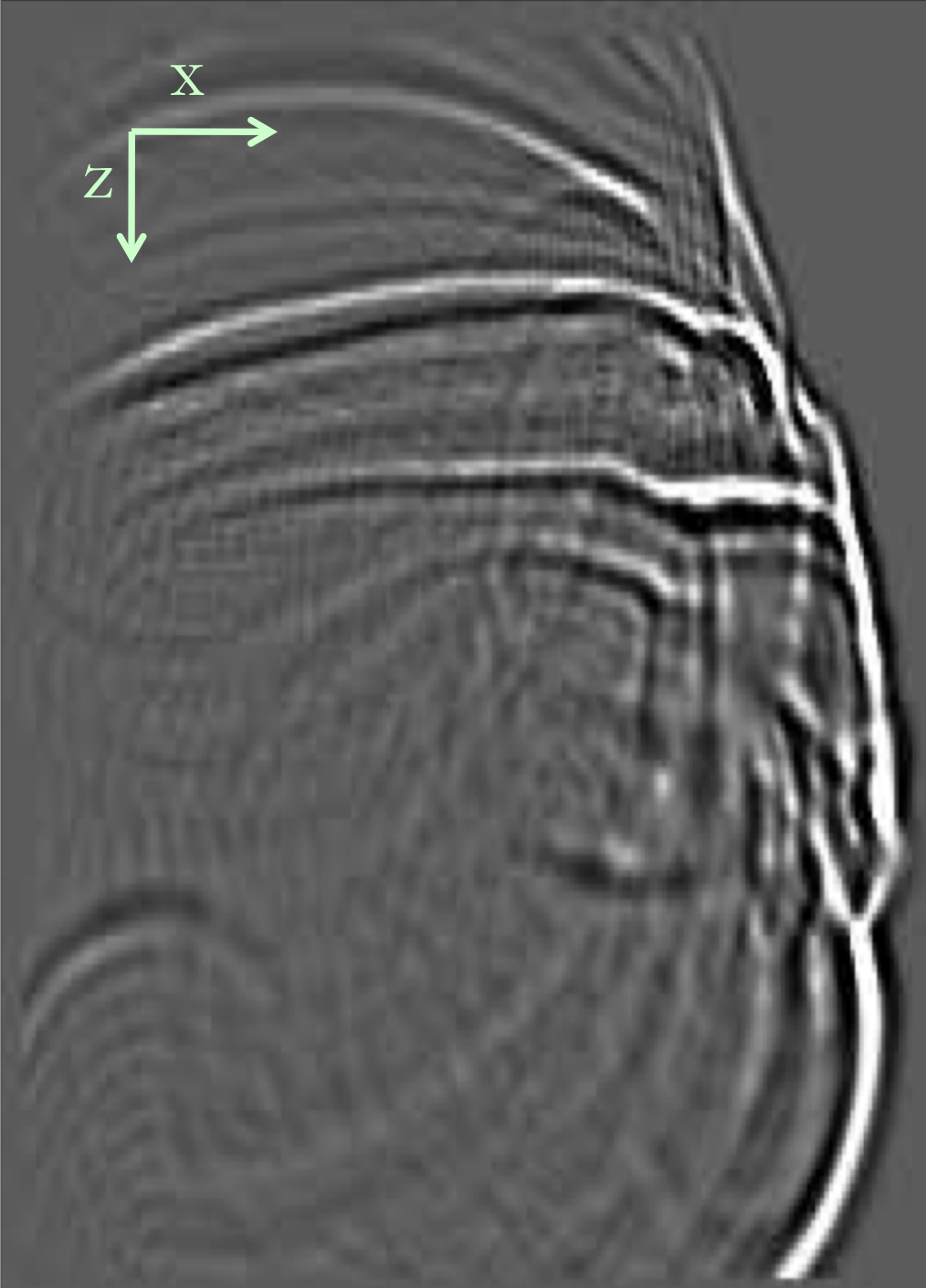

X
Z



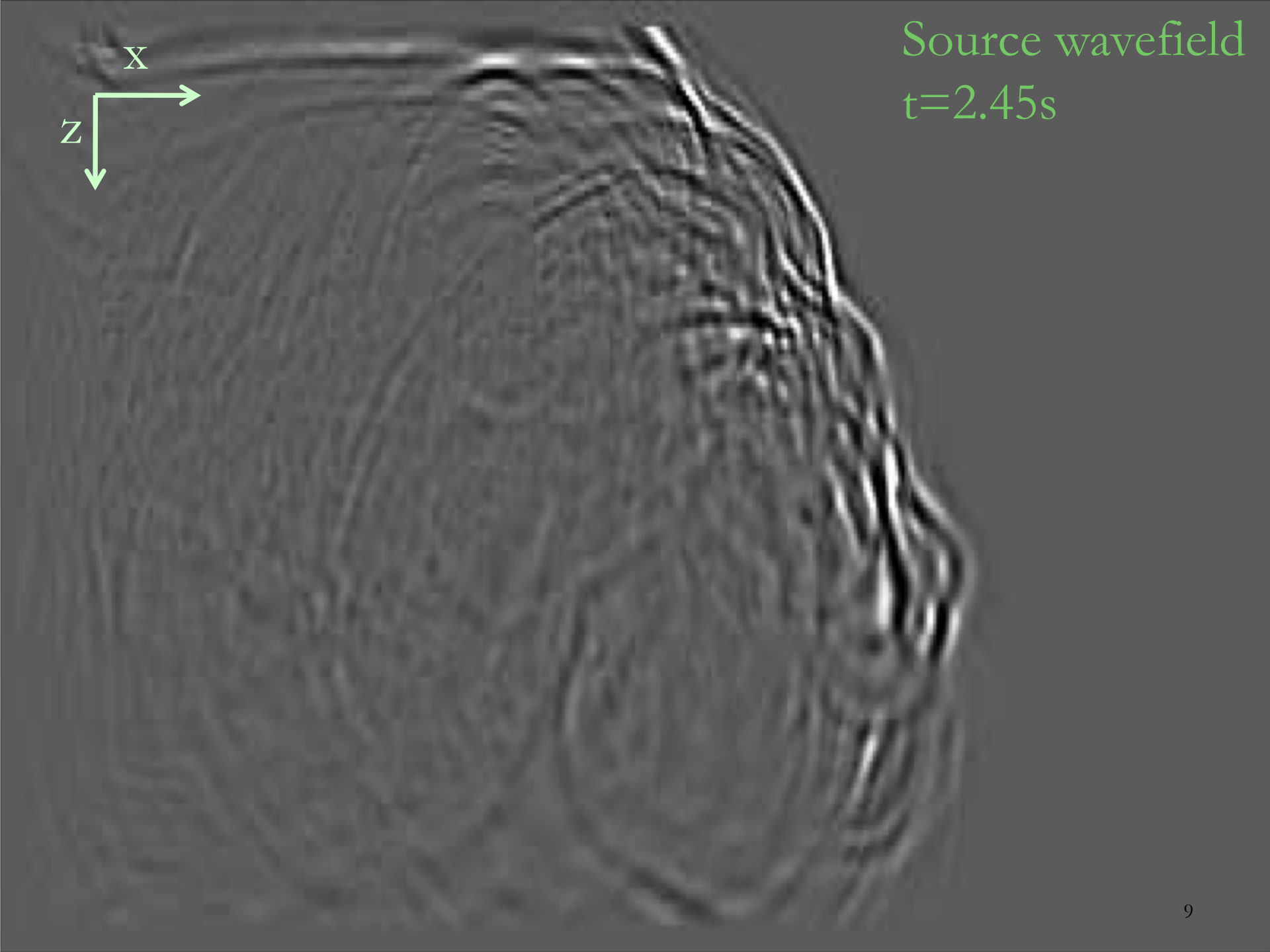
Source wavefield
t=0.35s



Source wavefield
t=1.05s

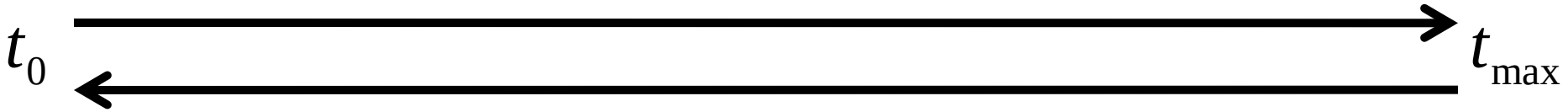


Source wavefield
t=1.75s

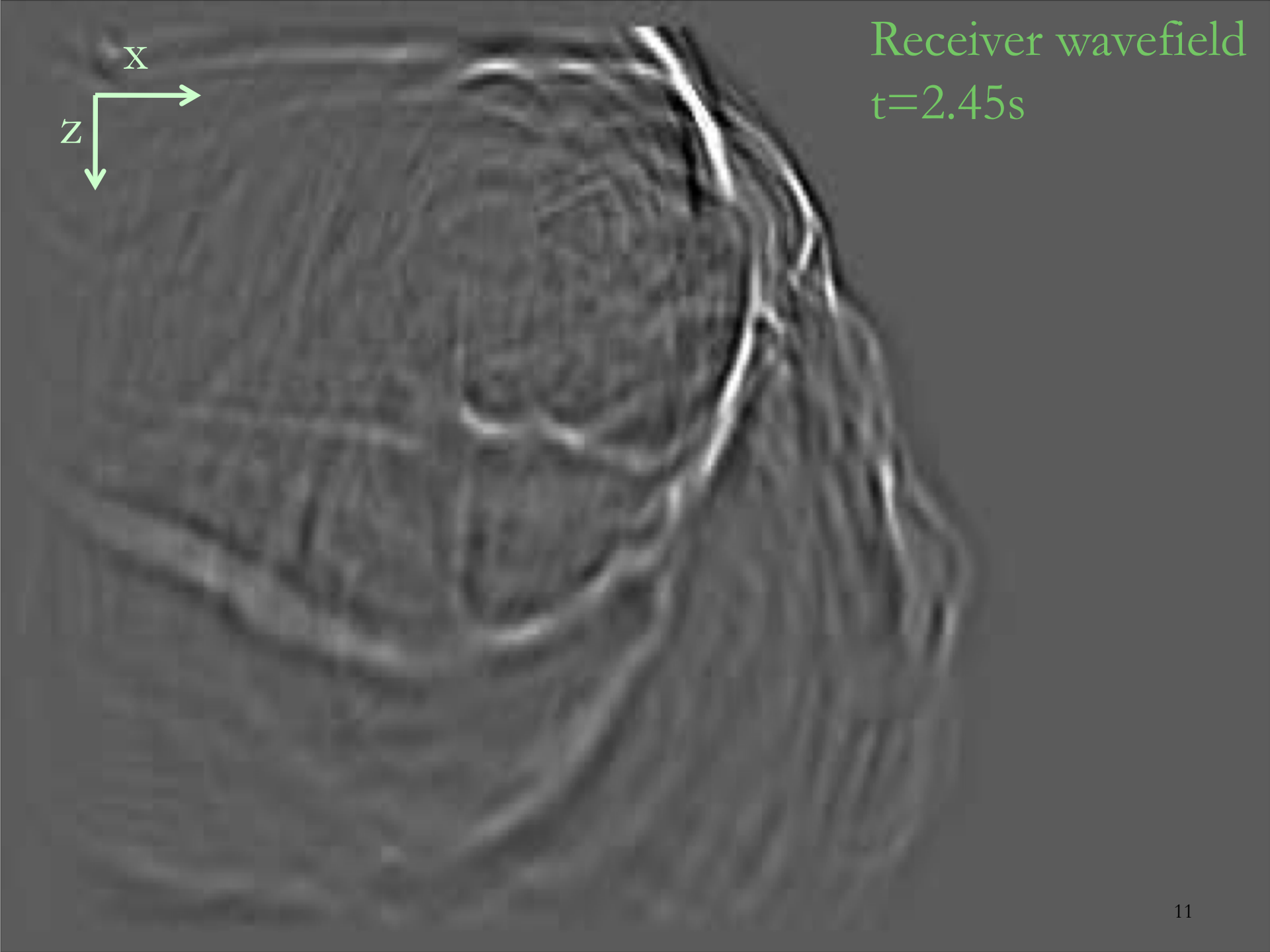

x
z
Source wavefield
t=2.45s

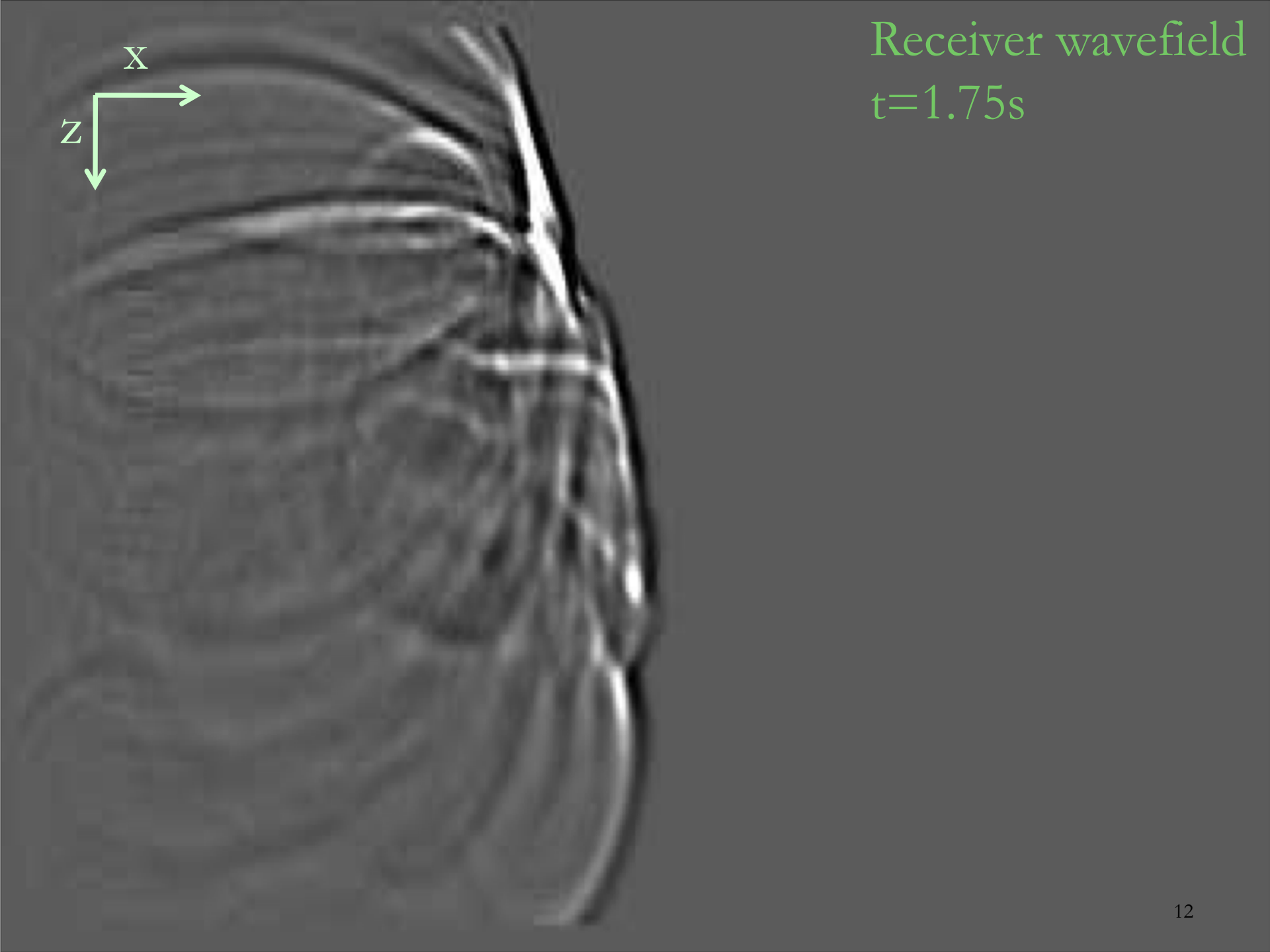
Reverse Time Migration Image calculation:

Source propagation , record source wavefield $\mathbf{U}_s$

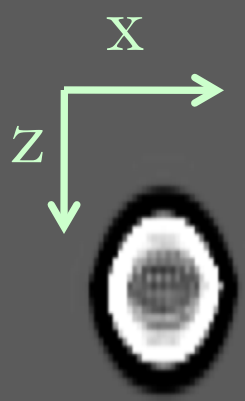


Data propagation, record receiver wavefield $\mathbf{U}_r$, correlate with $\mathbf{U}_s$


x
z
Receiver wavefield
t=2.45s


x
z
Receiver wavefield
t=1.75s

# Receiver wavefield
t=1.05s



# Receiver wavefield
t=0.35s

One shot image

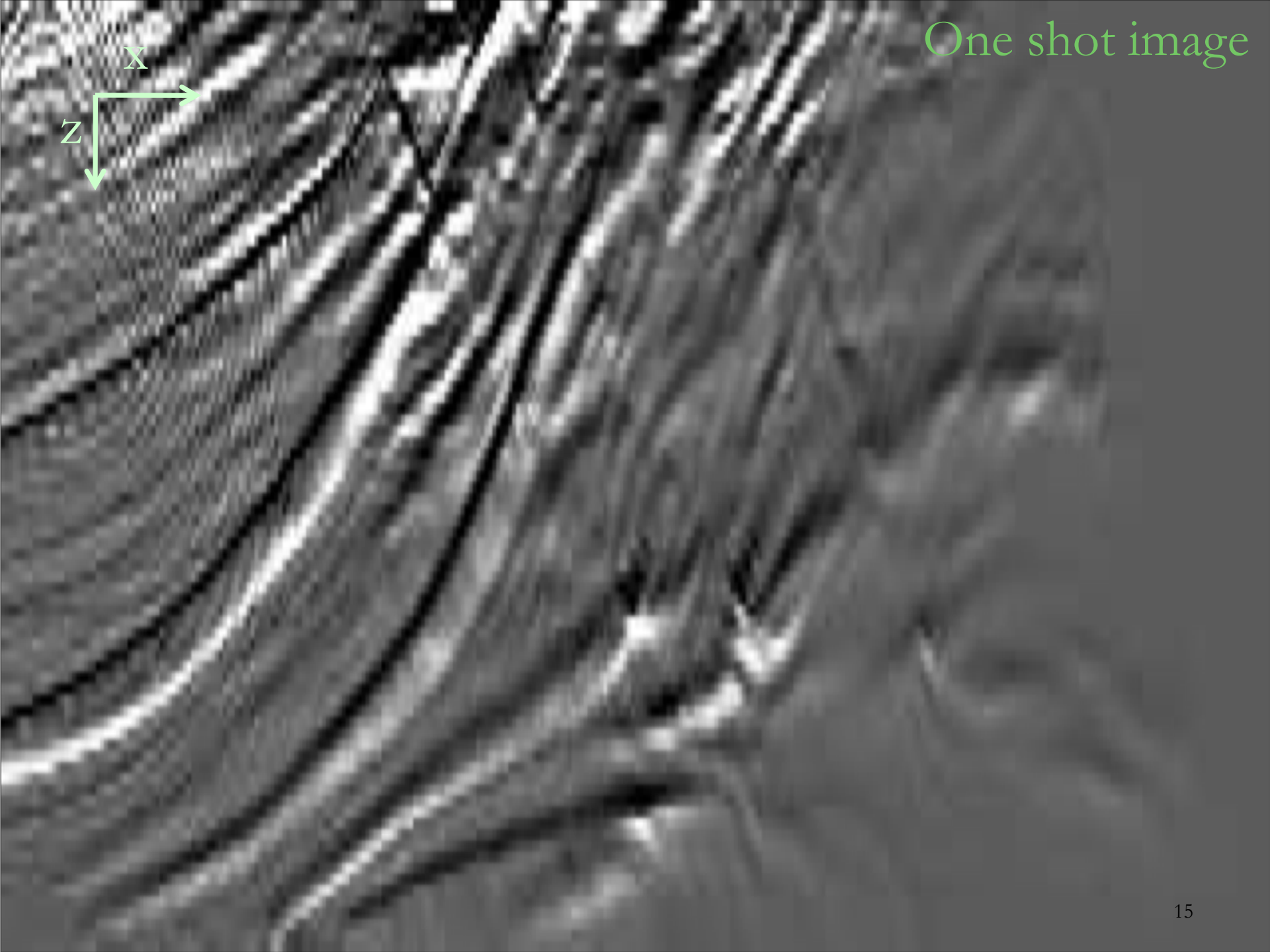


Image: $$\mathbf{g}(\mathbf{x}) = \mathbf{U_s}(\mathbf{x},t) * \mathbf{U_r}(\mathbf{x},t)$$

$\mathbf{U_r}$ Data residual wavefield

$\mathbf{U_s}$ Source wavefield

$\mathbf{x}$ Spatial location (x,y,z)

Image: $\mathbf{g(x) = U_s(x,t) * U_r(x,t)}$

$\mathbf{U_r}$ Data residual wavefield

$\mathbf{U_s}$ Source wavefield

$\mathbf{x}$ Spatial location (x,y,z)

Conventional:

Save at least one wavefield

e.g. nx*ny*nz*nt=500*500*500*500 ~ 250G

Image: $\mathbf{g}(\mathbf{x}) = \mathbf{U_s}(\mathbf{x},t) * \mathbf{U_r}(\mathbf{x},t)$

$\mathbf{U_r}$ Data residual wavefield

$\mathbf{U_s}$ Source wavefield

$\mathbf{x}$ Spatial location (x,y,z)

Conventional:

Save at least one wavefield

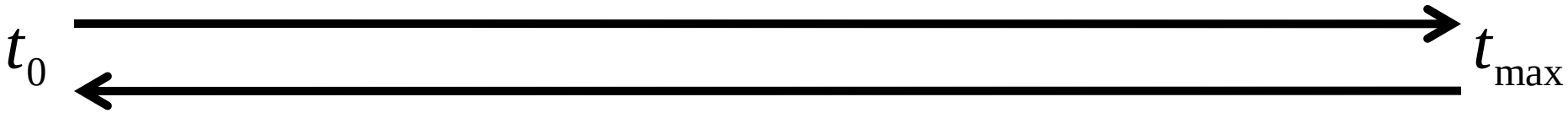
e.g. nx*ny*nz*nt=500*500*500*500 ~ 250G

Write wavefield to disk

Save everything in memory by avoiding saving wavefield movie!!!

Image calculation :

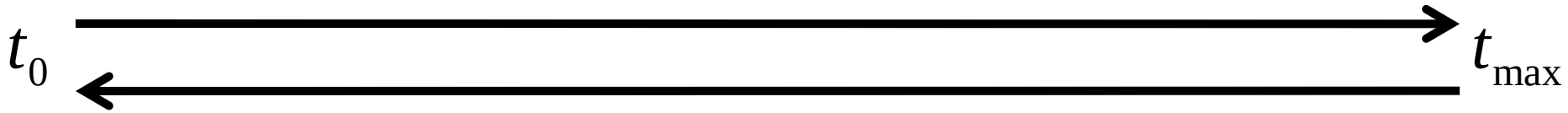
Source propagation , record source wavefield $\mathbf{U}_s$



Data propagation, record receiver wavefield $\mathbf{U}_r$, correlate with $\mathbf{U}_s$

Image calculation with random boundary:

Source propagation , record last 2 slices of $\mathbf{U}_s$



Data and source propagation, correlate $\mathbf{U}_s$ and $\mathbf{U}_r$

Image: $\mathbf{g}(\mathbf{x}) = \mathbf{U_s}(\mathbf{x},t) * \mathbf{U_r}(\mathbf{x},t)$

$\mathbf{U_r}$ Data residual wavefield

$\mathbf{U_s}$ Source wavefield

$\mathbf{x}$ Spatial location (x,y,z)

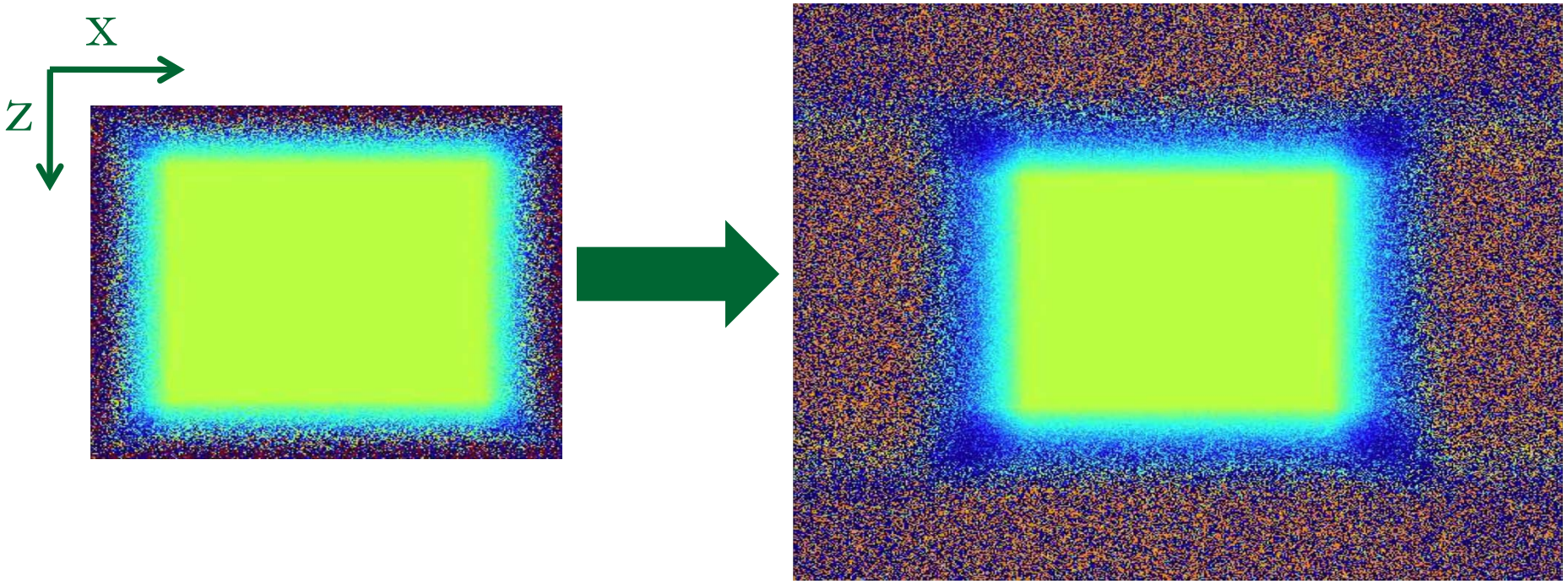
Random boundary condition:

No need to save whole wavefield

e.g. nx*ny*nz*2=500*500*500*2 ~ 1.0G

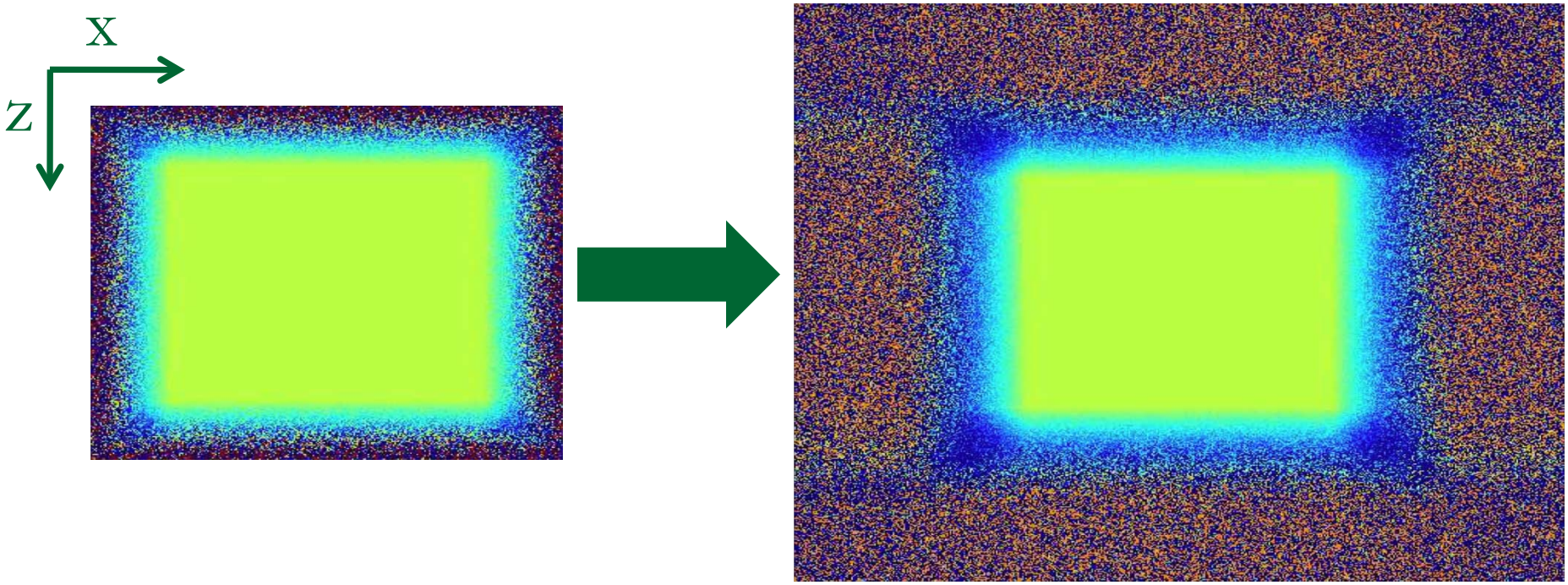
How to design random boundary for low frequency wave

How to design random boundary for low frequency wave



How to design random boundary for low frequency wave

Big increase in computation and memory



How to design **economic** random boundary for low frequency wave

RTM

FWI

Random anomalies used for RTM is too small compared to the dominant wavelength in FWI

Outline

1. Motivation
2. Design of random boundary for low frequency
3. Examples

Use bigger anomaly grains in random boundary

Use bigger anomaly grains in random boundary

1. Grain size

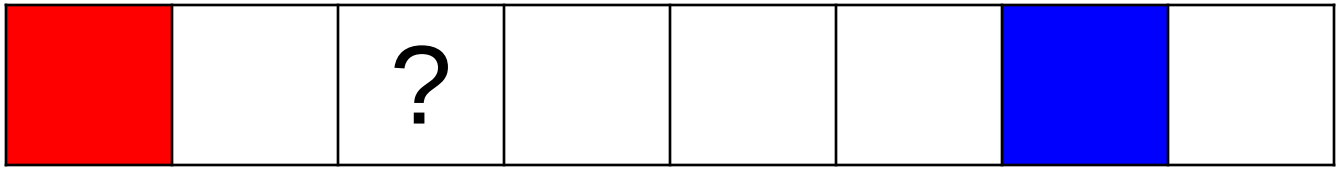
2. Grain shape

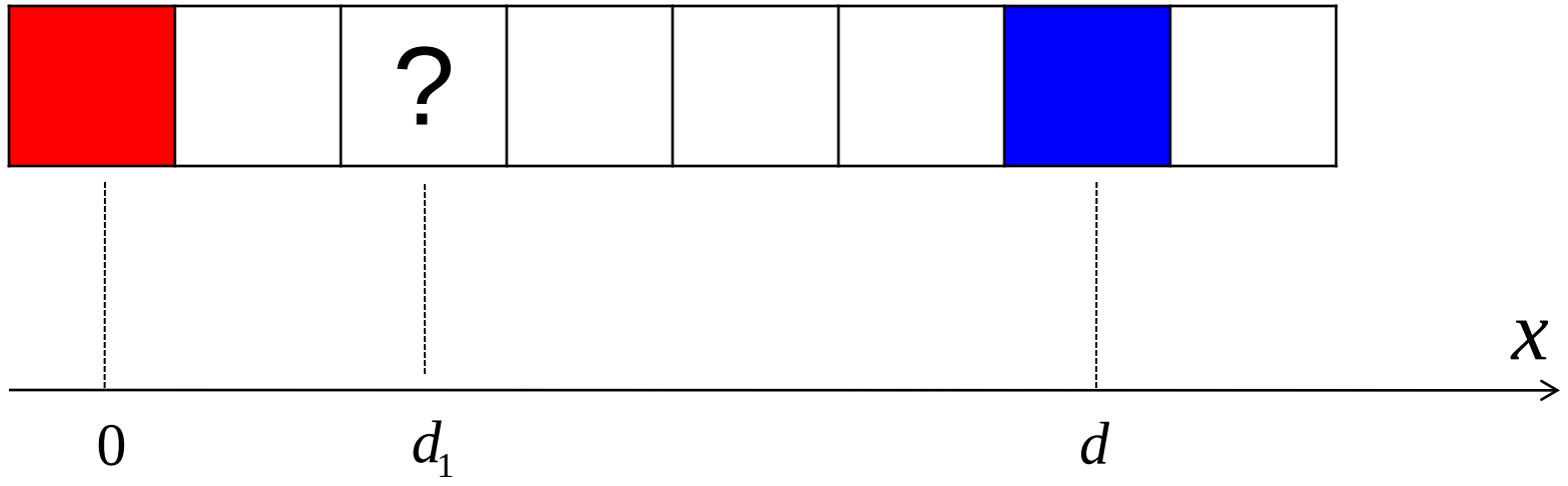
Use bigger anomaly grains in random boundary

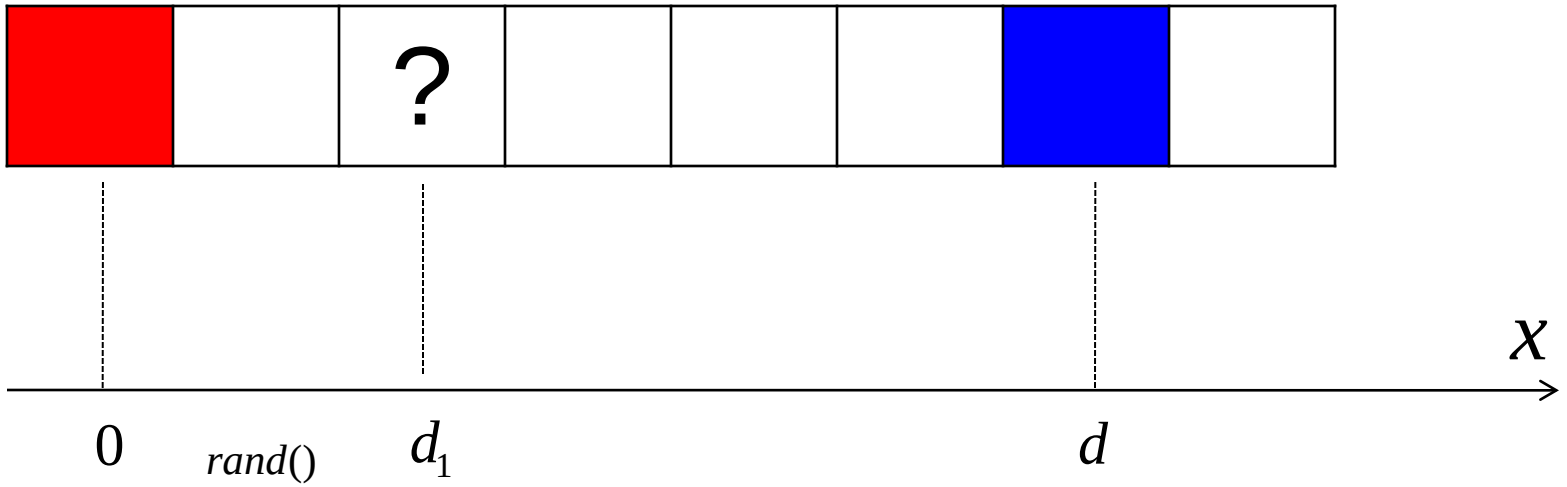
1. Grain size

2. Grain shape

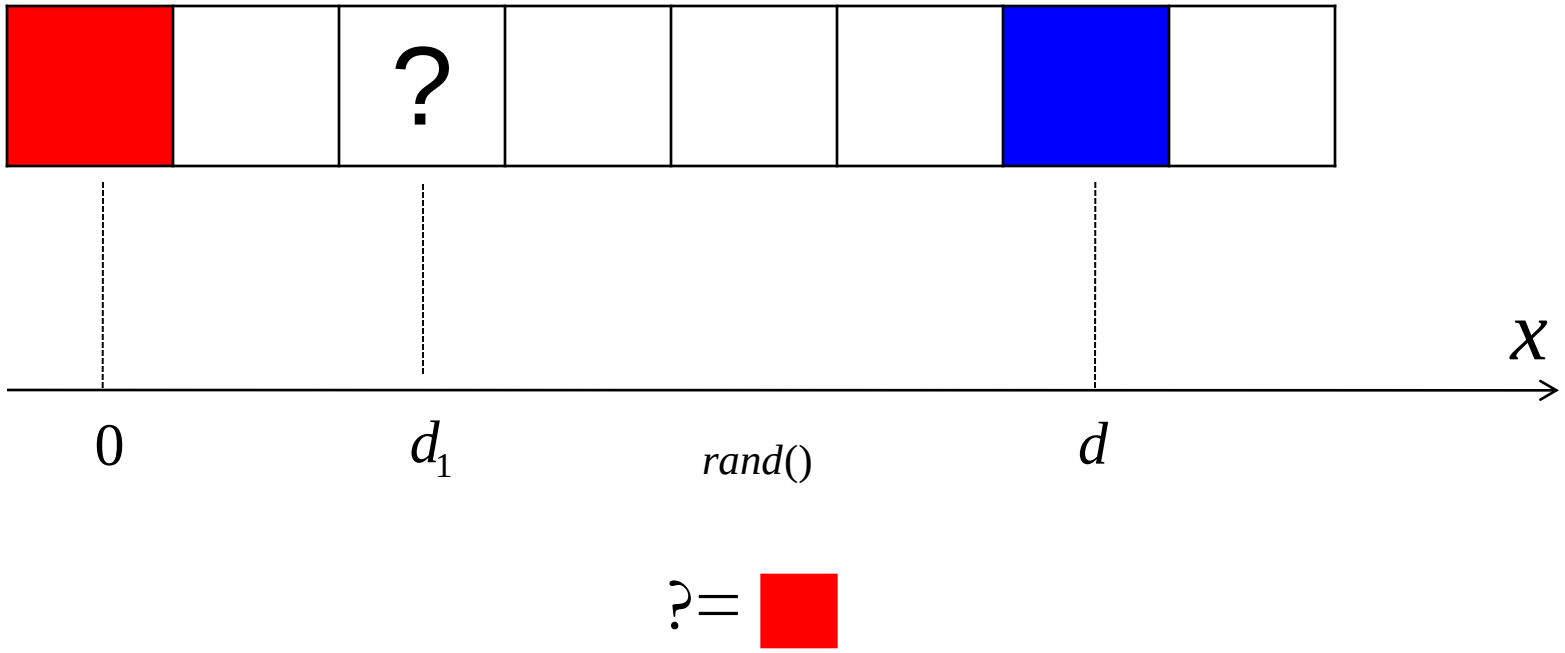
   Cubic
   Random




?
x
0
d_1
d


?
x
0
rand()
d_1
d

?=


?
x
0
d_1
rand()
d
?=

Outline

1. Motivation
2. Design random boundary for low frequency
3. Examples

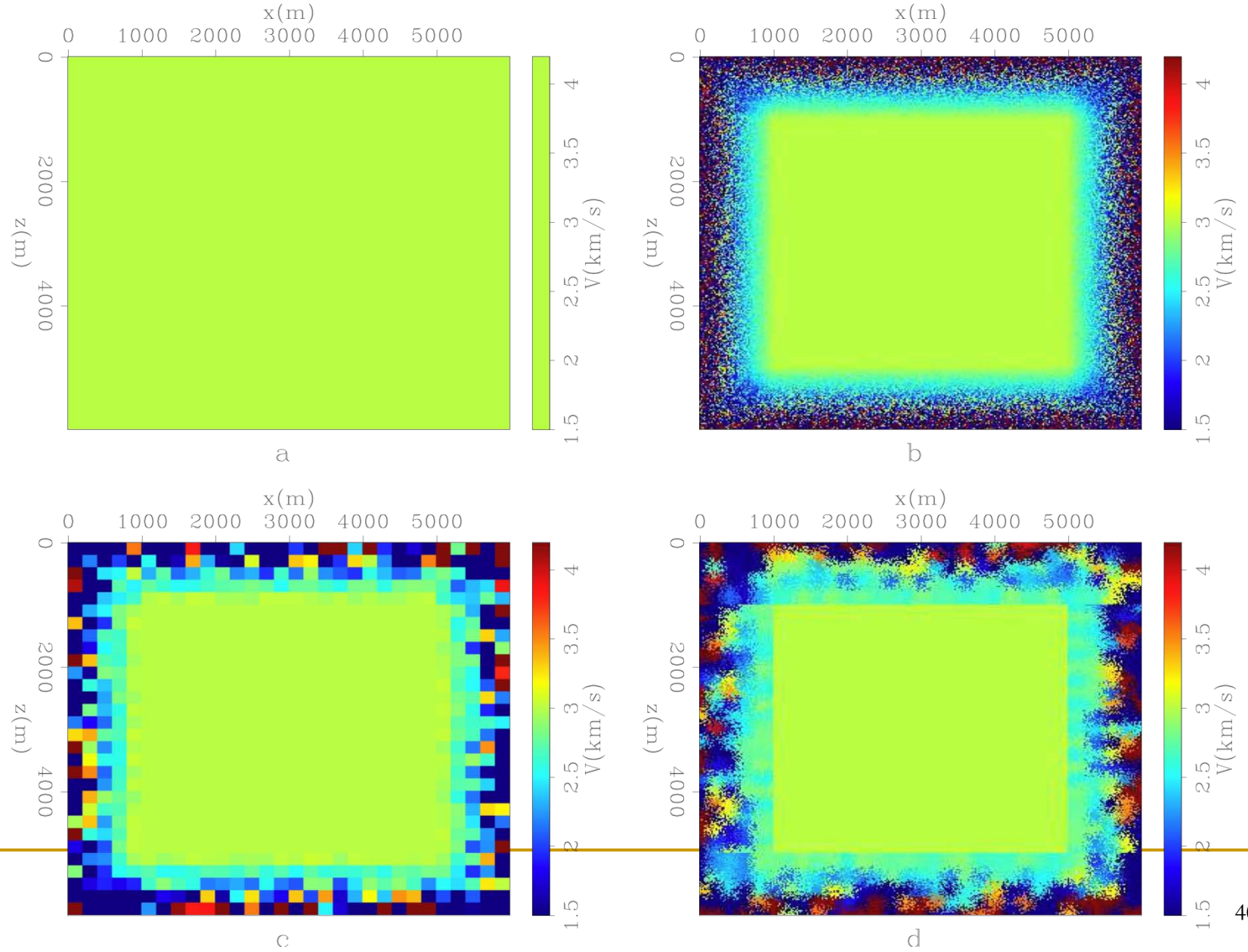
Modeling: Parameters

Full velocity mode: 300*300*300 in grids
20m grid spacing in all directions

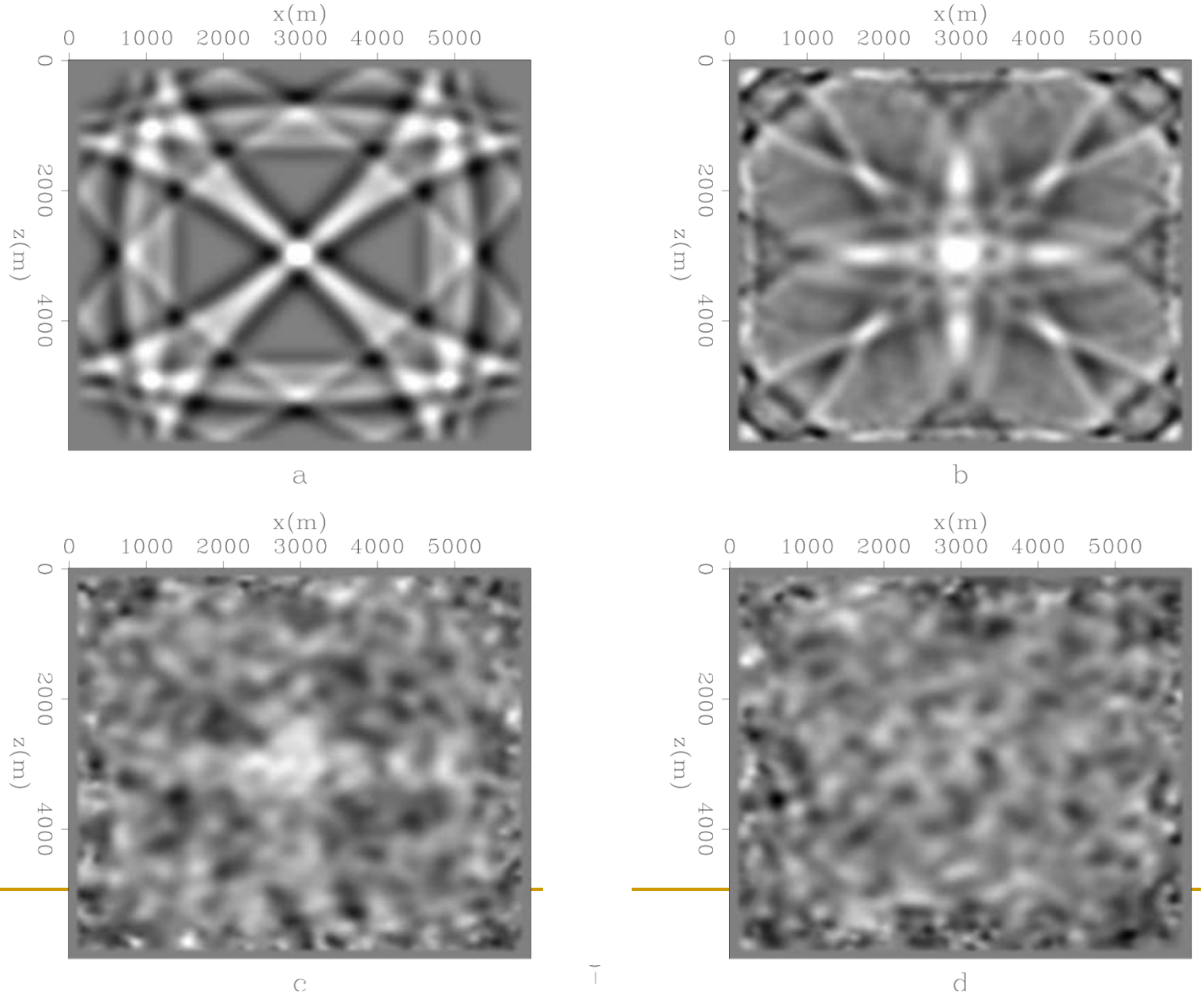
Boundary region: 50 grids on each side

Velocity: mean 3km/s, min 1.5km/s, max 4.2km/s

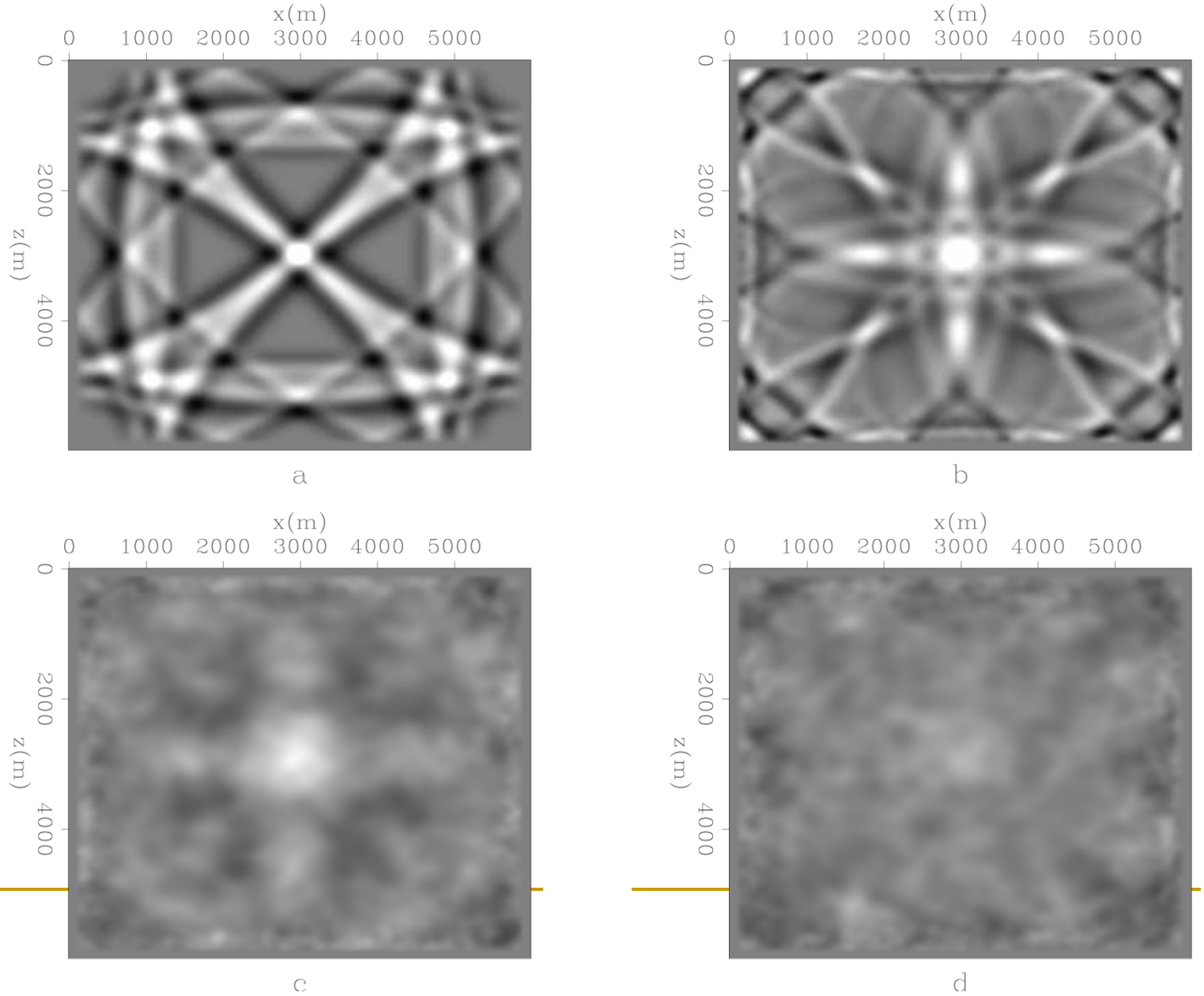
Modeling: Velocity



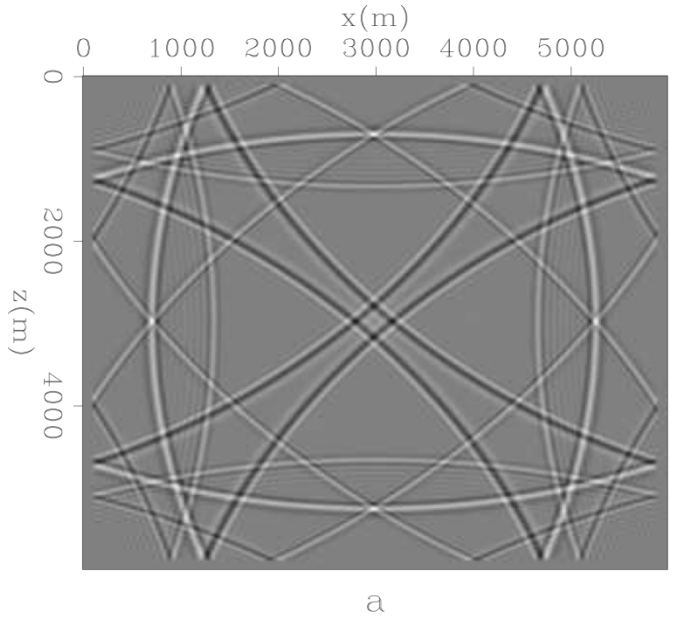
Modeling: Low frequency (6Hz peak)



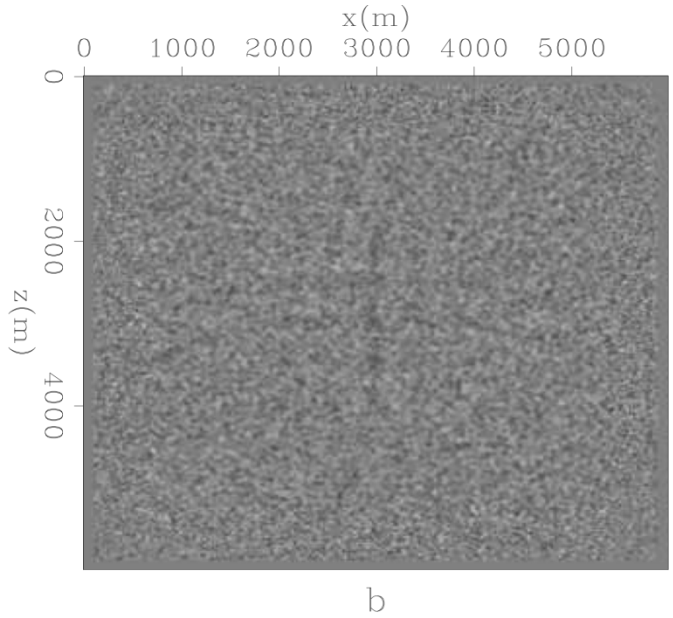
Modeling: Low frequency (6Hz peak)



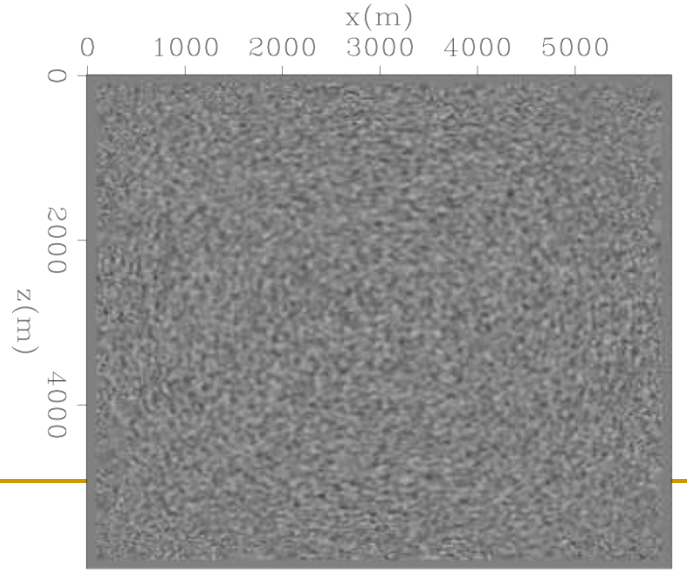
Modeling: Broadband (25Hz peak)



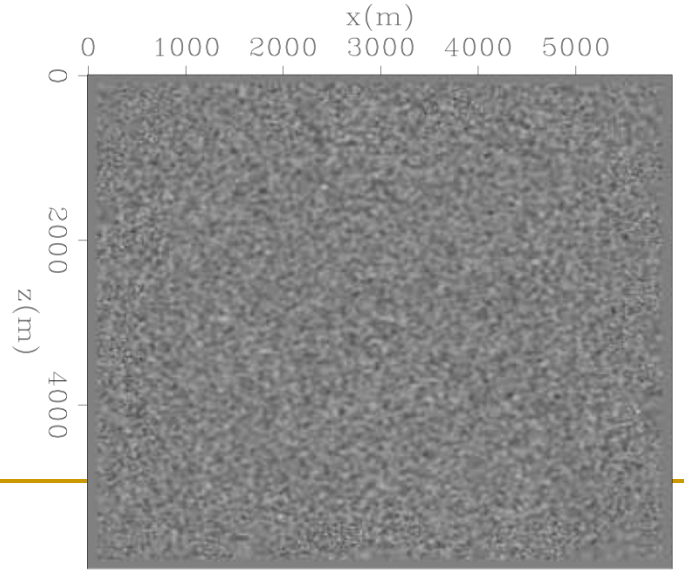
a



b

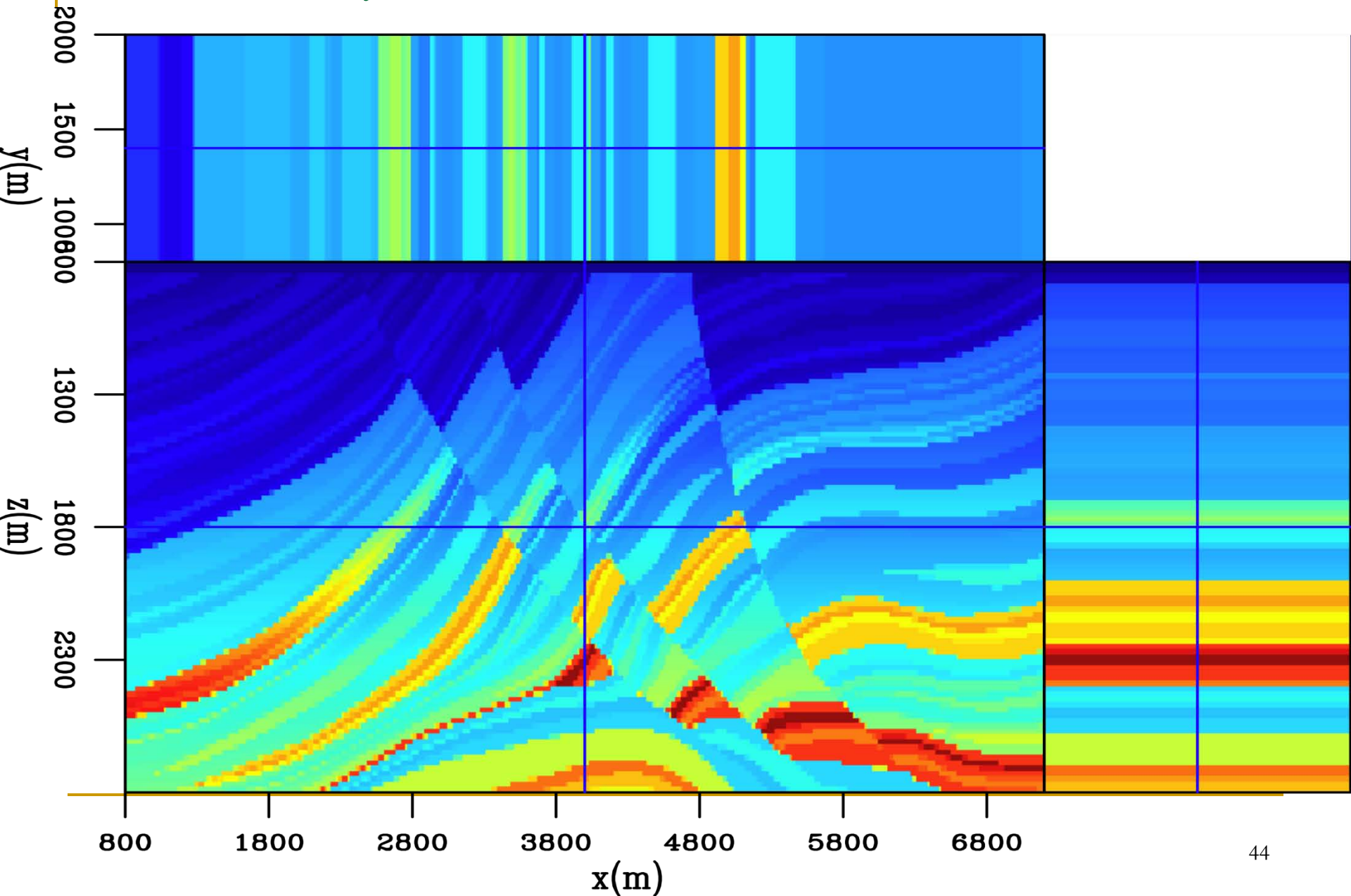


c

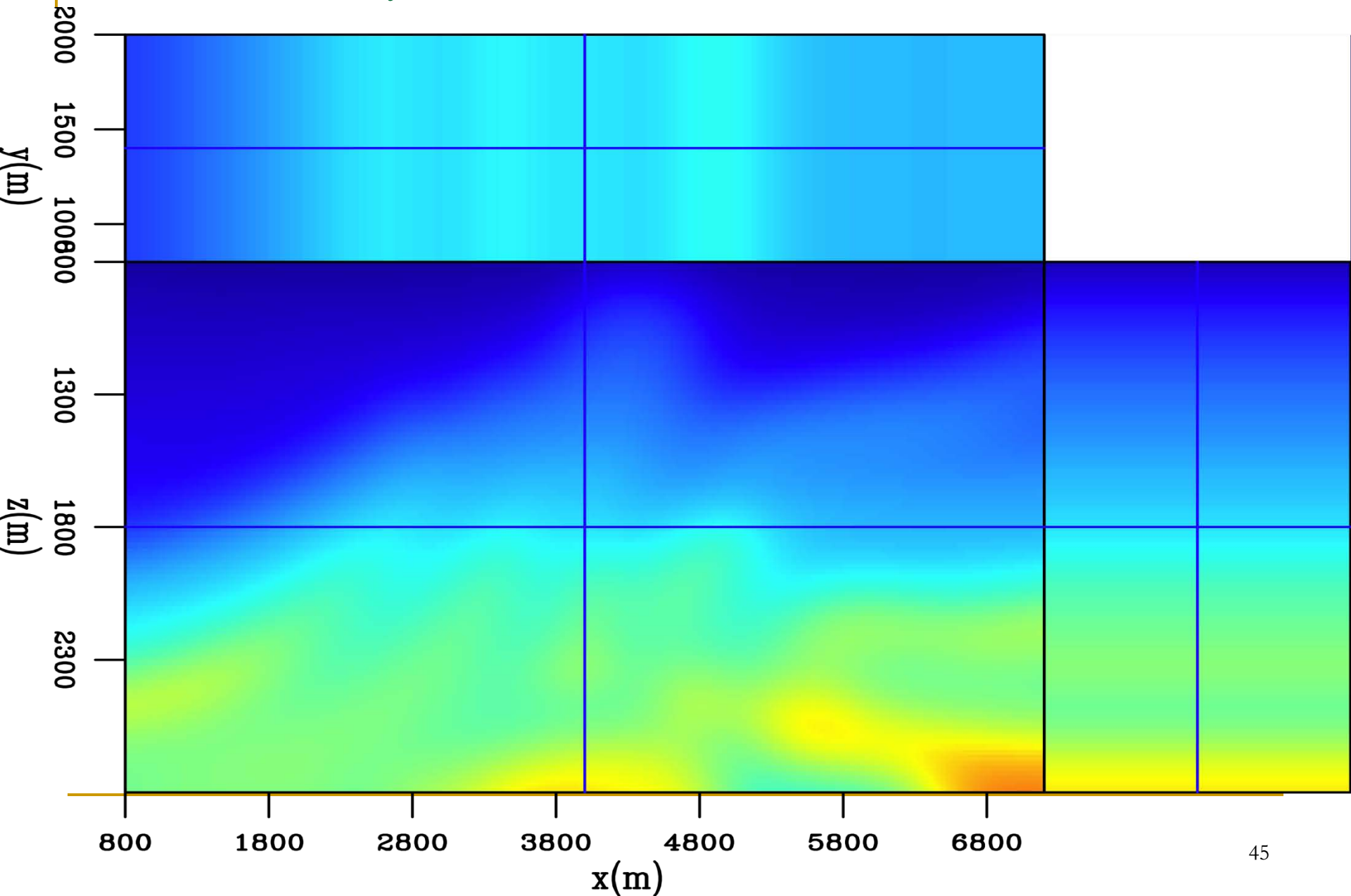


d

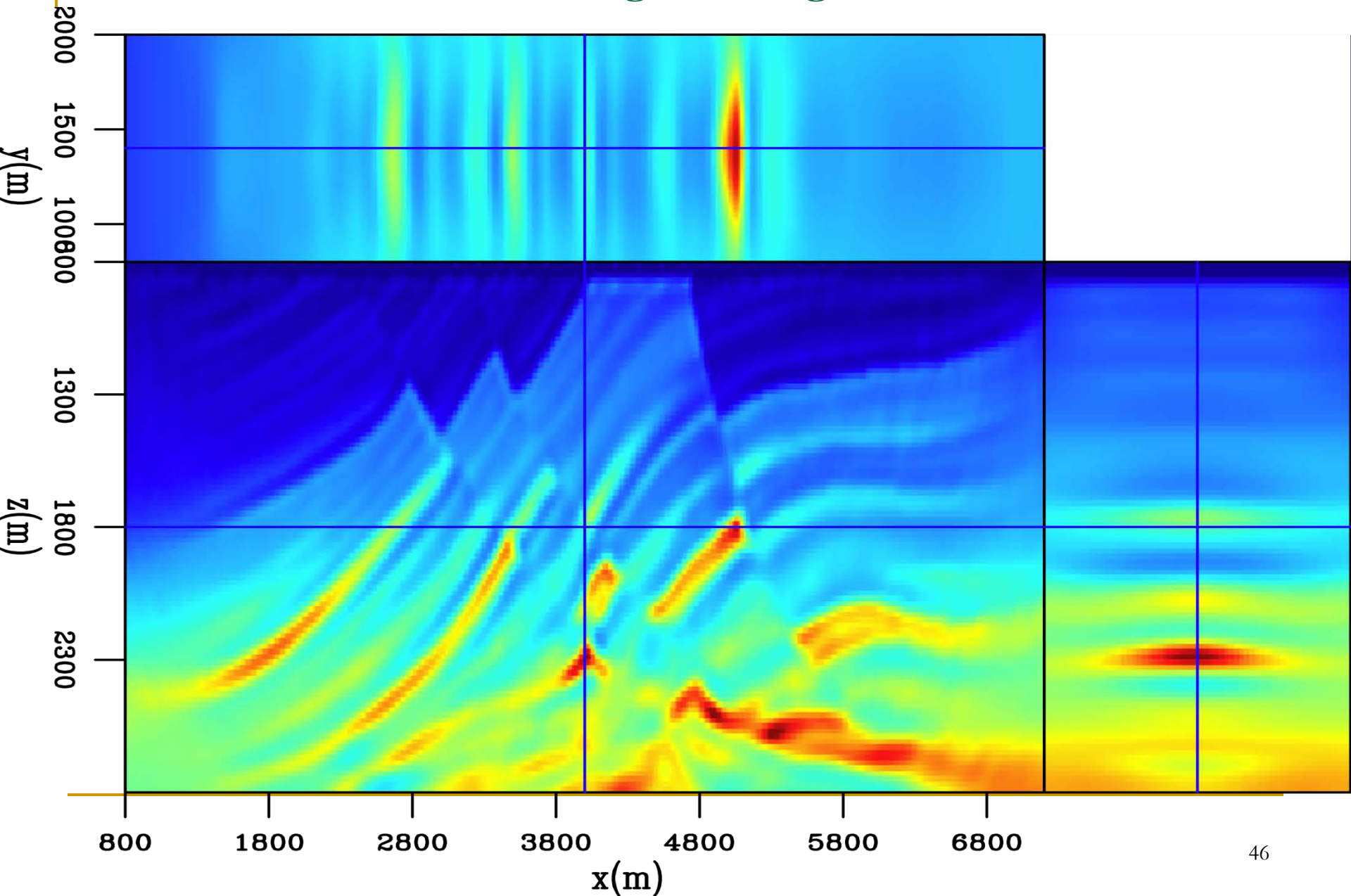
True velocity



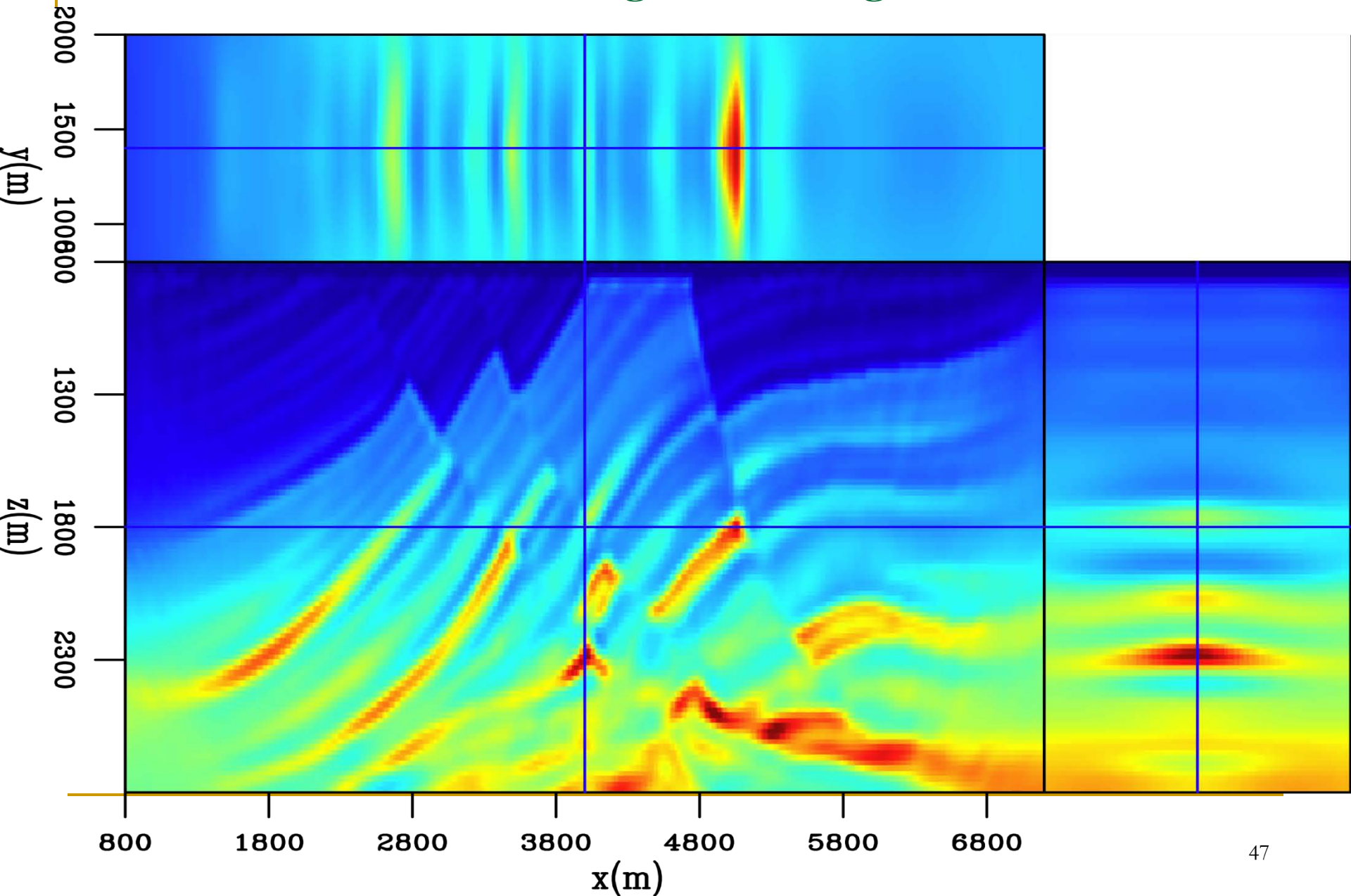
Initial velocity



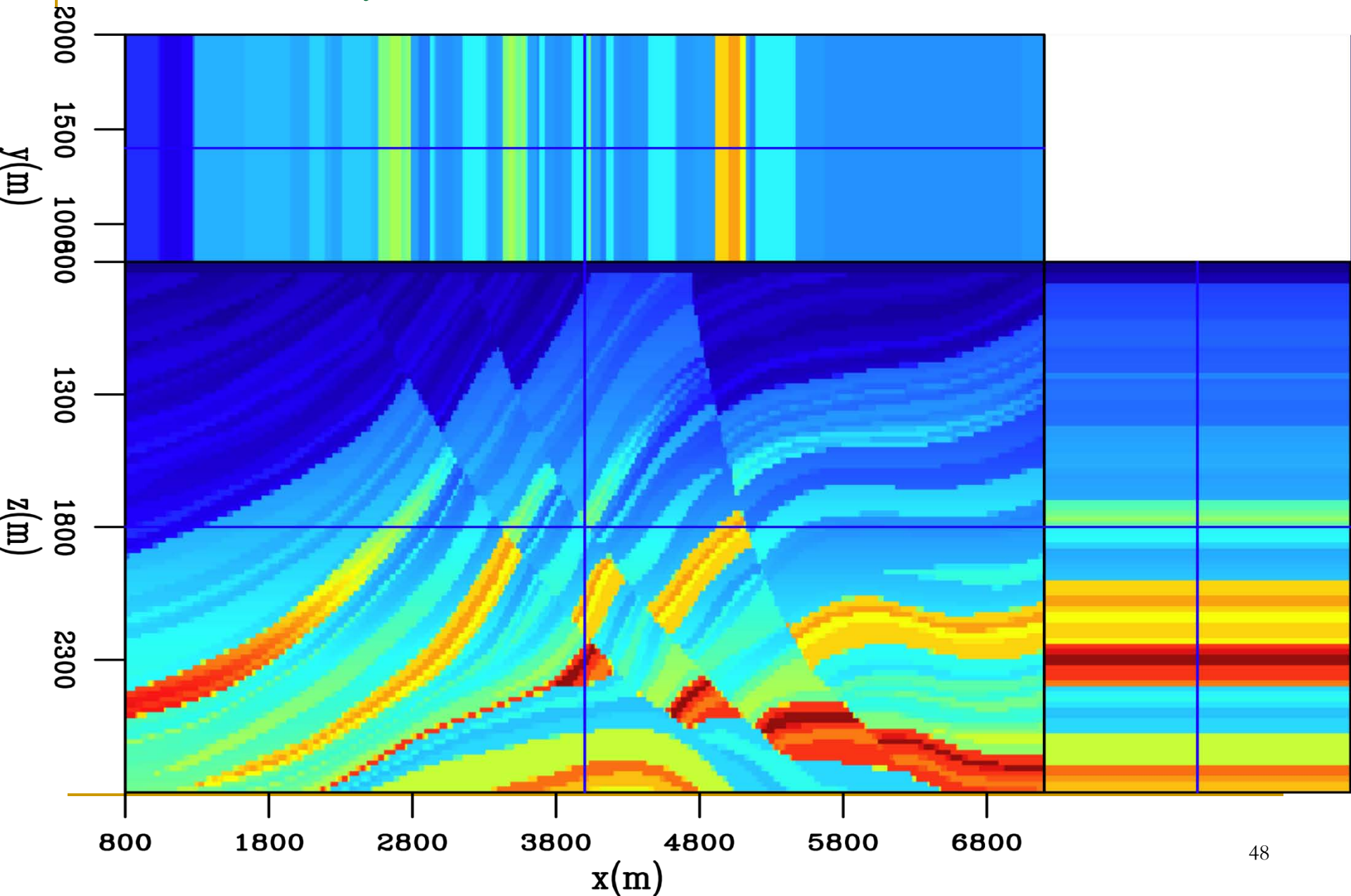
Inversion result with big cubic grain



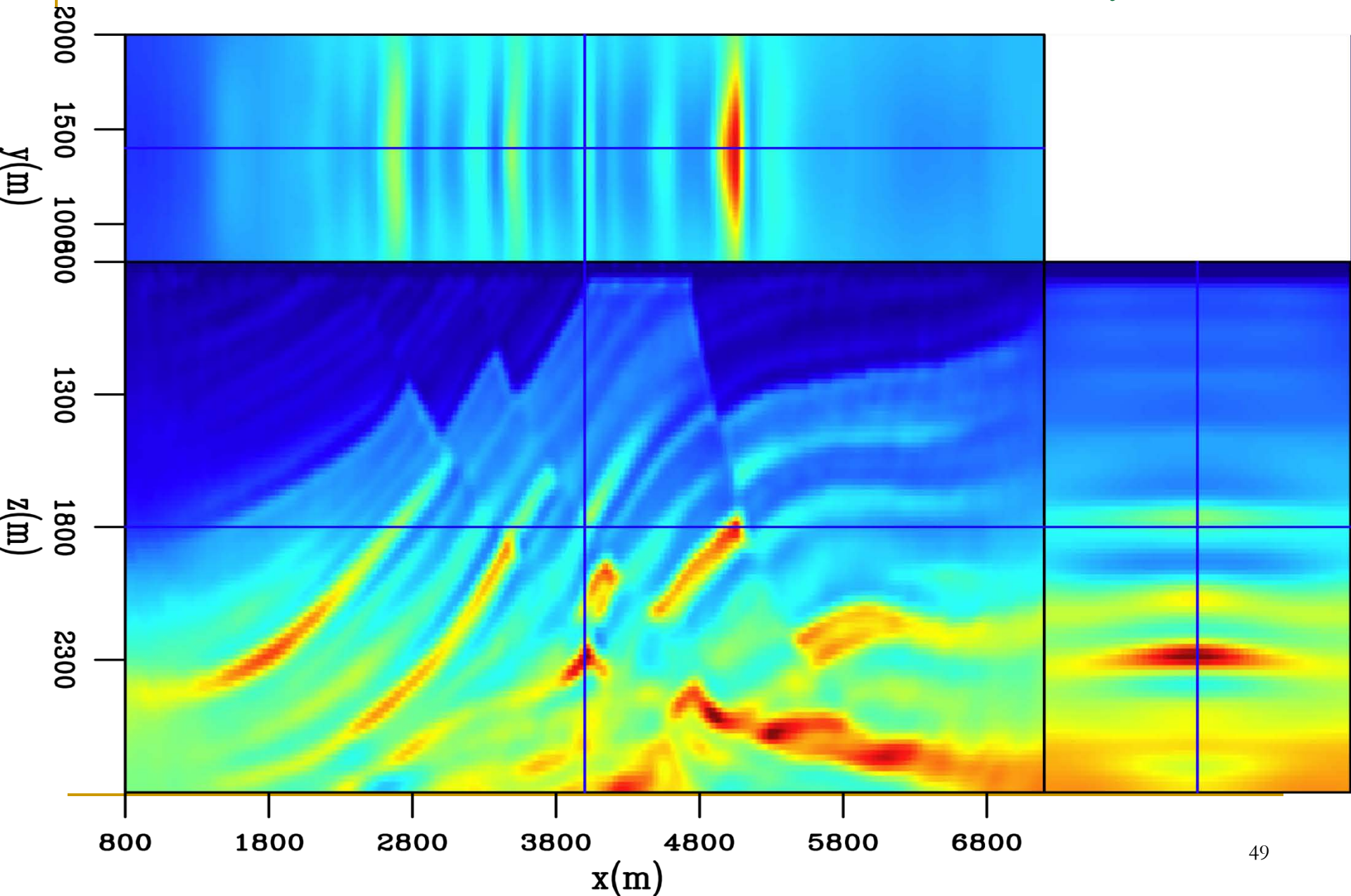
Inversion result with big random grain



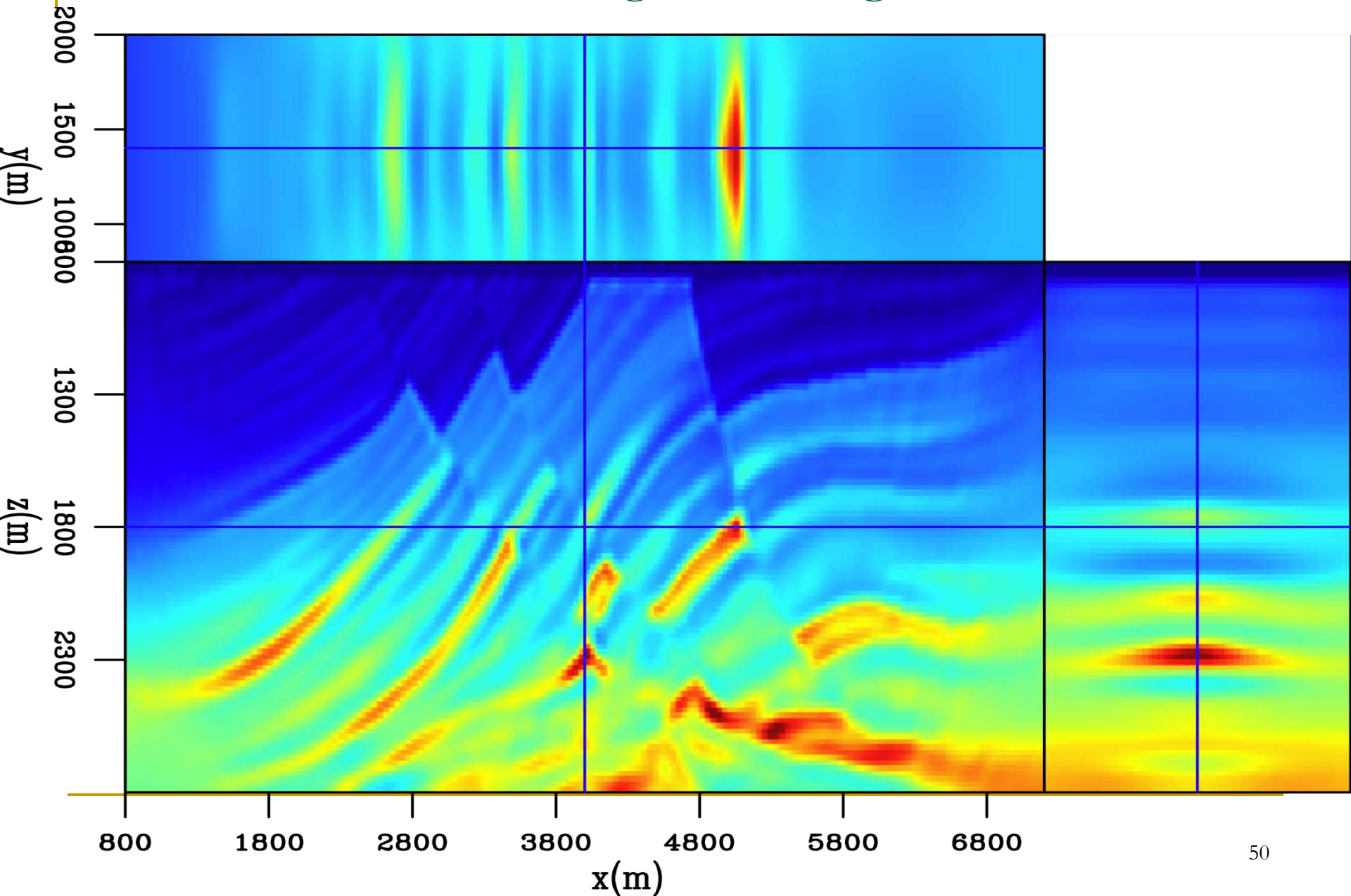
True velocity



Inversion result with continuation of velocity



Inversion result with big random grain



Why constant pad works almost equally well as random boundary in this case?

Why constant pad works almost equally well as random boundary in this case?

Boundary reflection in complex synthetic does not put much artifacts in gradient

Conclusions

- **Randomly-shaped grains in the random boundary effectively scatter low frequency wave without increasing the boundary size**

Conclusions

- Randomly-shaped grains in the random boundary effectively scatter low frequency wave without increasing the boundary size
- Waveform inversion using random boundary works

Conclusions

- Randomly-shaped grains in the random boundary effectively scatter low frequency wave without increasing the boundary size
- Waveform inversion using random boundary works
- Whether using continuation of velocity works in complex synthetic/real data is to be determined

Thank you

Questions & Suggestions

pseudo code for gradient calculation using random boundaries:

{

generate d_{cal} using absorbing boundary

$d_{res} = f(d_{obs}, d_{cal})$

for $t = 0, t_{\max}$

propagate the same source using random boundary

for $t = t_{\max}, 0$

propagate source and residual wavefield, generate gradient

}

pseudo code for gradient calculation using random boundaries:

{

generate d_{cal} using absorbing boundary

$d_{res} = f(d_{obs}, d_{cal})$

for $t = 0, t_{\max}$

propagate the same source using random boundary

for $t = t_{\max}, 0$

propagate source and residual wavefield, generate gradient

}

4 propagations instead of 2, but avoid saving wavefield.

Can be further optimized by simultaneous encoding of residual/source wavefield