FWI with different boundaries

Xukai Shen*, Robert Clapp

SEP147 p159

2012 SEP Sponsor Meeting

May 23rd 2012

Outline

1. Introduction
2. Motivation
3. Synthetic Example
4. Conclusion

Outline

1. Introduction
2. Motivation
3. Synthetic Example
4. Conclusion

Full Waveform inversion (FWI):

Iterative velocity estimation by matching modeled data to recorded data

Full Waveform inversion (FWI):

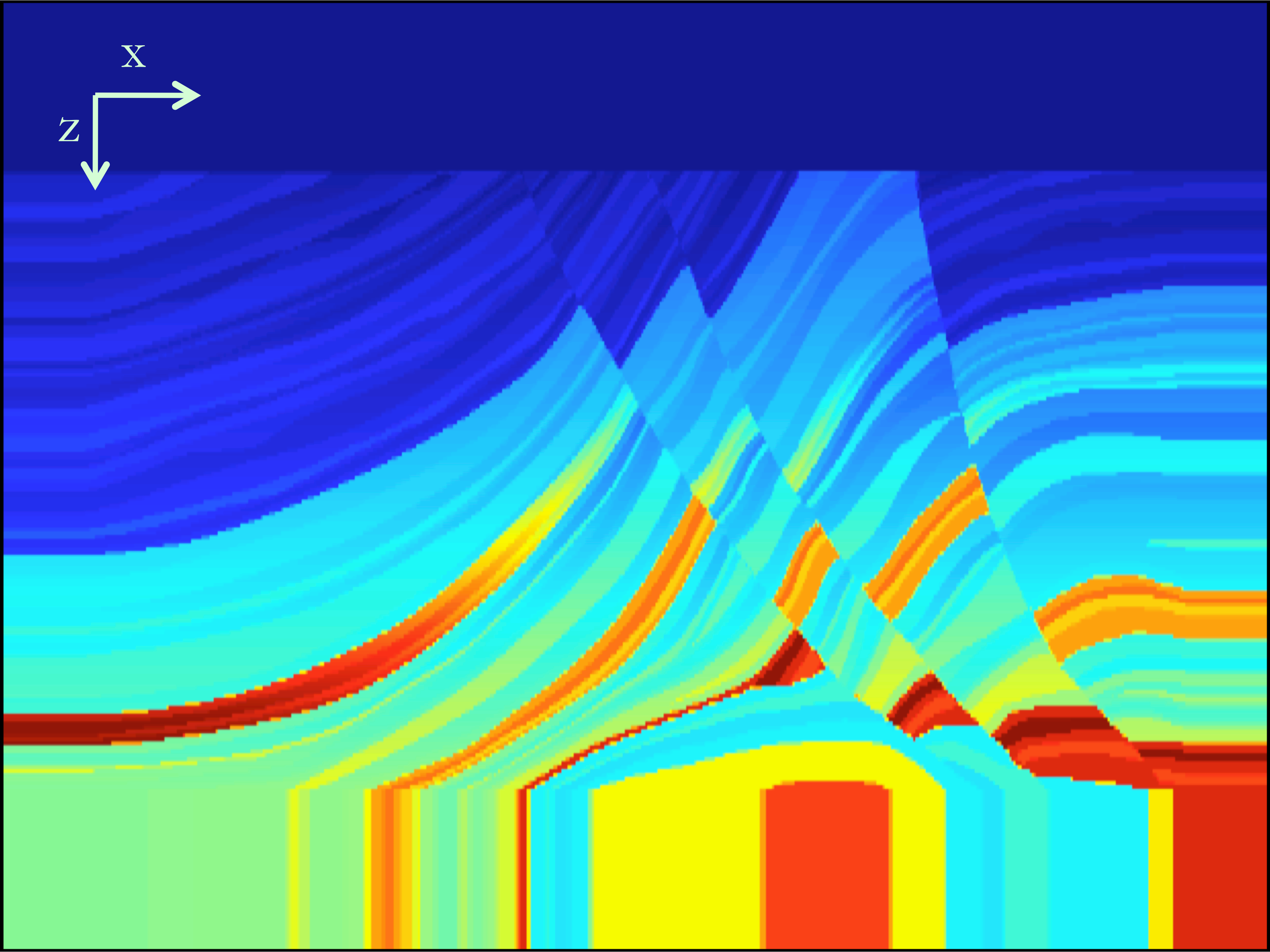
Iterative velocity estimation by matching modeled data to recorded data

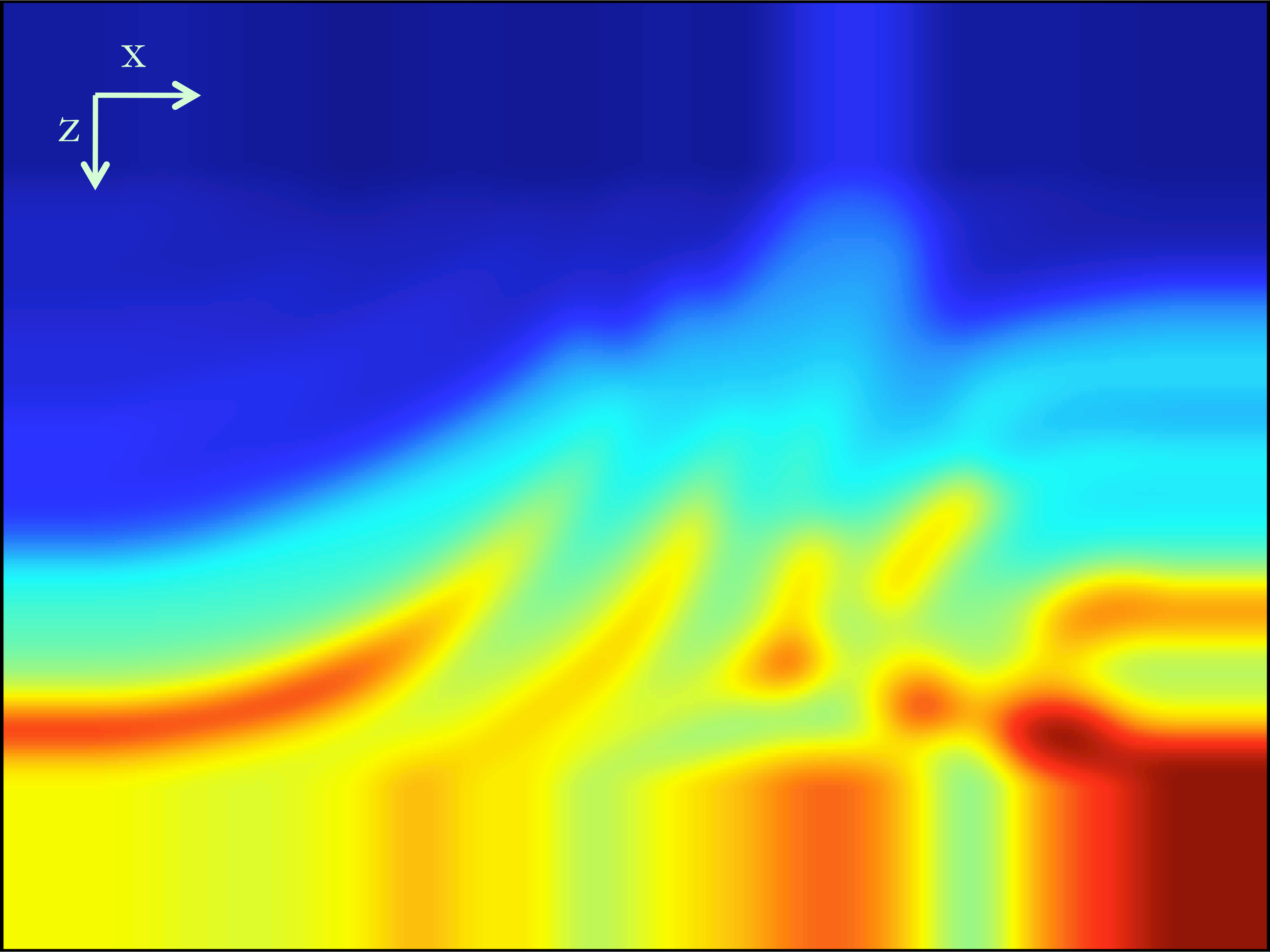
Each iteration (gradient based methods)

= Gradient calculation

+

Step length calculation

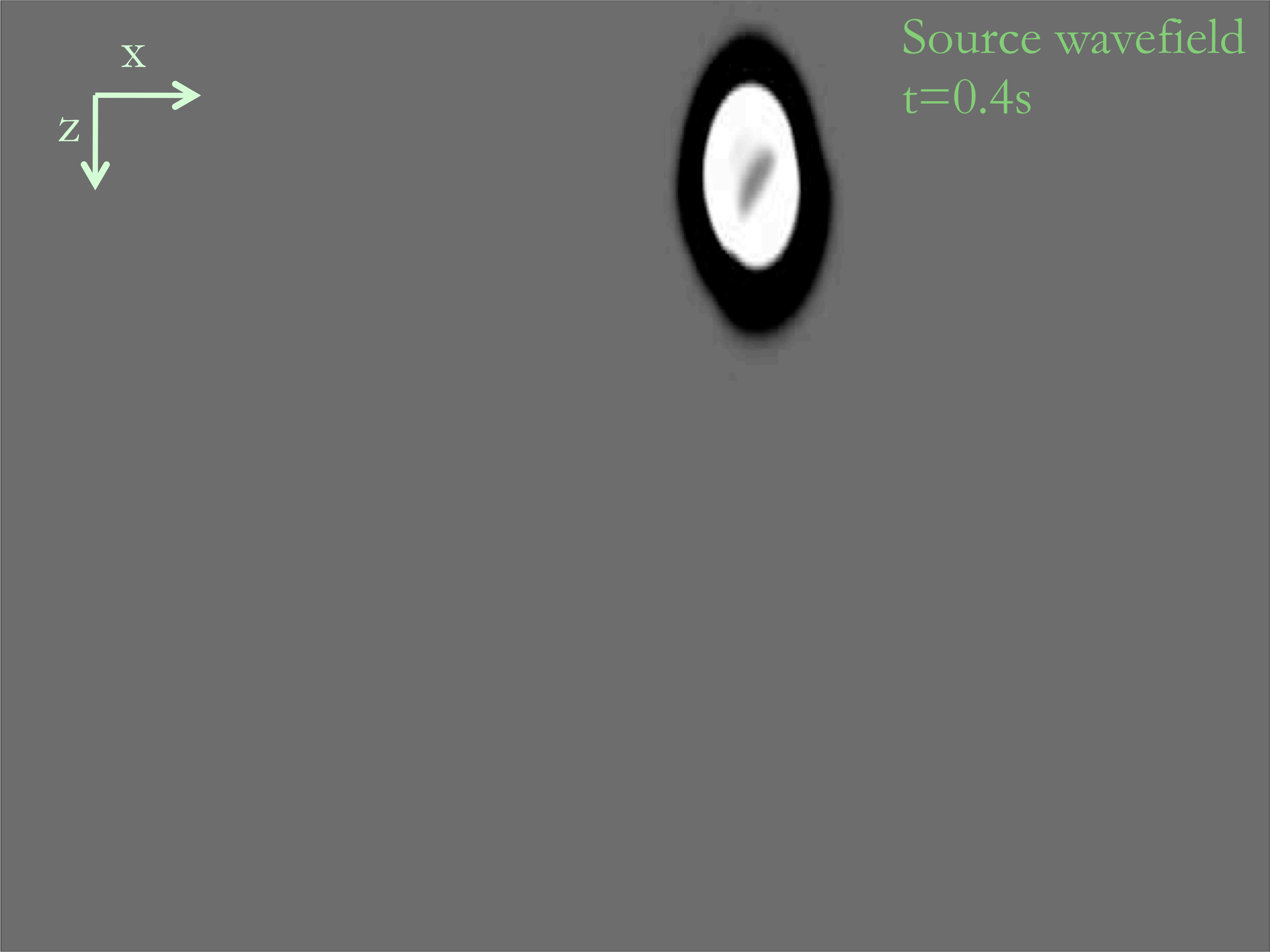

x
z


x
z

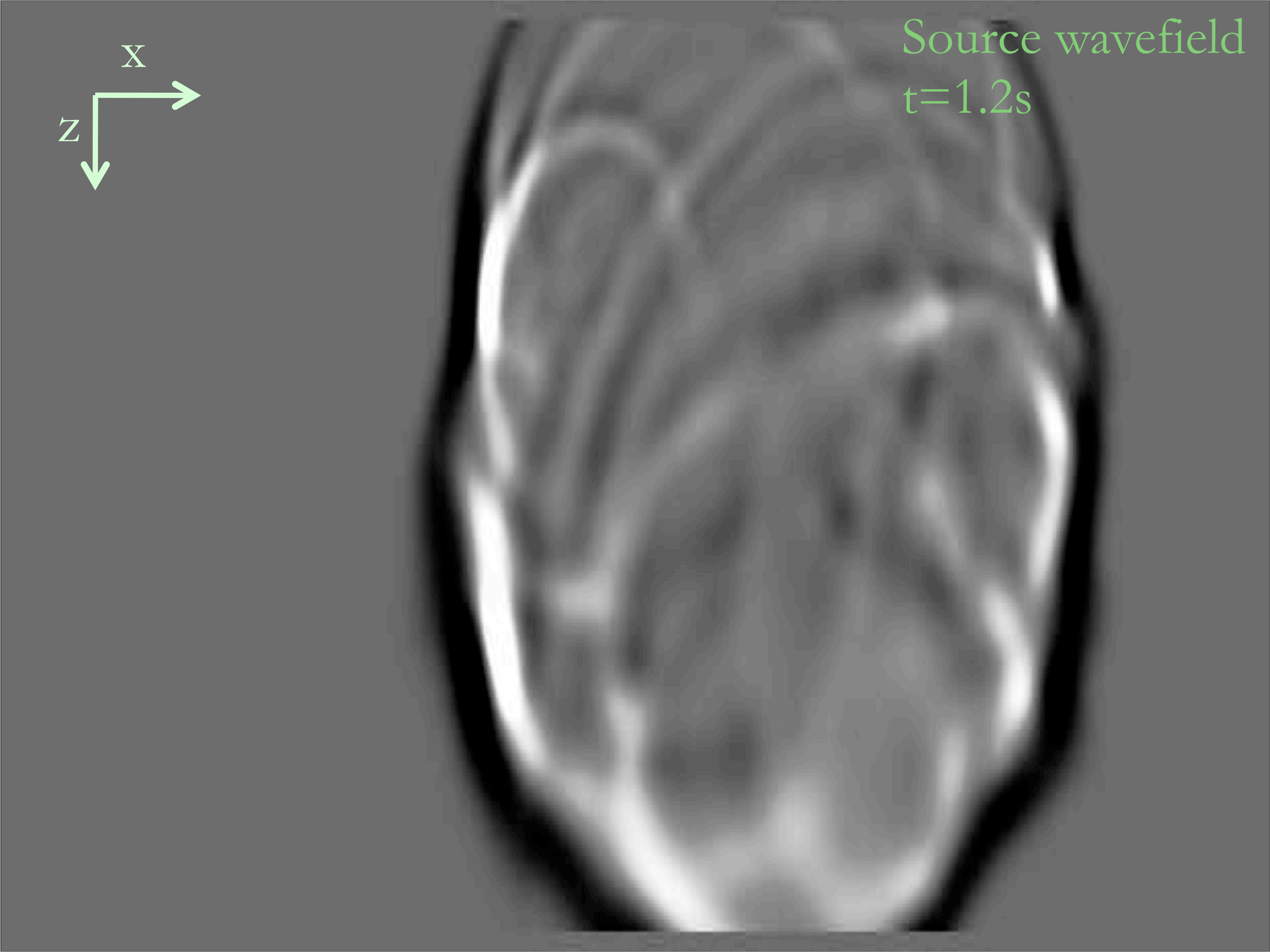
FWI gradient calculation:

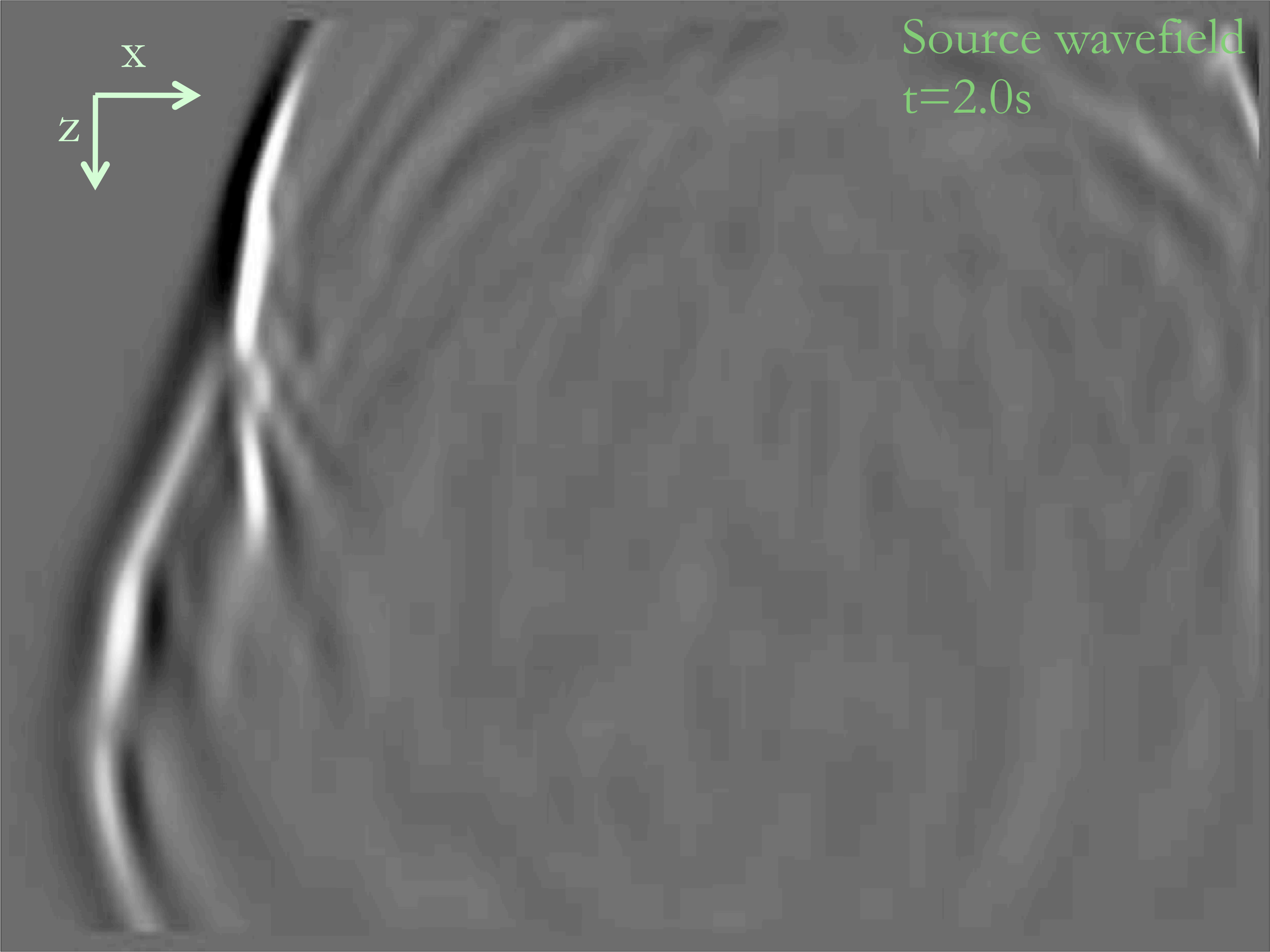
Source propagation , record source wavefield $\mathbf{U}_s$

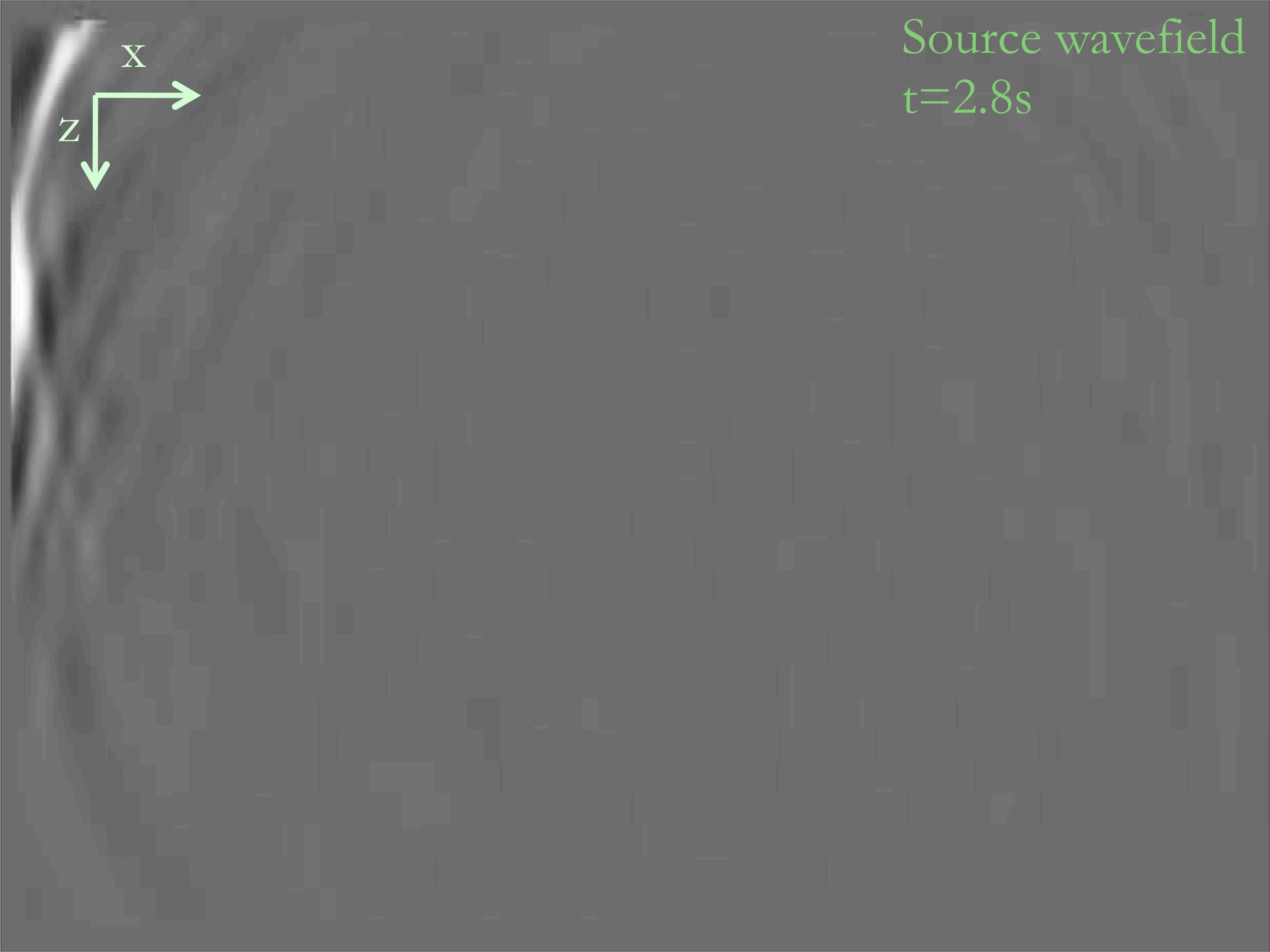
t_0 → $t_{\max}$

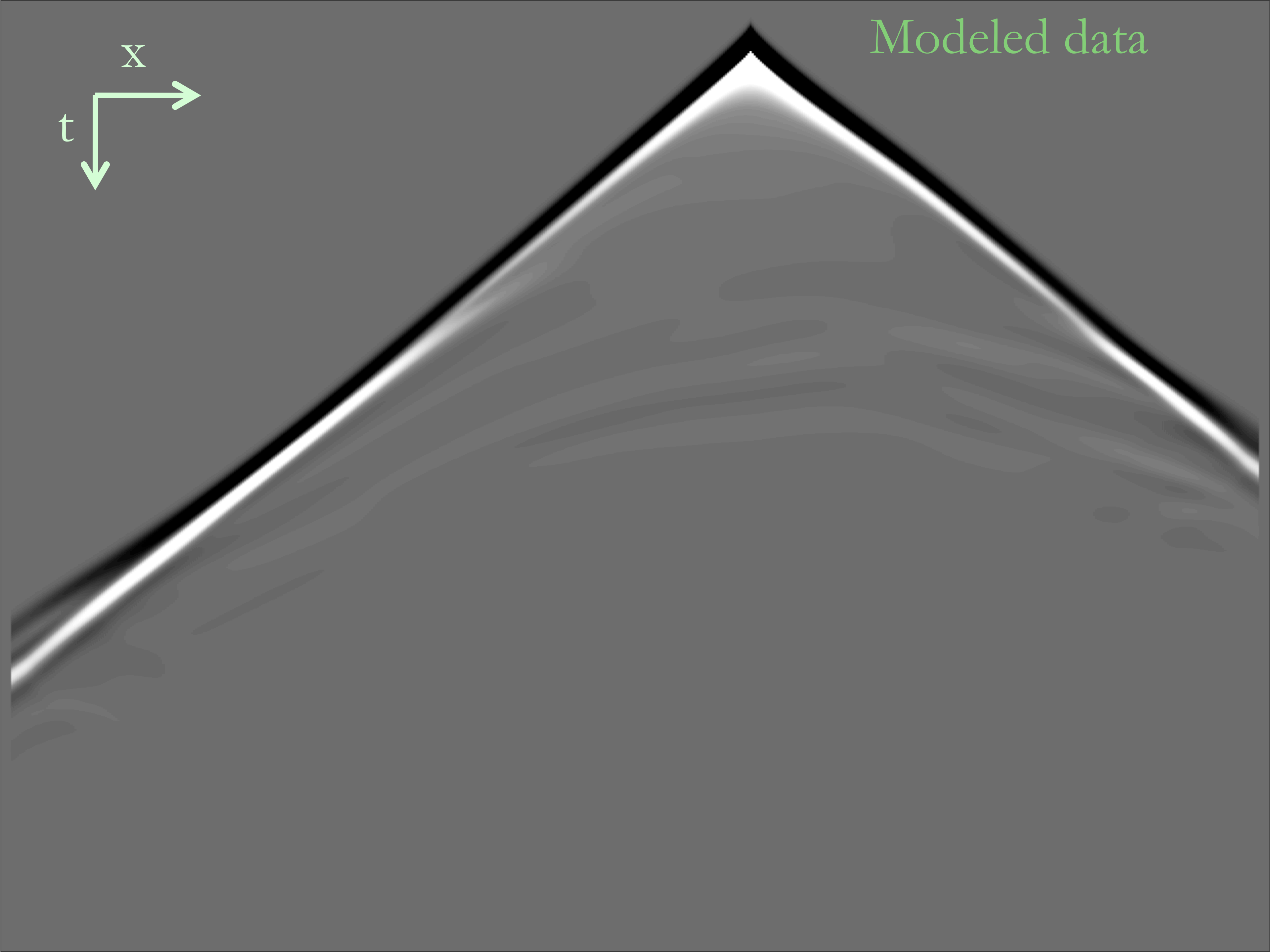


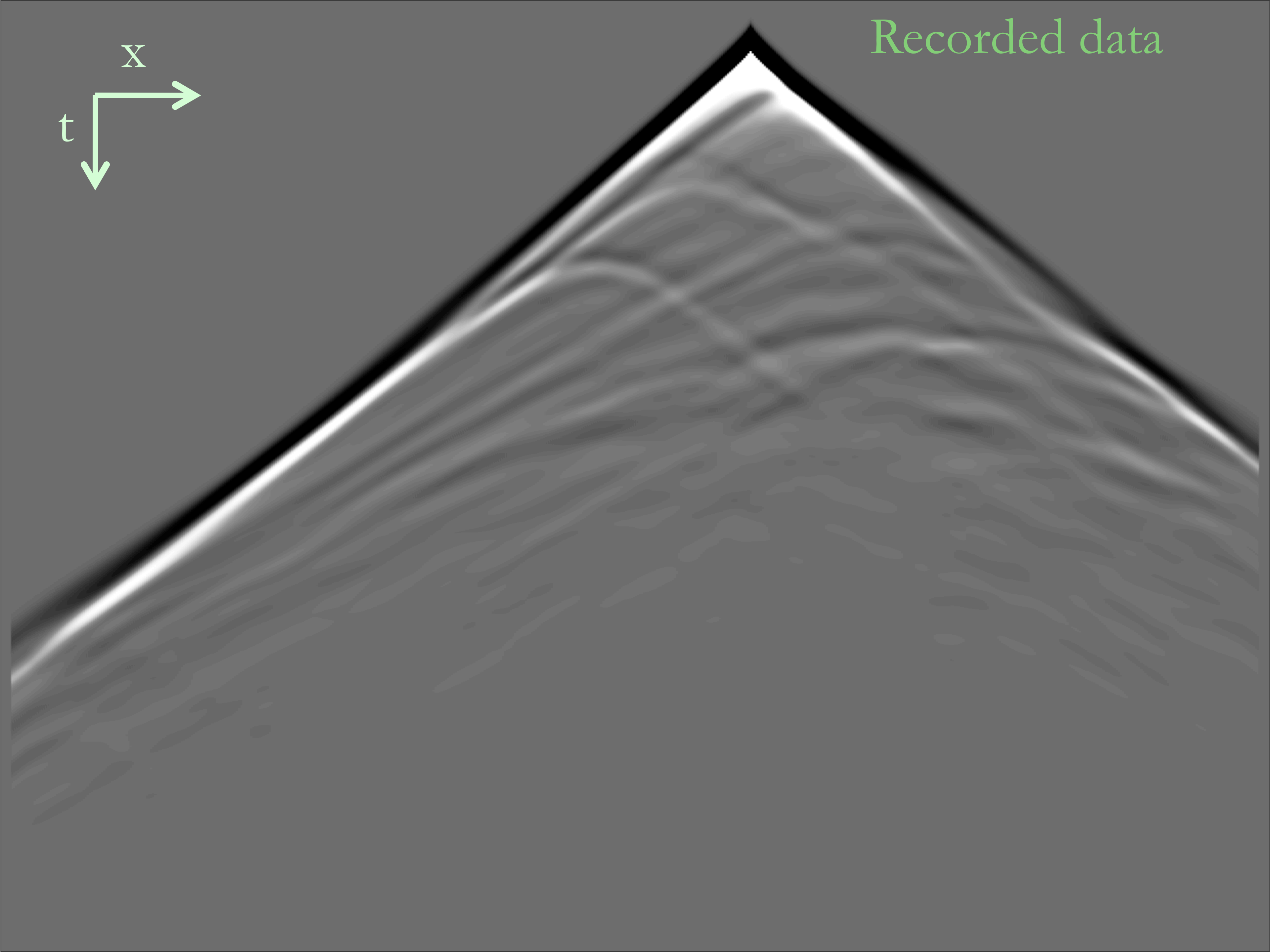
Source wavefield
t=0.4s
x
z

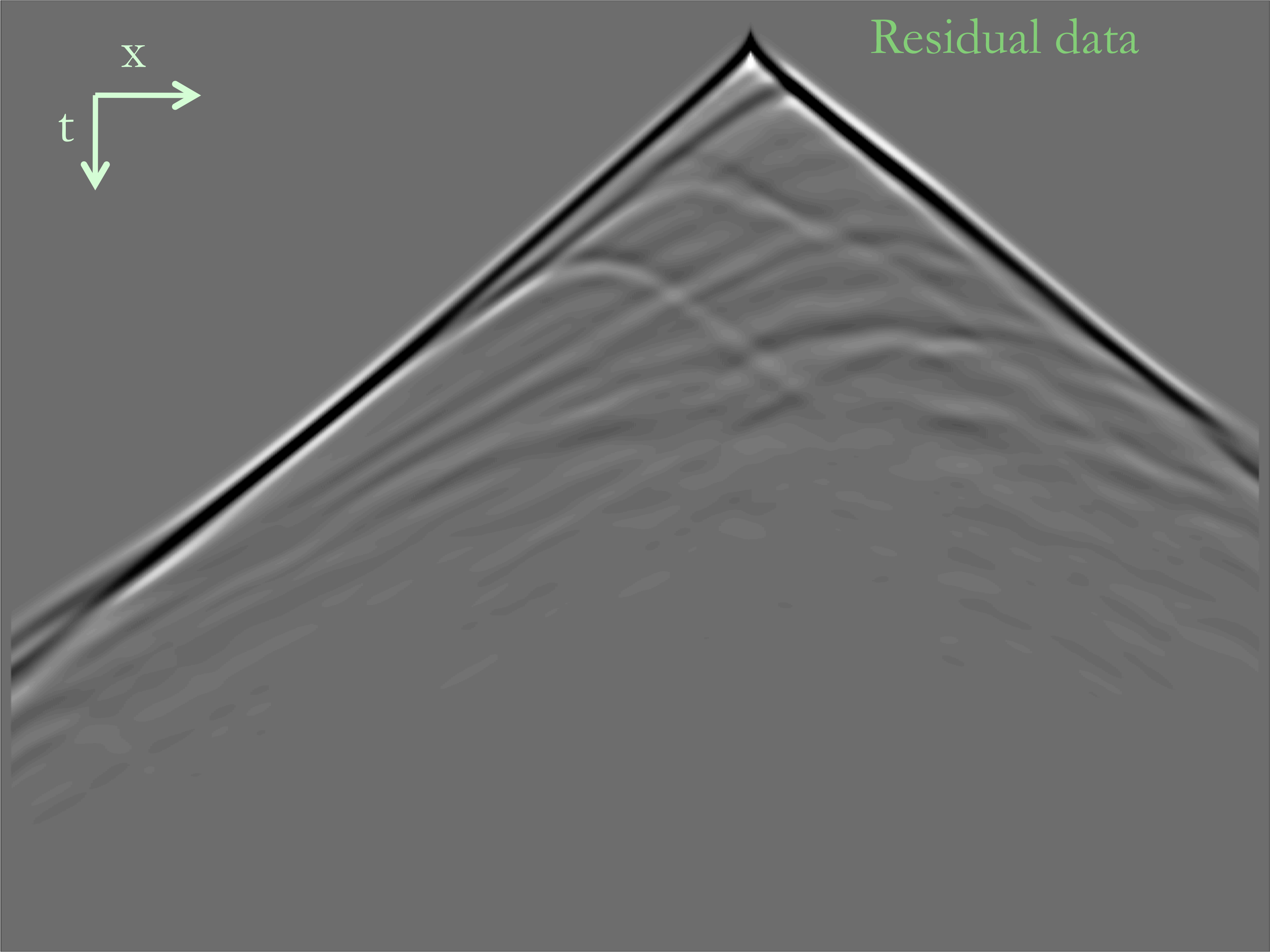

x
z
Source wavefield
t=1.2s


x
z
Source wavefield
t=2.0s


Source wavefield
t=2.8s
x
z


Modeled data
x
t


Recorded data
x
t


Residual data
x
t

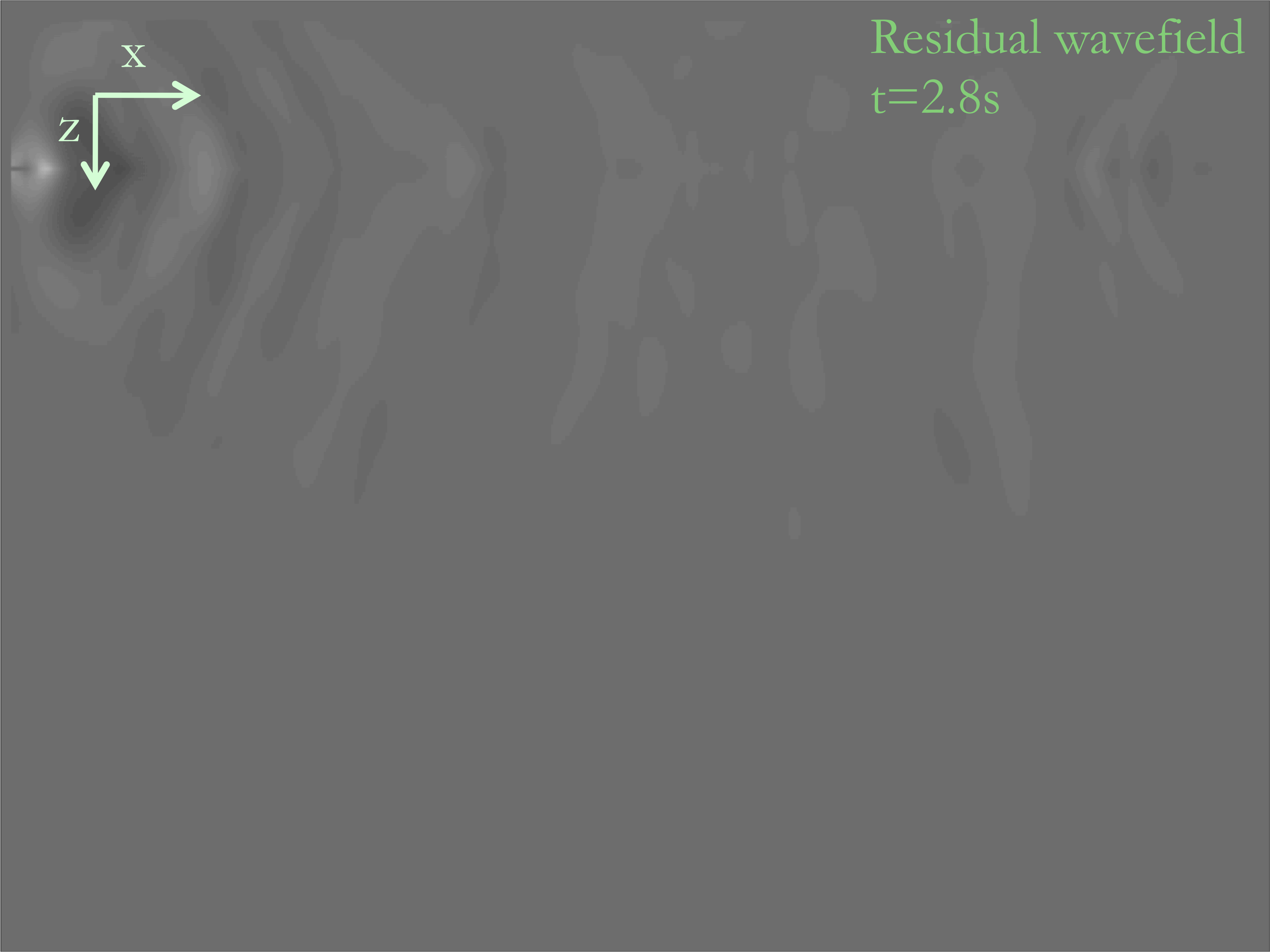
FWI gradient calculation:

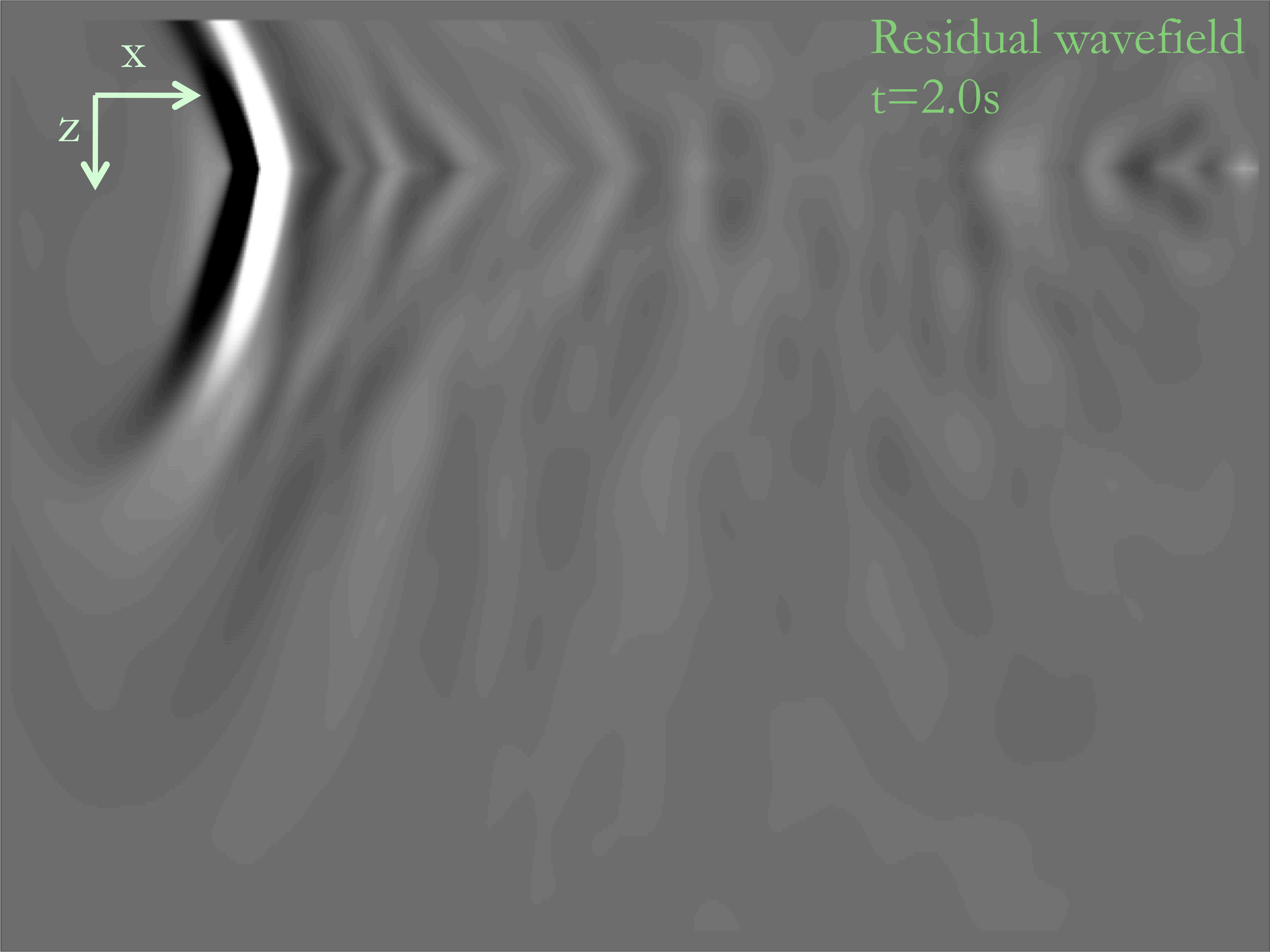
Source propagation , record source wavefield $\mathbf{U}_s$

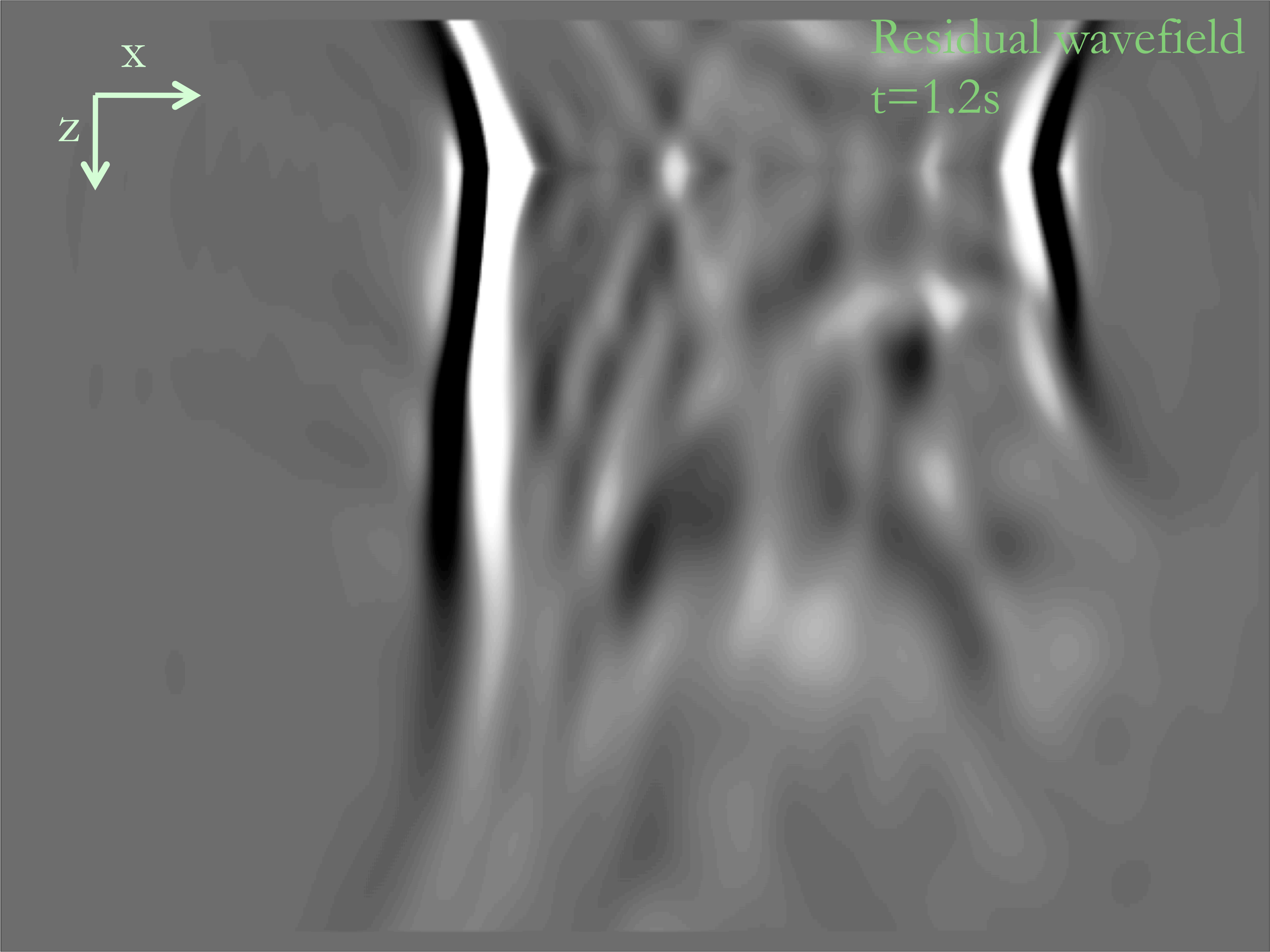
t_0 → $t_{\max}$

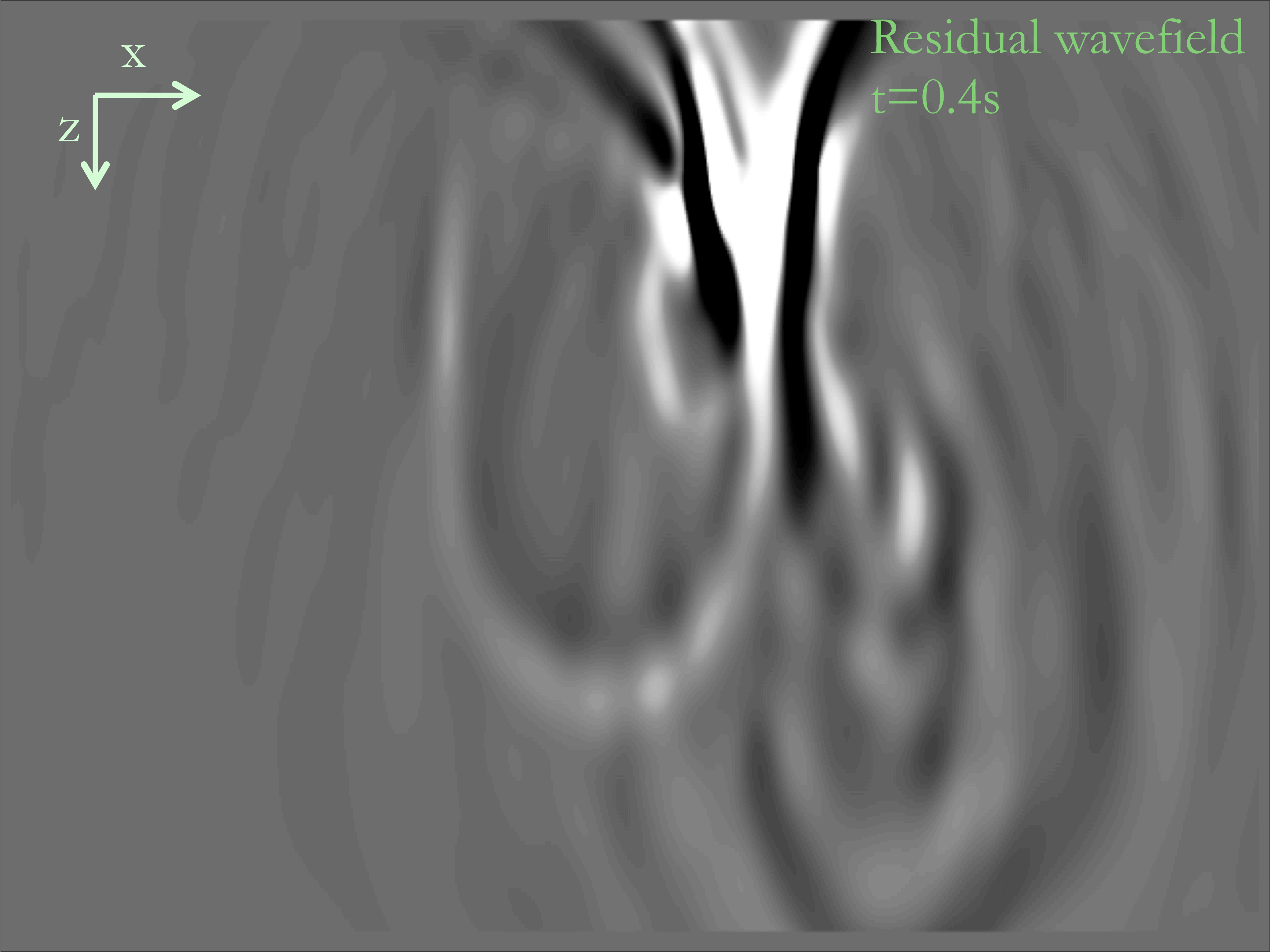
t_0 ← $t_{\max}$

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


Residual wavefield
t=2.8s
x
z


Residual wavefield
t=2.0s
x
z


x
z
Residual wavefield
t=1.2s



x
z
Residual wavefield
t=0.4s


One shot gradient
x
z

FWI step length calculation:

$$\mathbf{m_1} = \mathbf{m_0} + \alpha \bullet \mathbf{g}$$

Computational summary for 1 iteration:

4 wave propagations (2 for gradient
2 for step length)

+

1 wave field saving (gradient calculation)

Computational summary for 1 iteration:

4 wave propagations (2 for gradient
2 for step length)

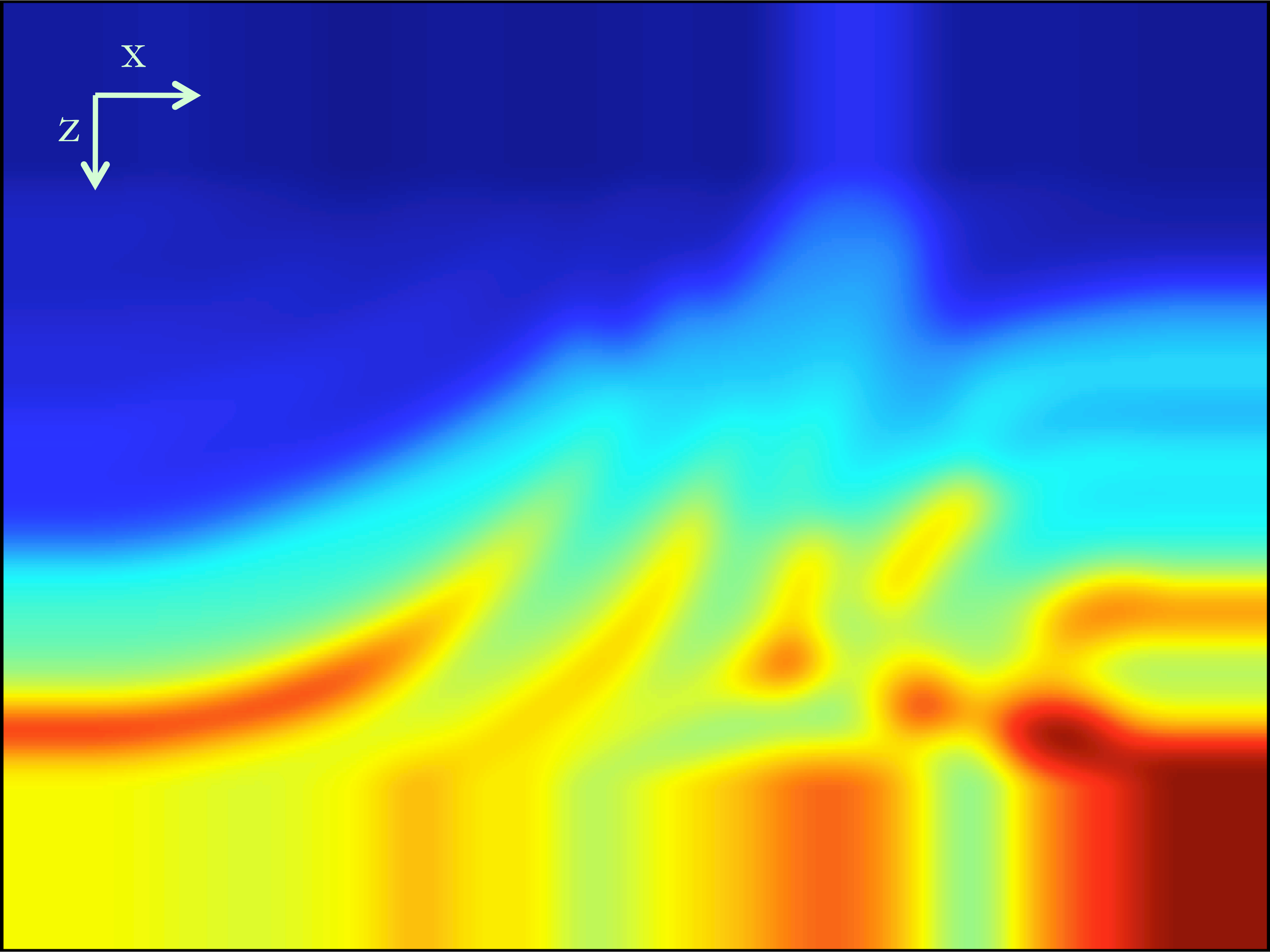
+

1 wave field saving (gradient calculation)

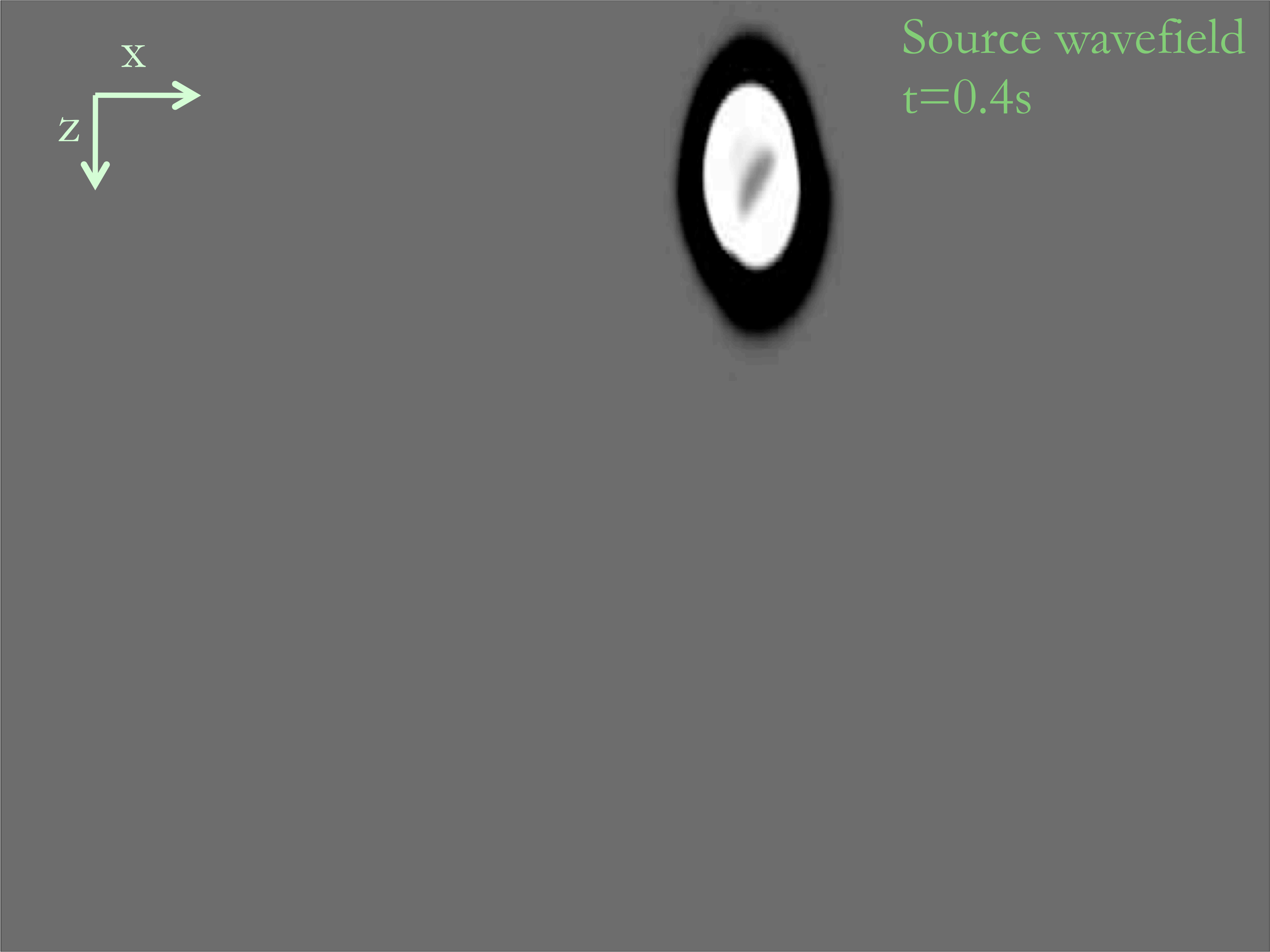
↓

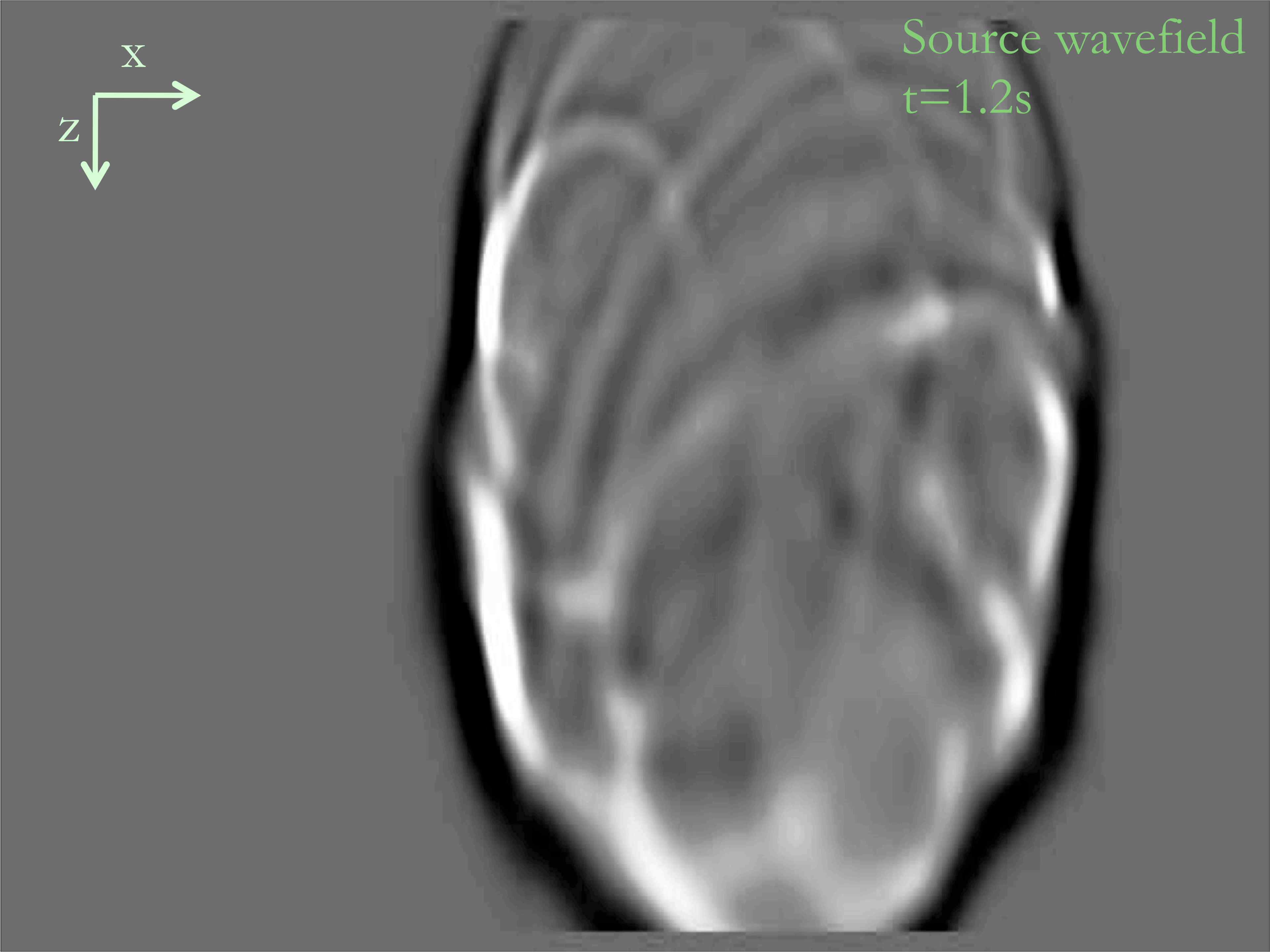
nx*ny*nz*nt=500*500*500*500 ~ 250G

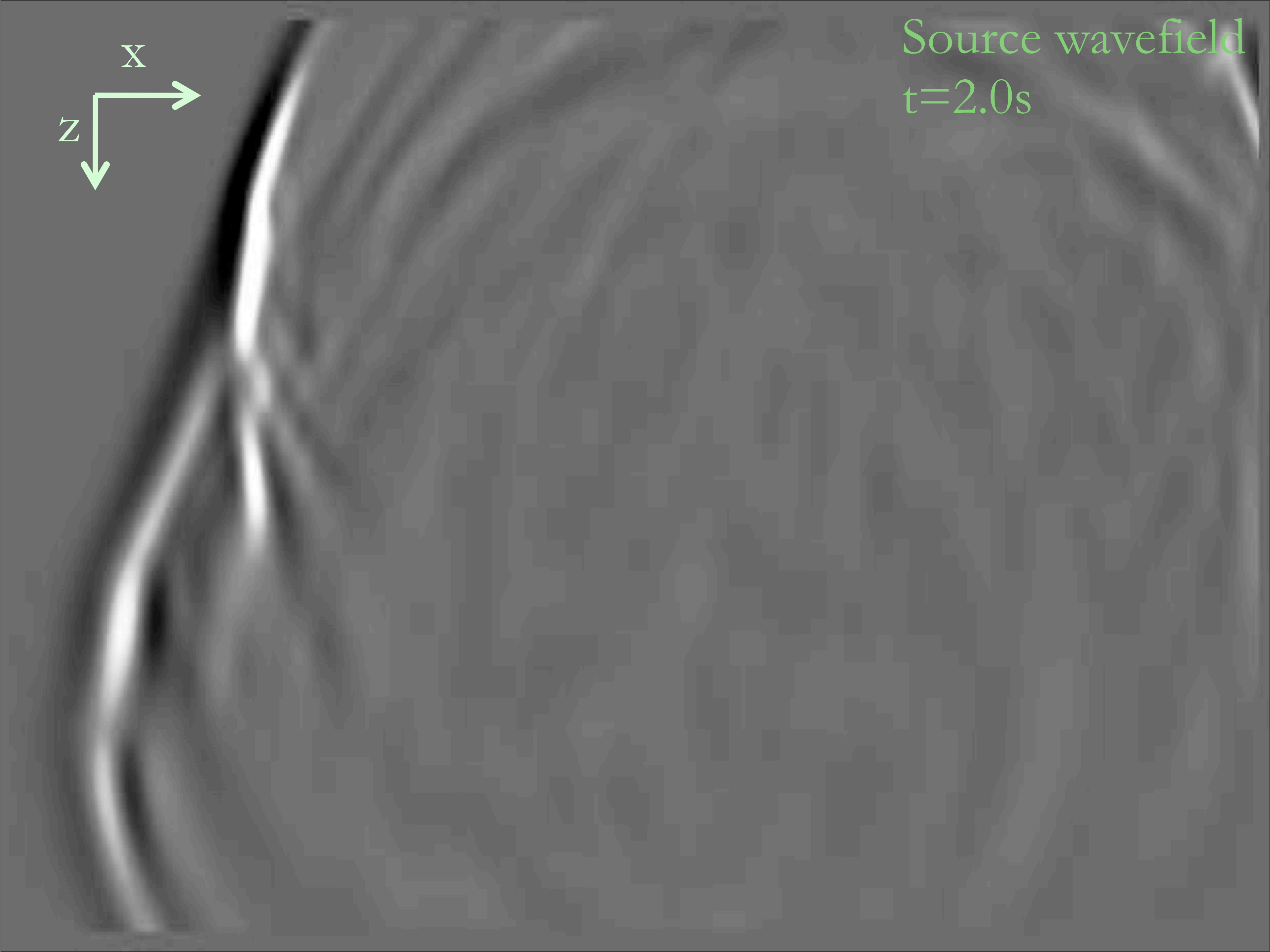
Use random boundary (Clapp, 2010; Shen and Clapp, 2011) to reduce memory requirement!!!

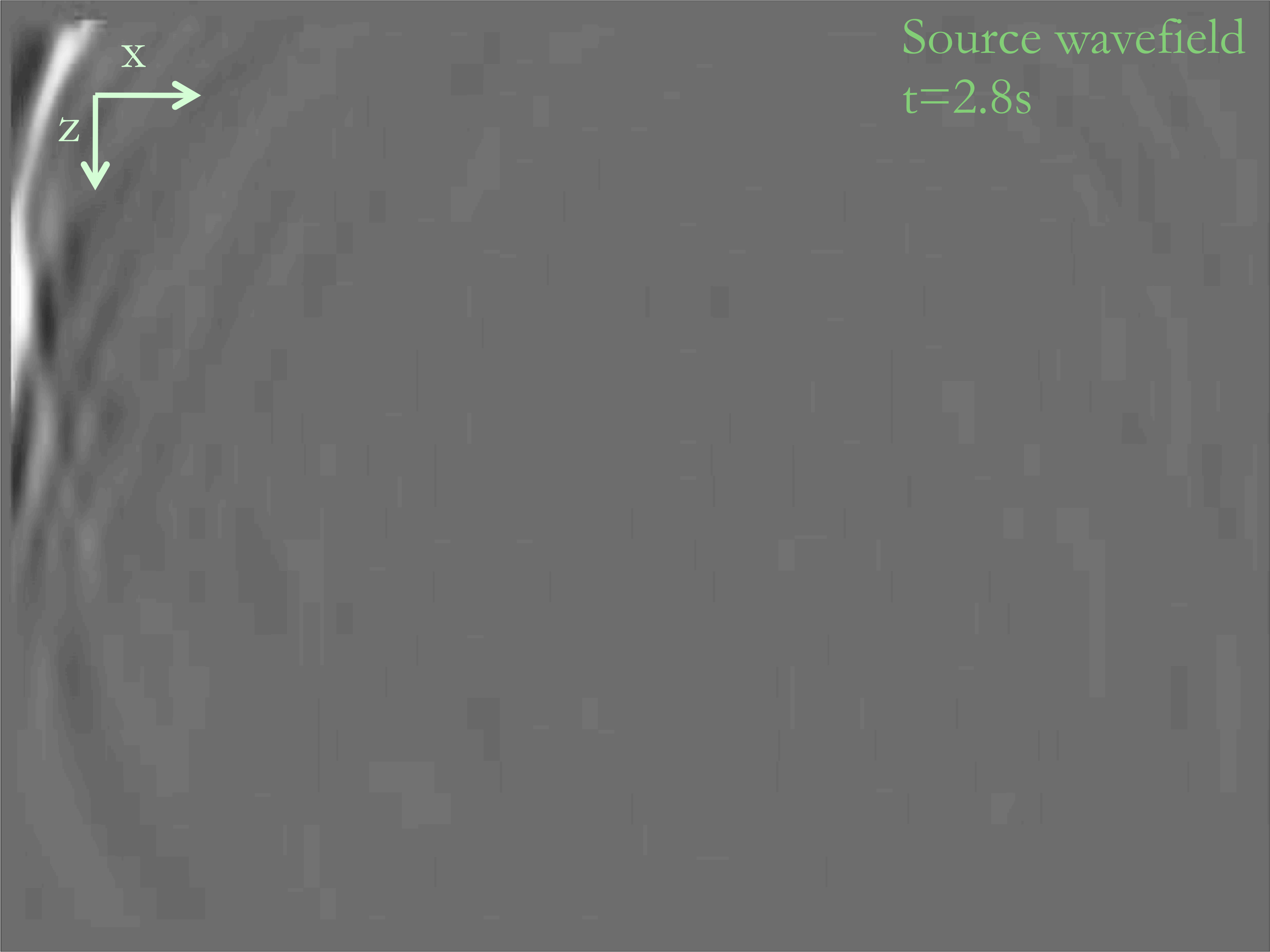


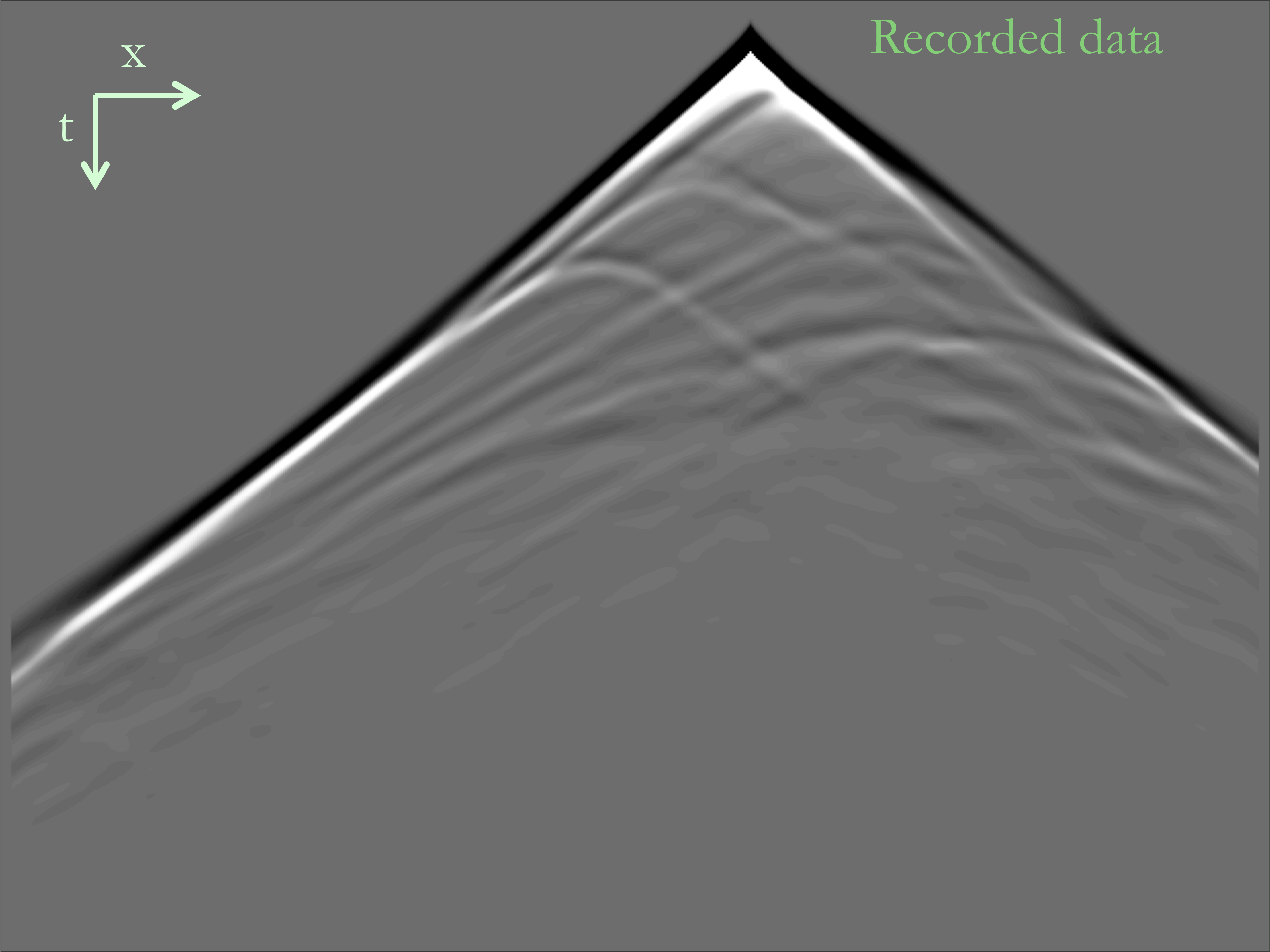
X
Z

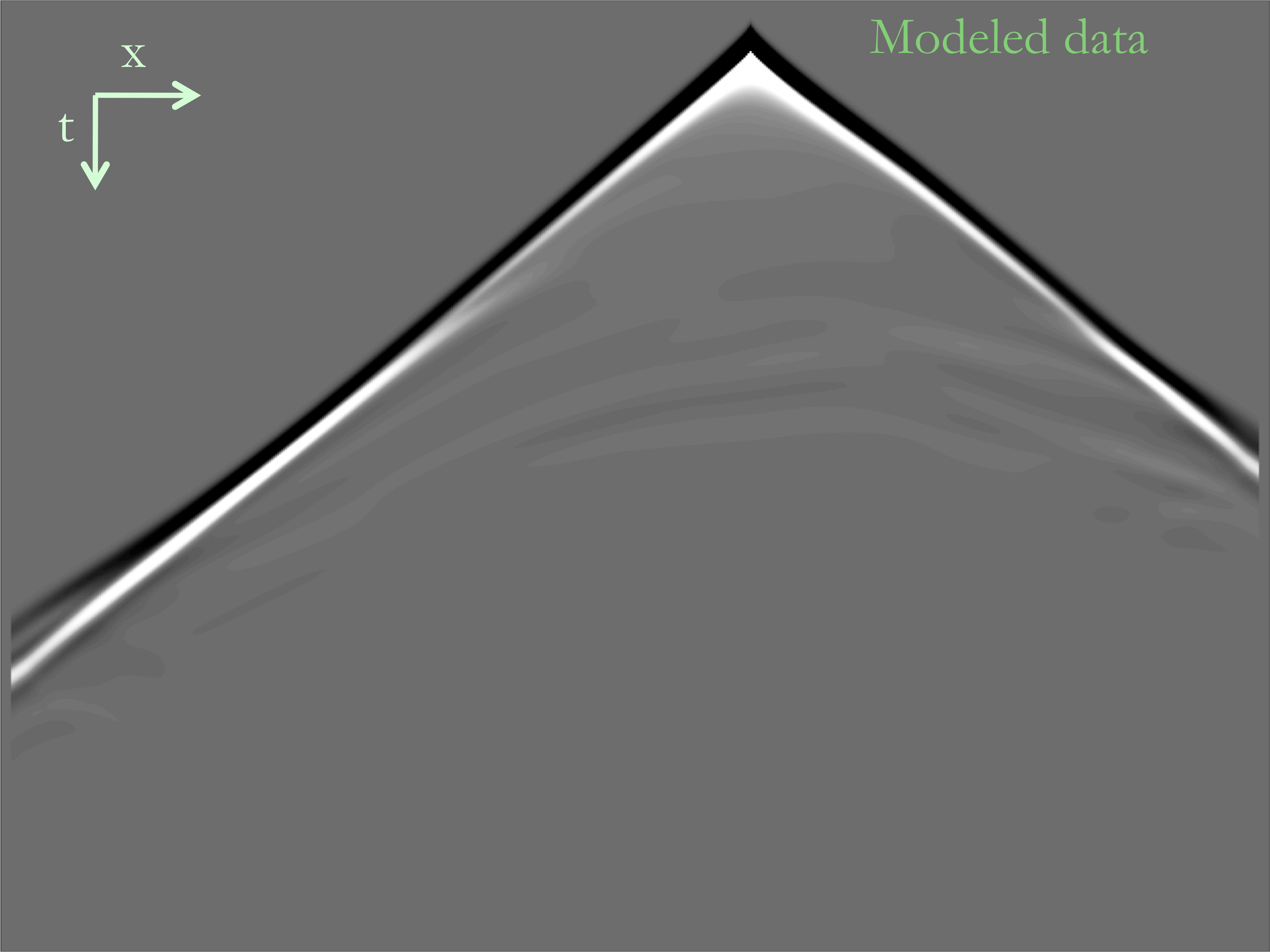

Source wavefield
t=0.4s
x
z

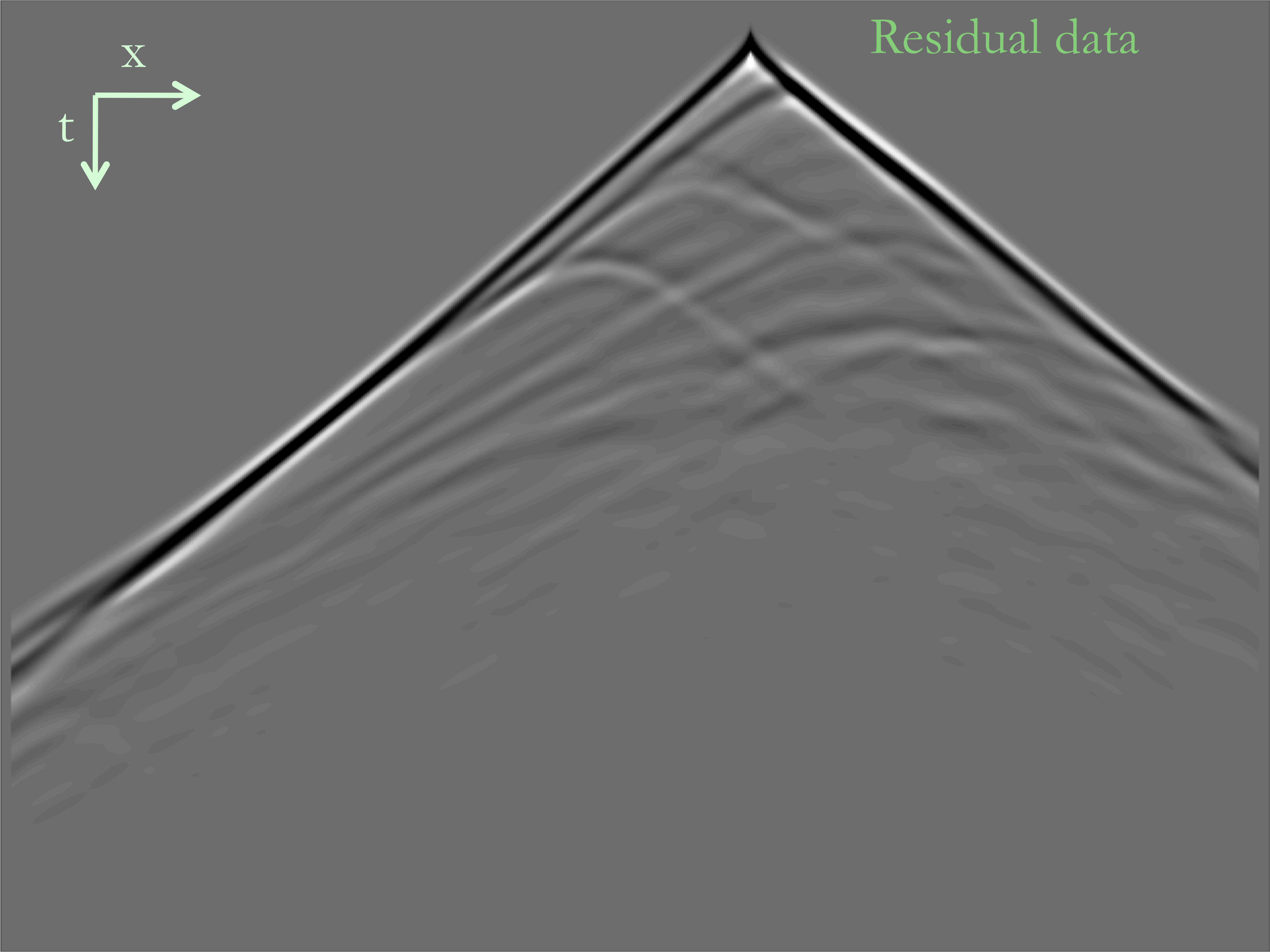

x
z
Source wavefield
t=1.2s

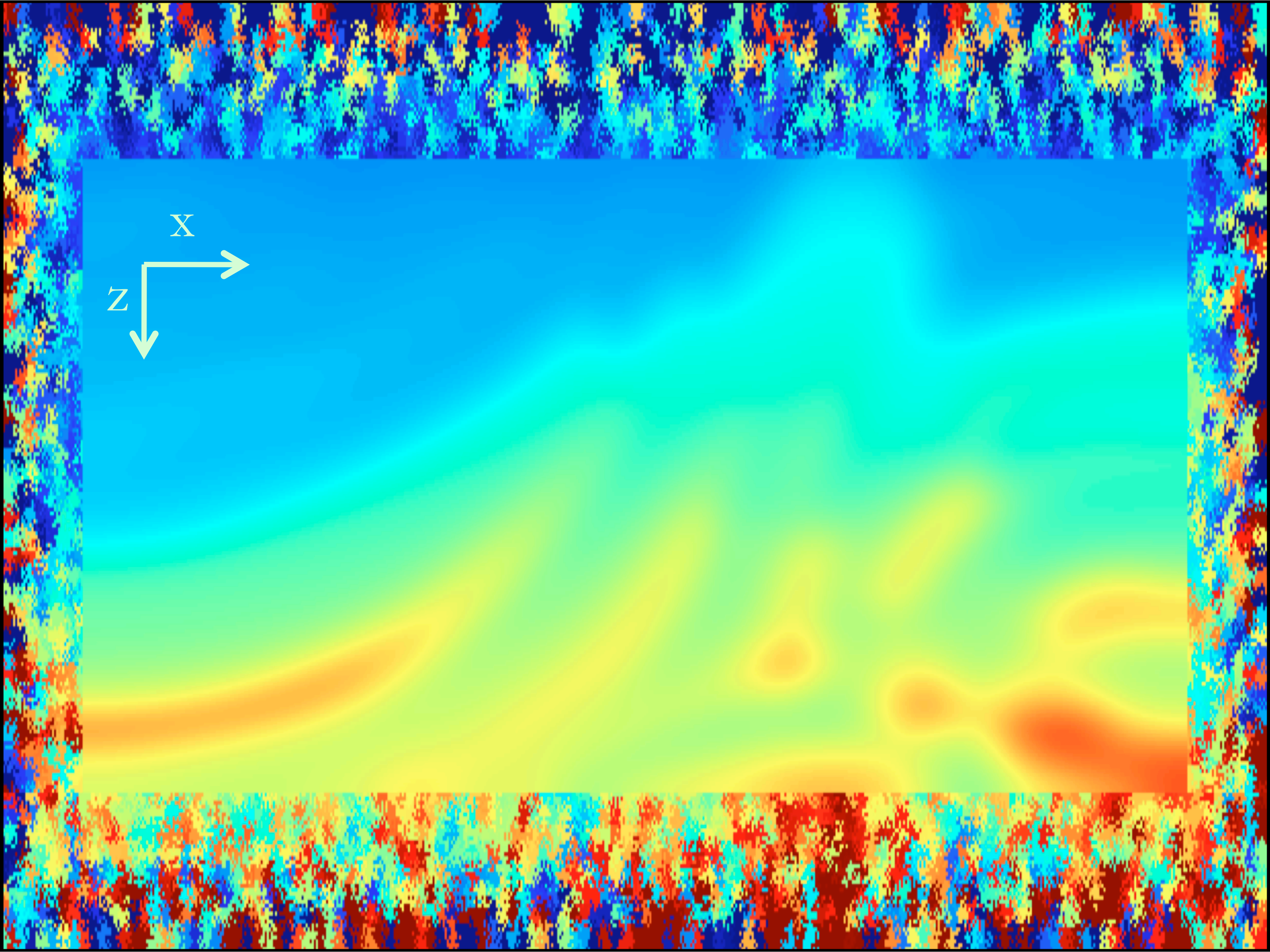

x
z
Source wavefield
t=2.0s

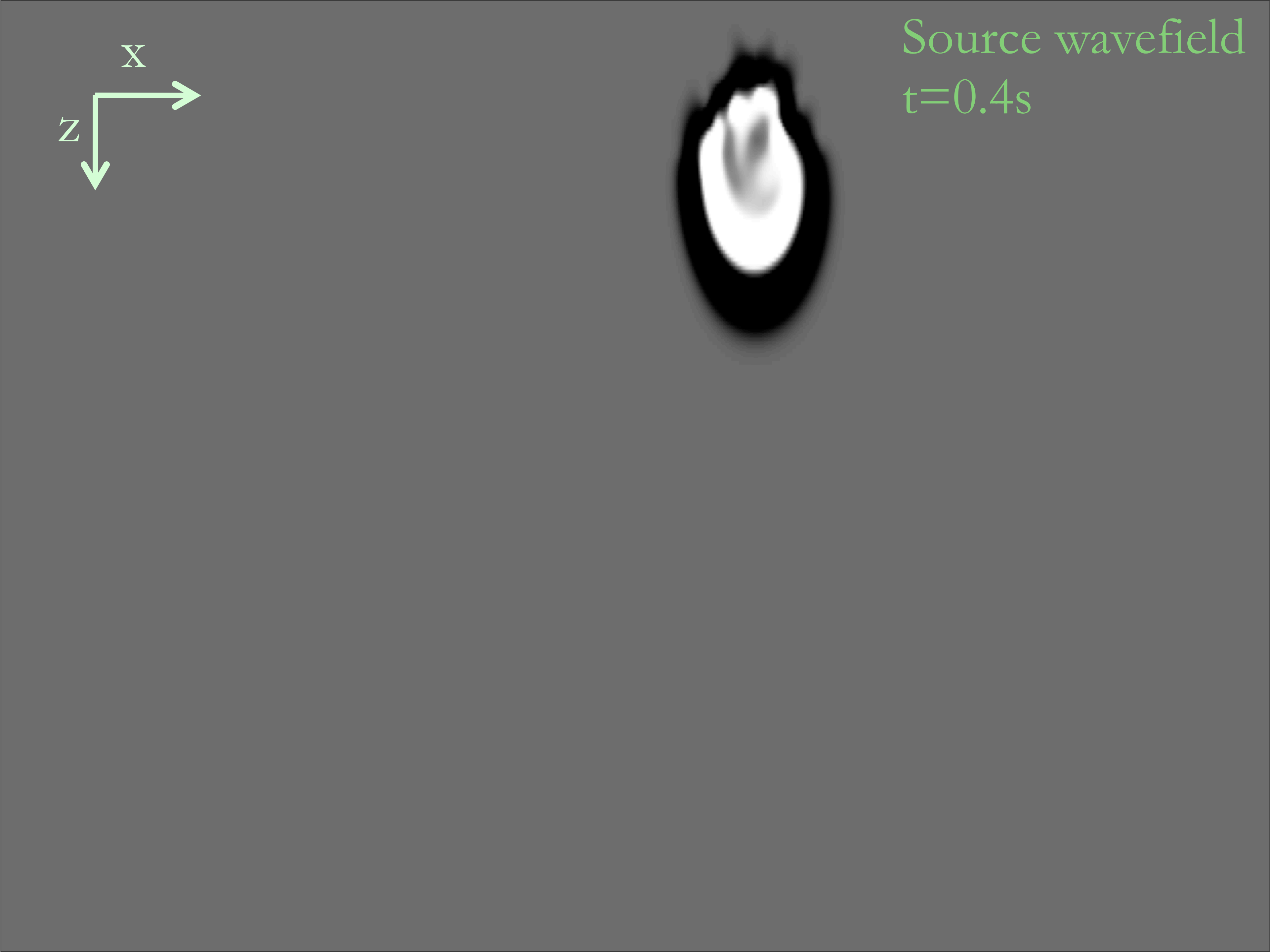

Source wavefield
t=2.8s
x
z


Recorded data
x
t

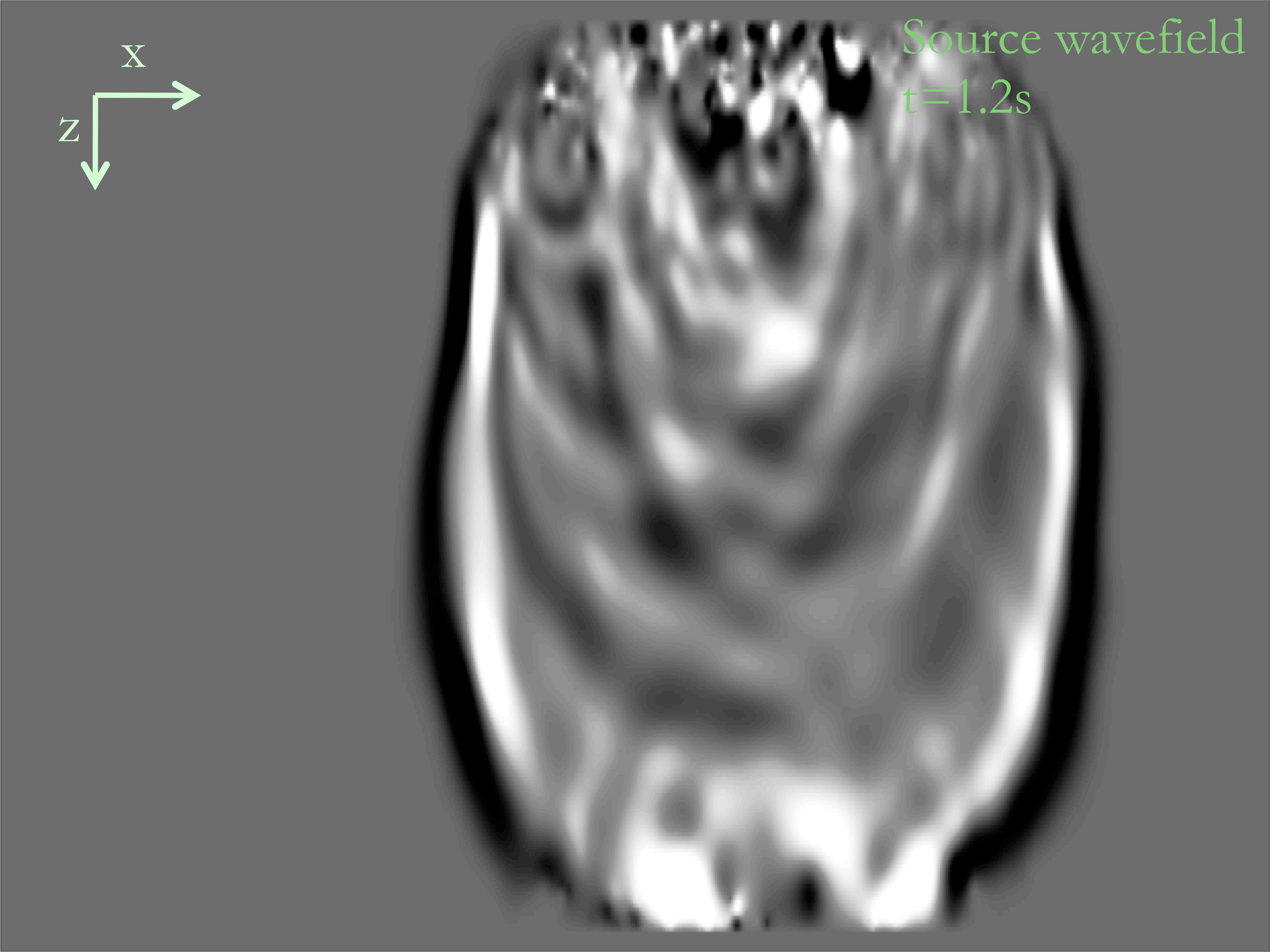

Modeled data
x
t

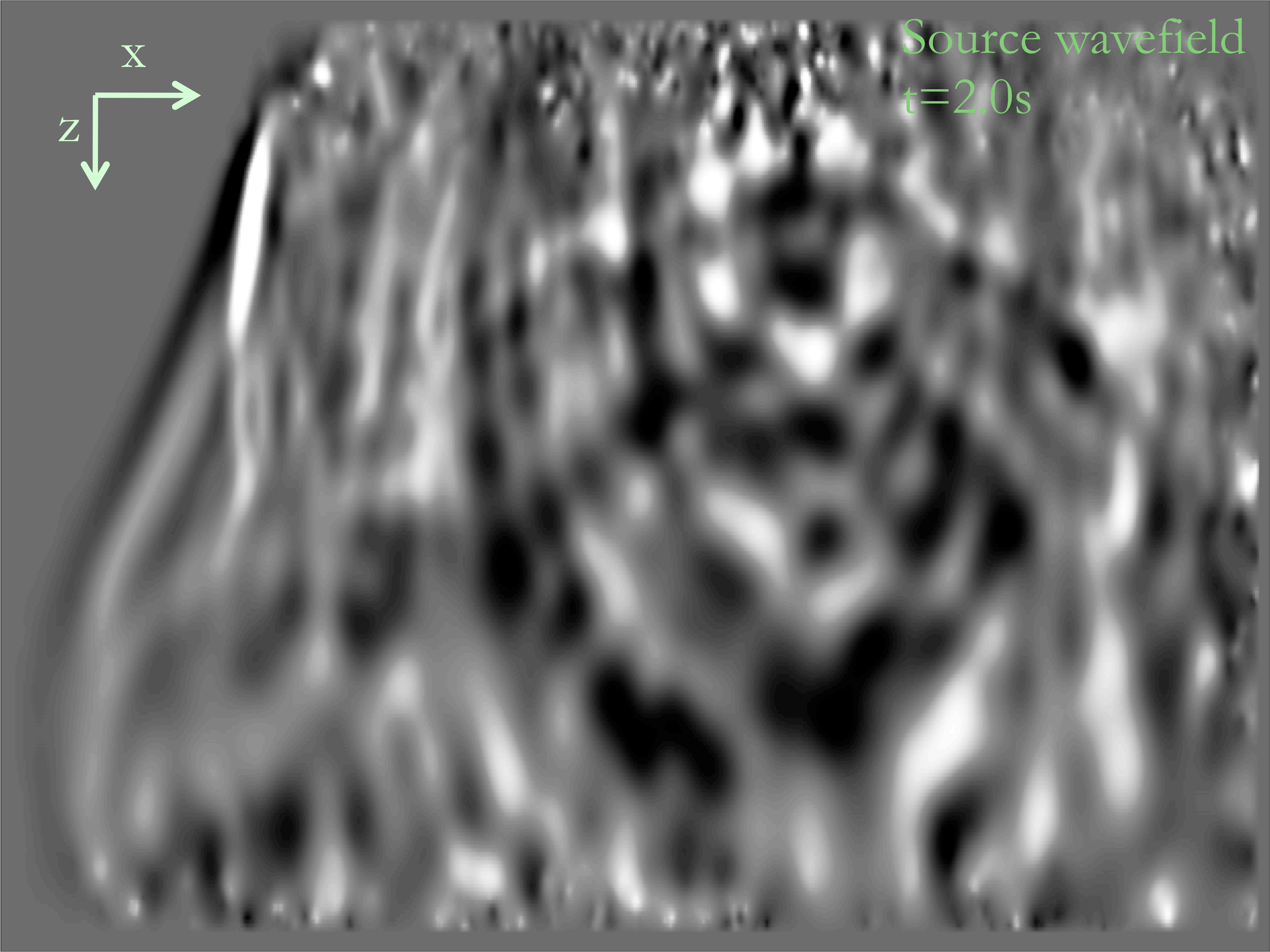

Residual data
x
t

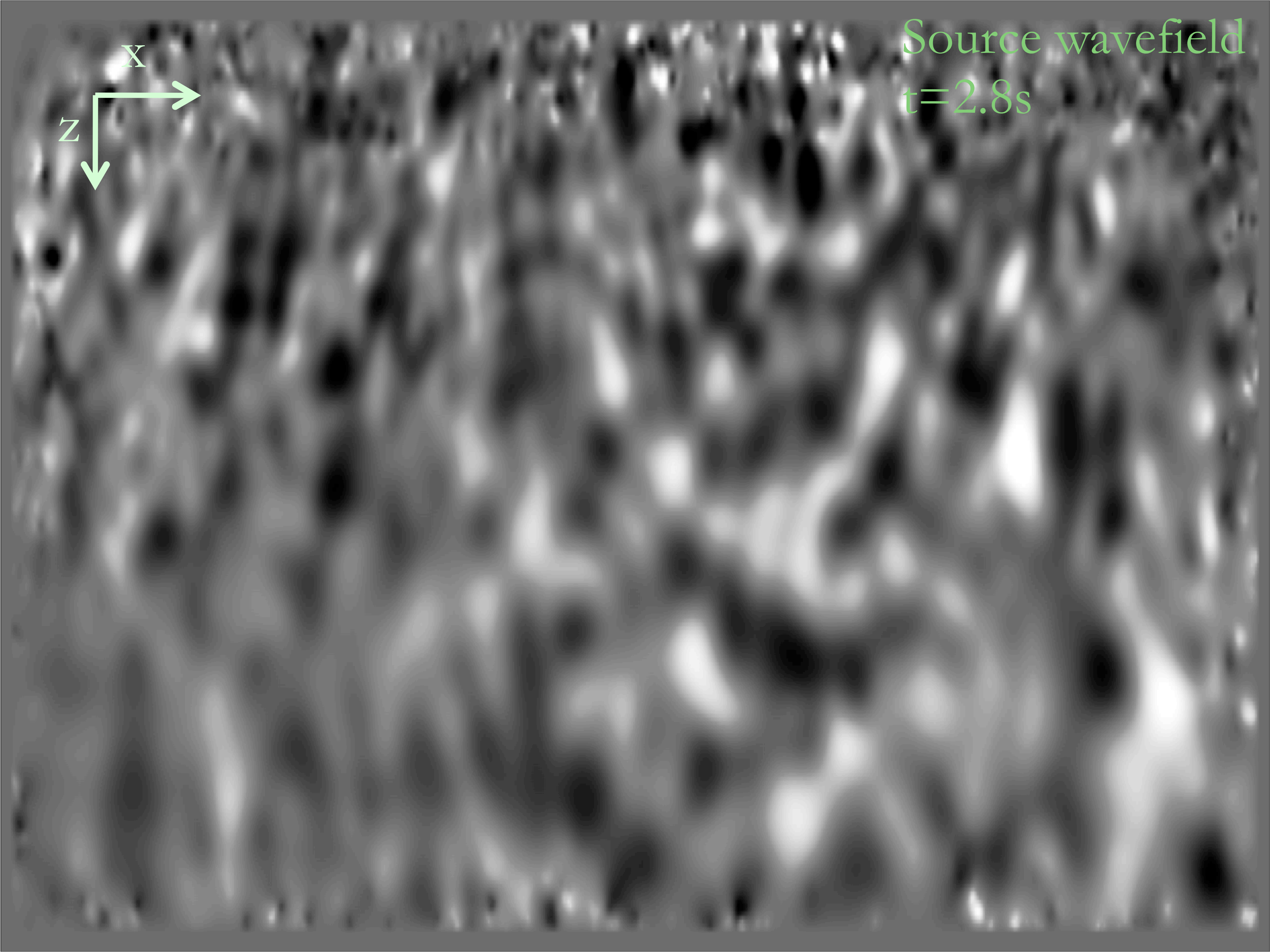

X
Z

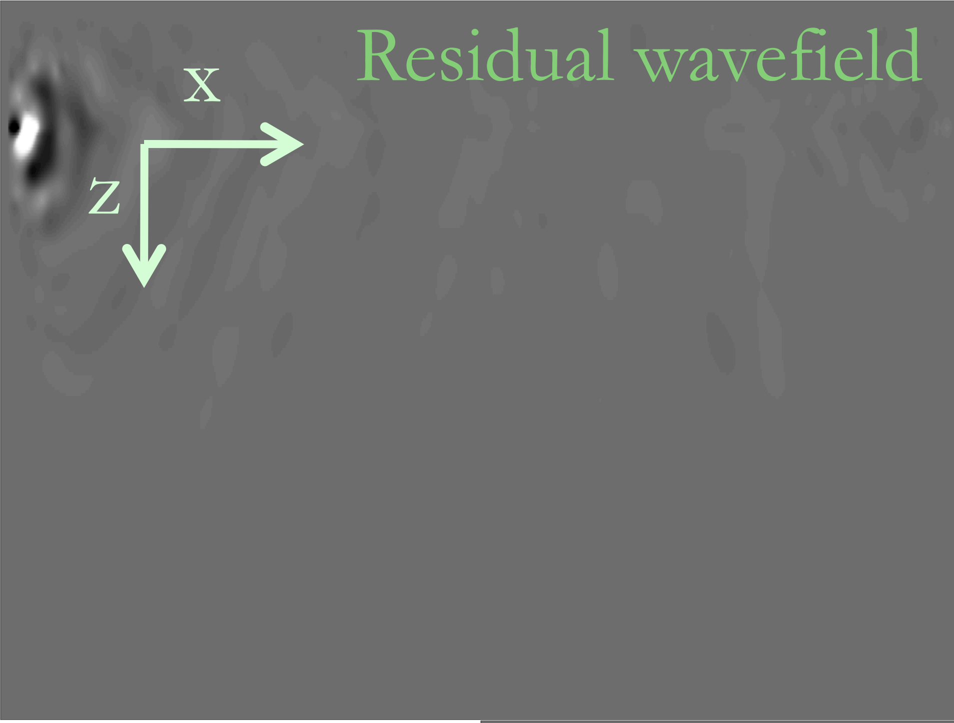


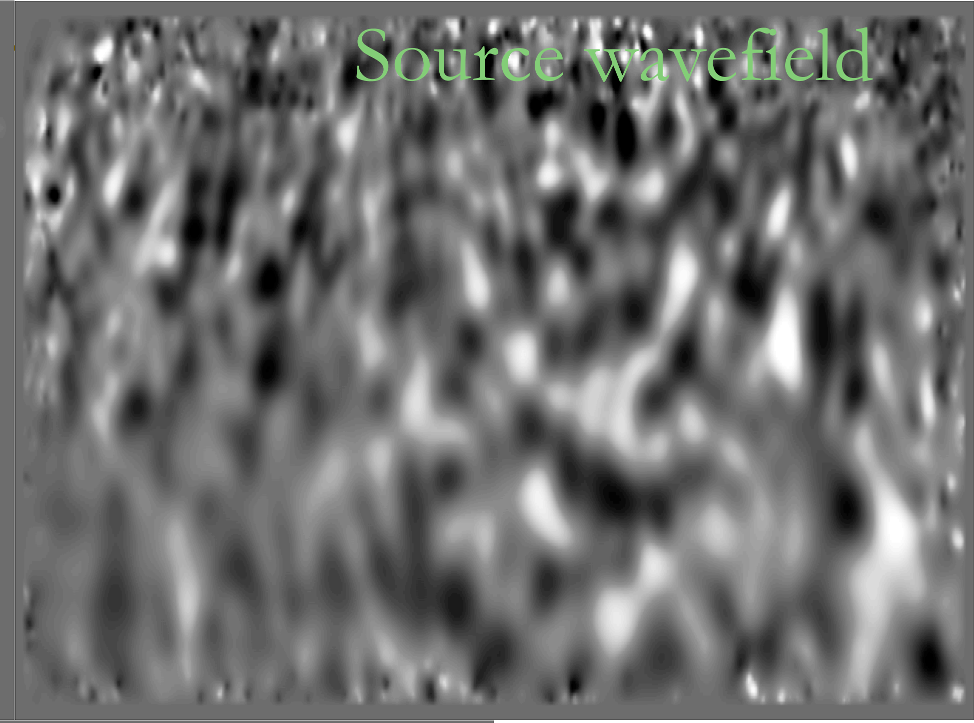
x
z
Source wavefield
t=0.4s


x
z
Source wavefield
t=1.2s

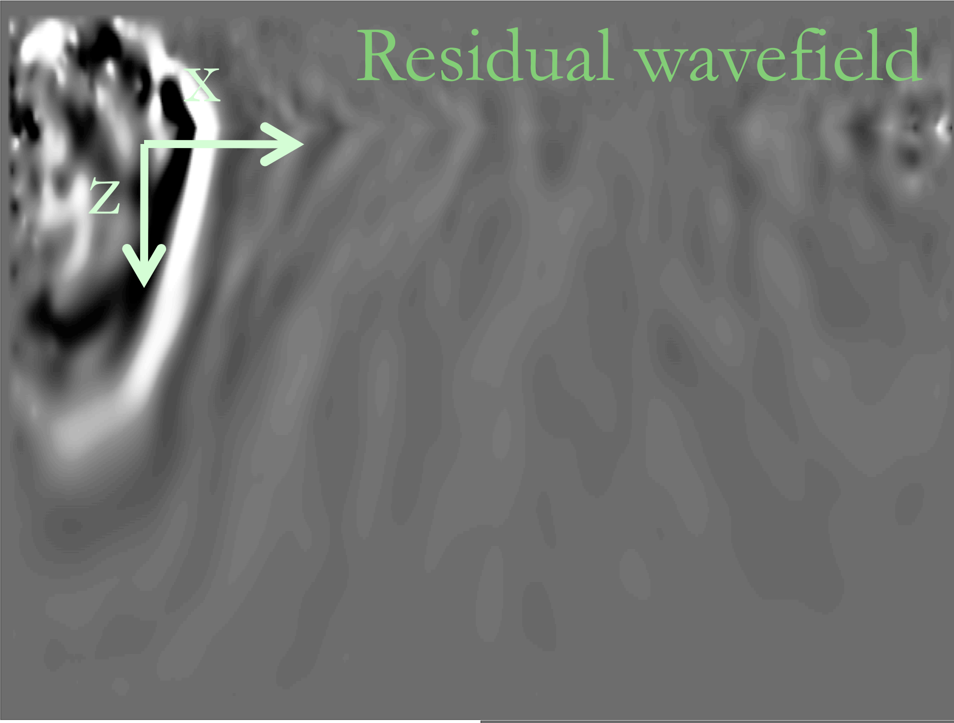

Source wavefield
t=2.0s
x
z

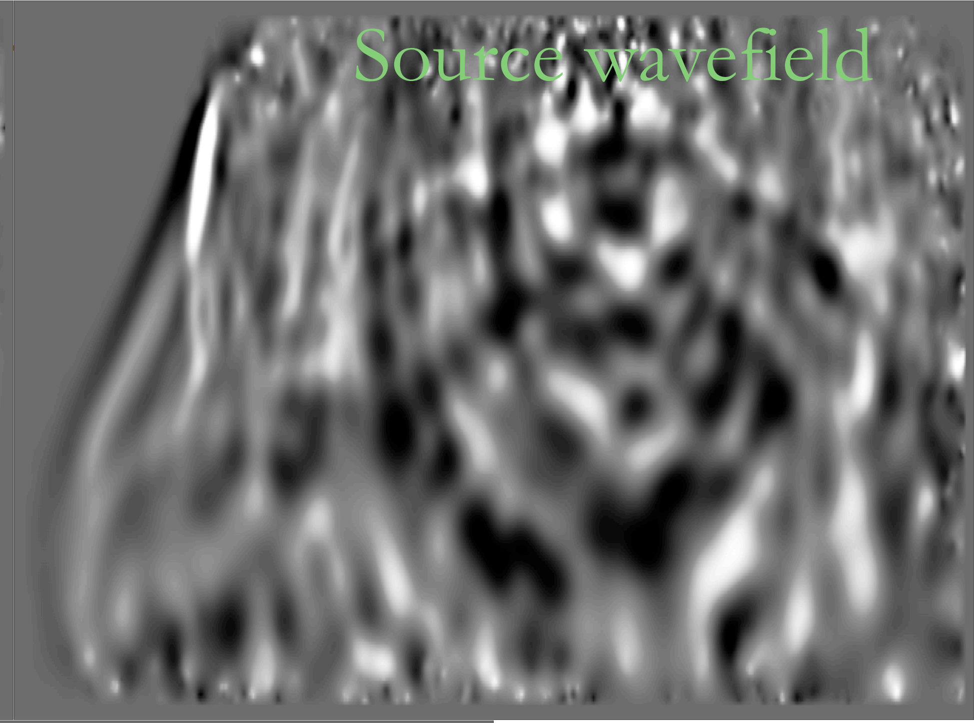

Source wavefield
t=2.8s
x
z



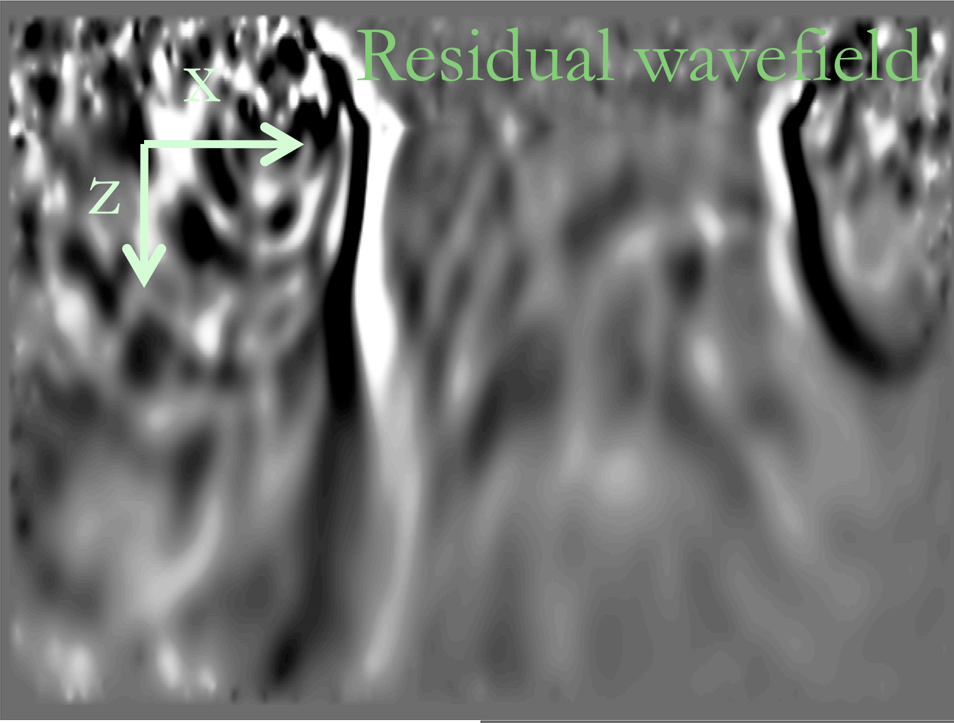


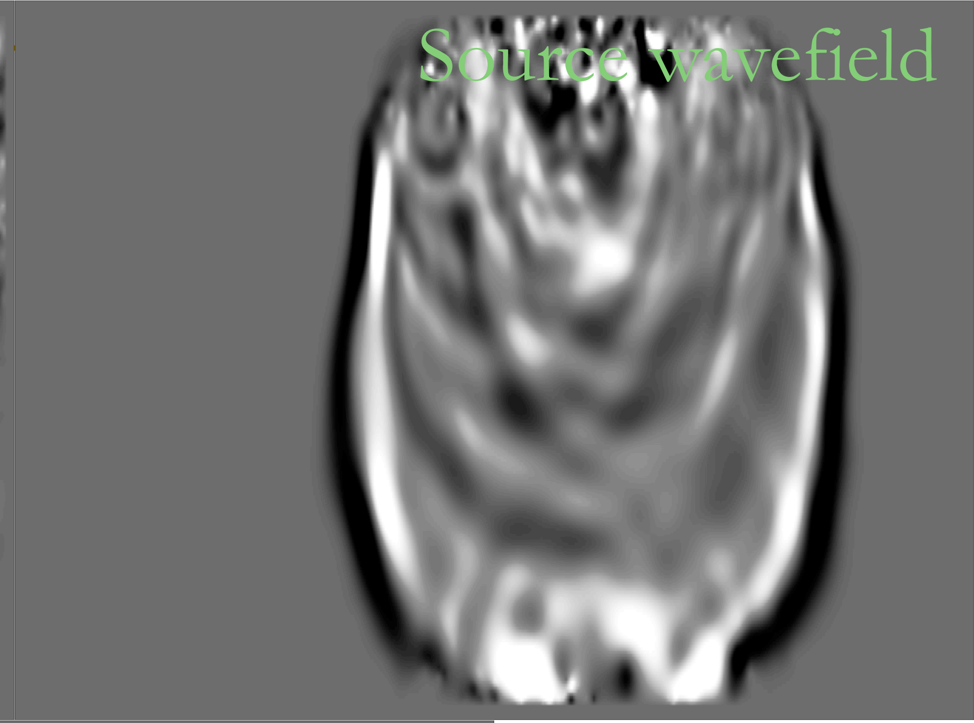
t=2.8s



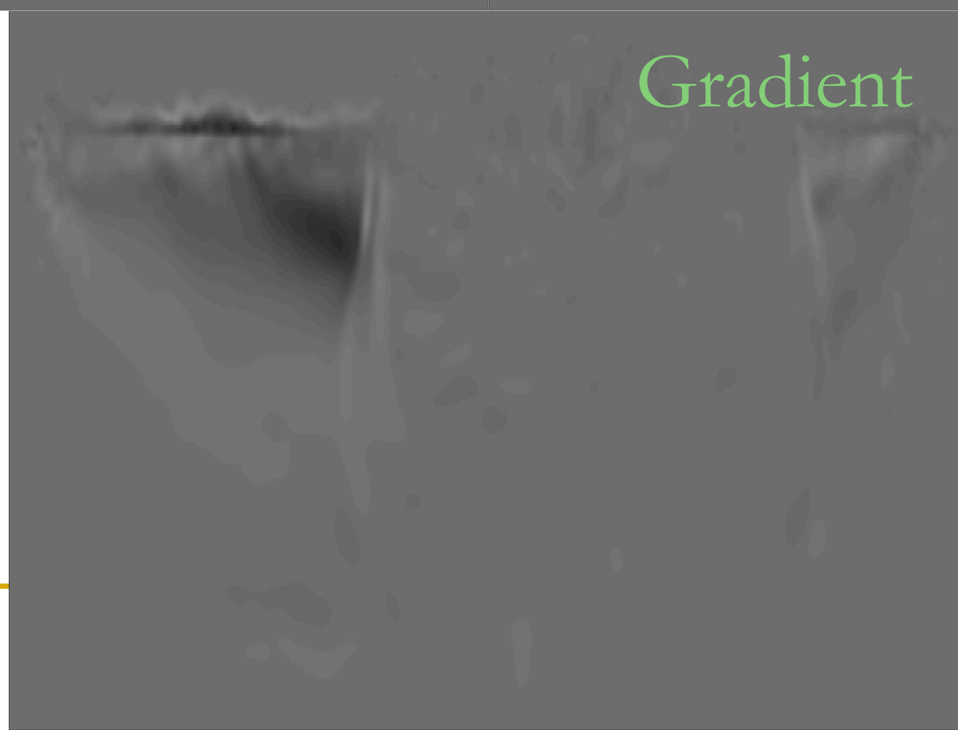


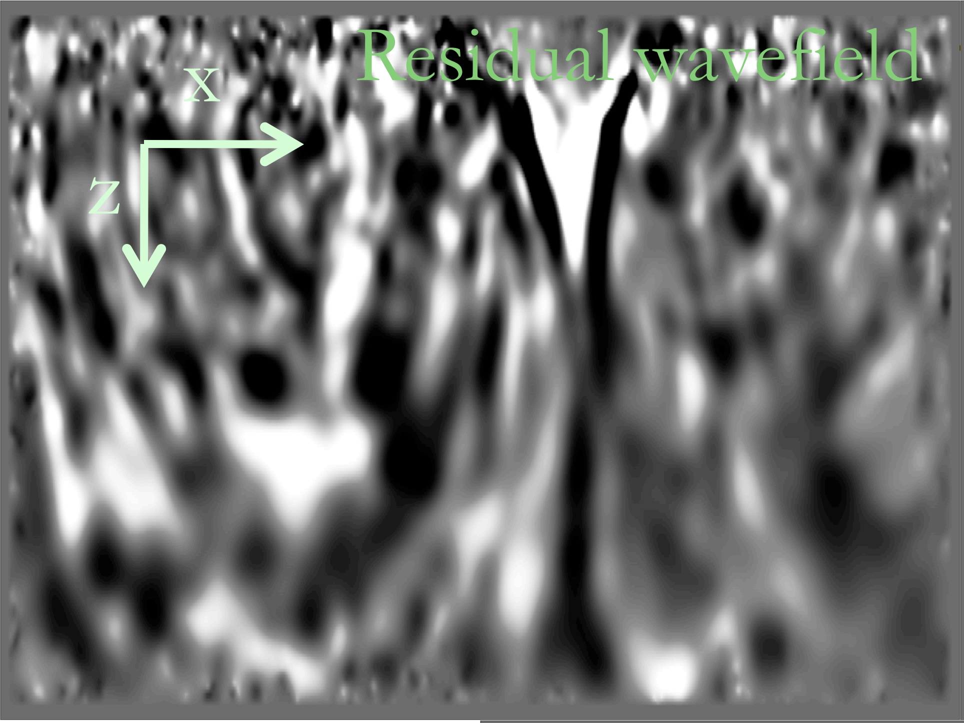
t=2.0s

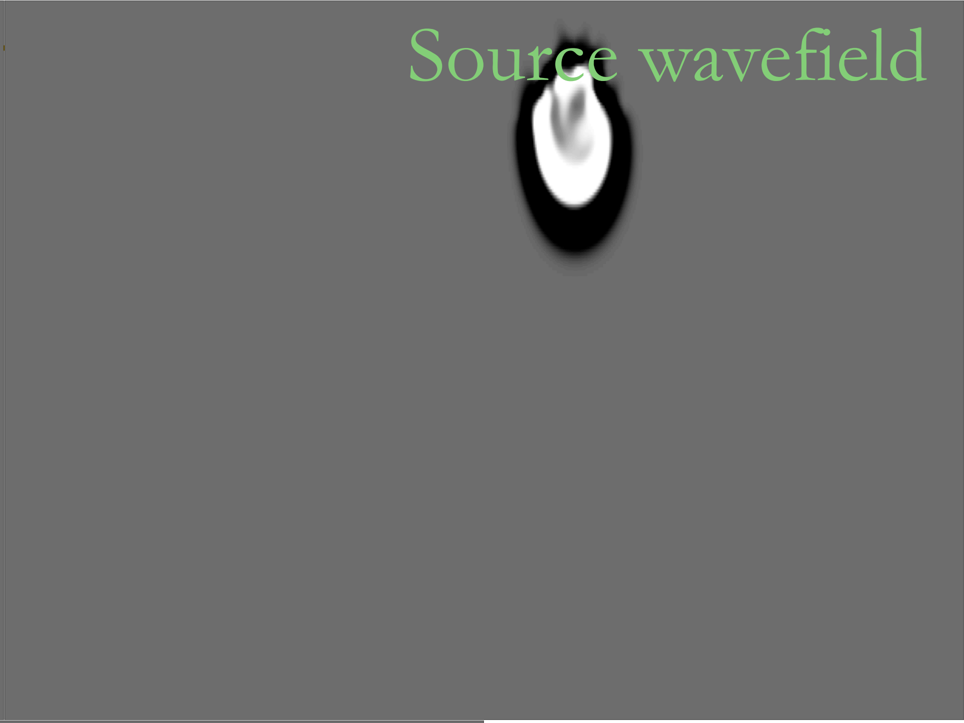




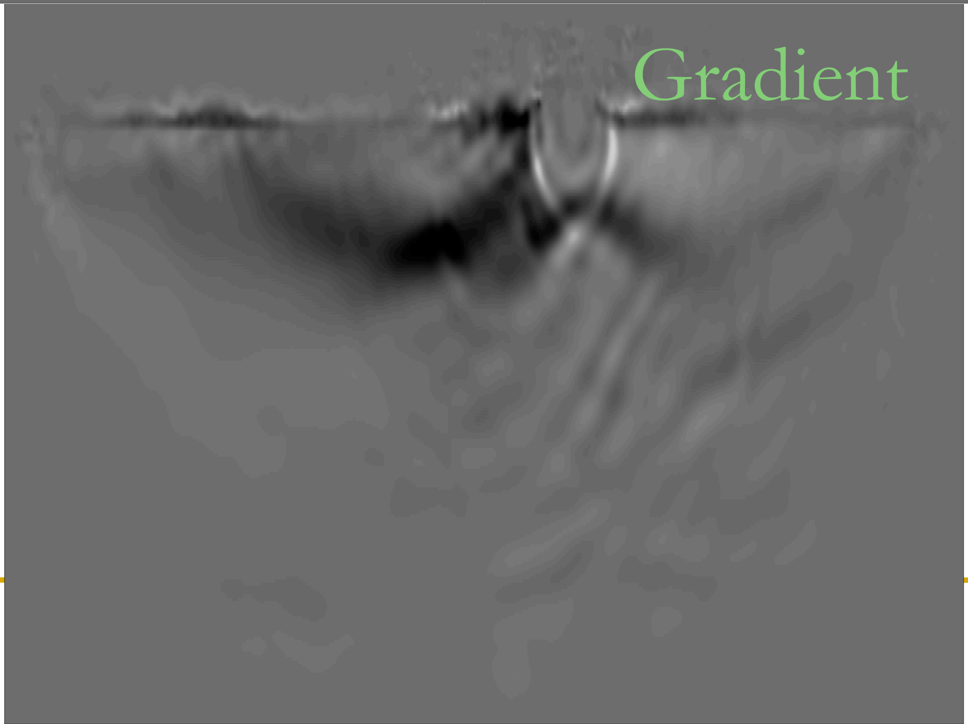
t=1.2s

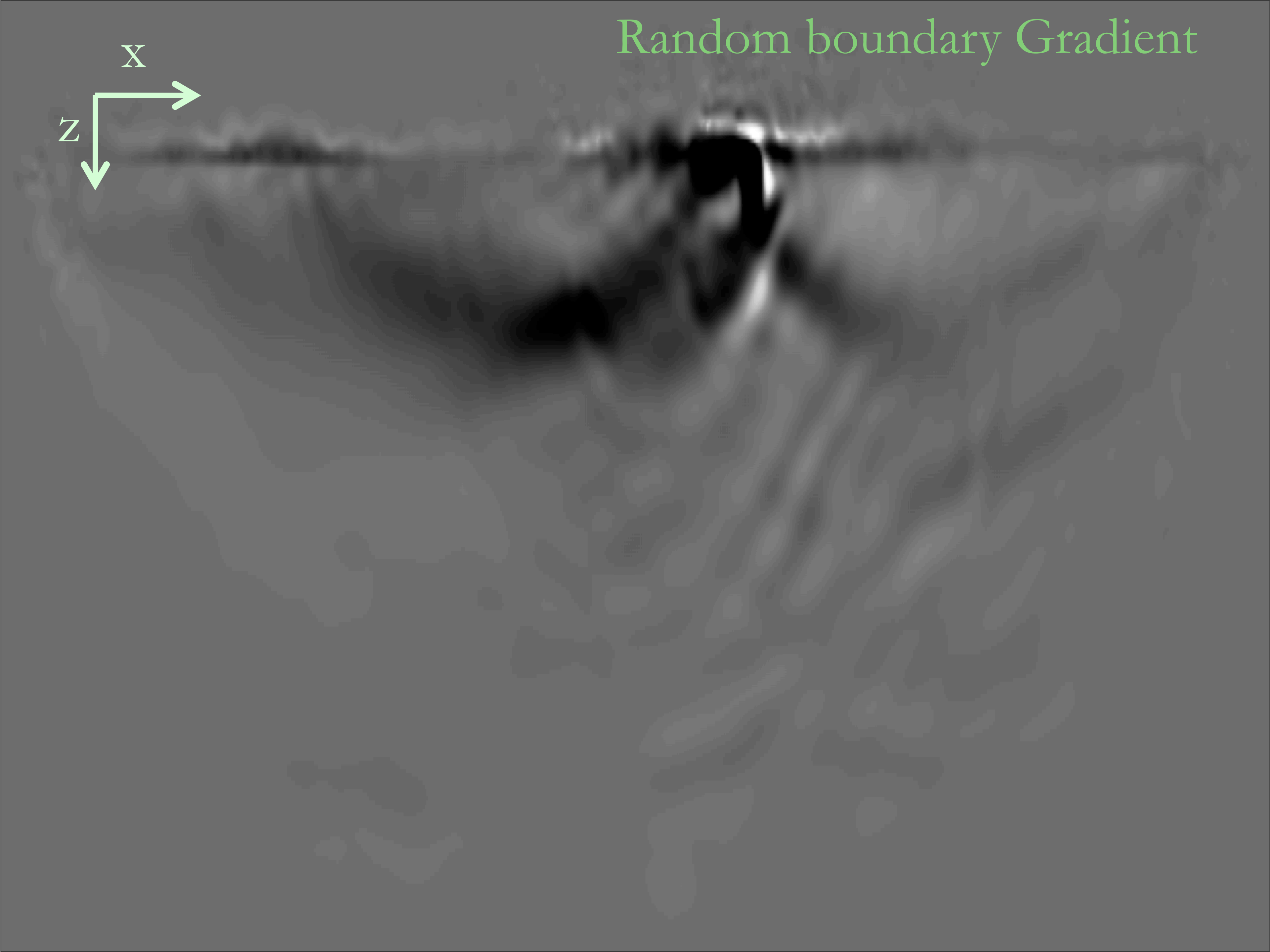


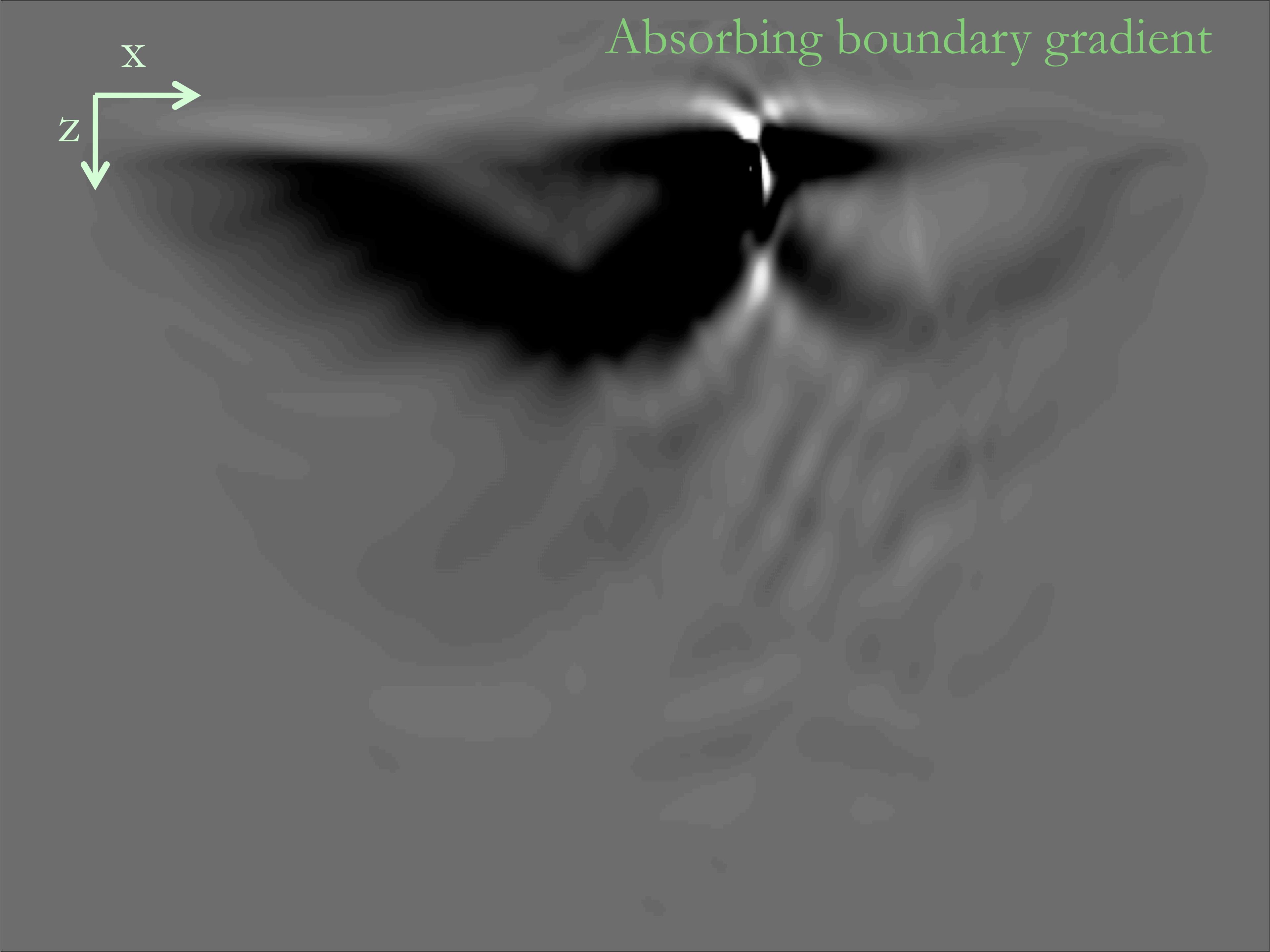




t=0.4s




Random boundary Gradient
x
z


Absorbing boundary gradient
x
z

Computational summary for 1 iteration:

(using random boundary)

6 wave propagations (4 for gradient
2 for step length)

+

2 wave field slices saving (gradient calculation)

↓

nx*ny*nz*2=500*500*500*2 ~ 1G

Computational summary for 1 iteration:

4 wave propagations (2 for gradient
2 for step length)

+

1 wave field saving (gradient calculation)

↓

nx*ny*nz*nt=500*500*500*500 ~ 250G

Trade off by using random boundary condition

More computation

Huge memory saving

Outline

1. Introduction
2. Motivation
3. Synthetic Example
4. Conclusion

Gradient calculation in FWI is very similar to RTM

Gradient calculation in FWI is very similar to RTM

↓

Similar conclusion for RTM can be drawn for FWI gradient calculation

Gradient calculation in FWI is very similar to RTM

⬇

Similar conclusion for RTM can be drawn for FWI gradient calculation

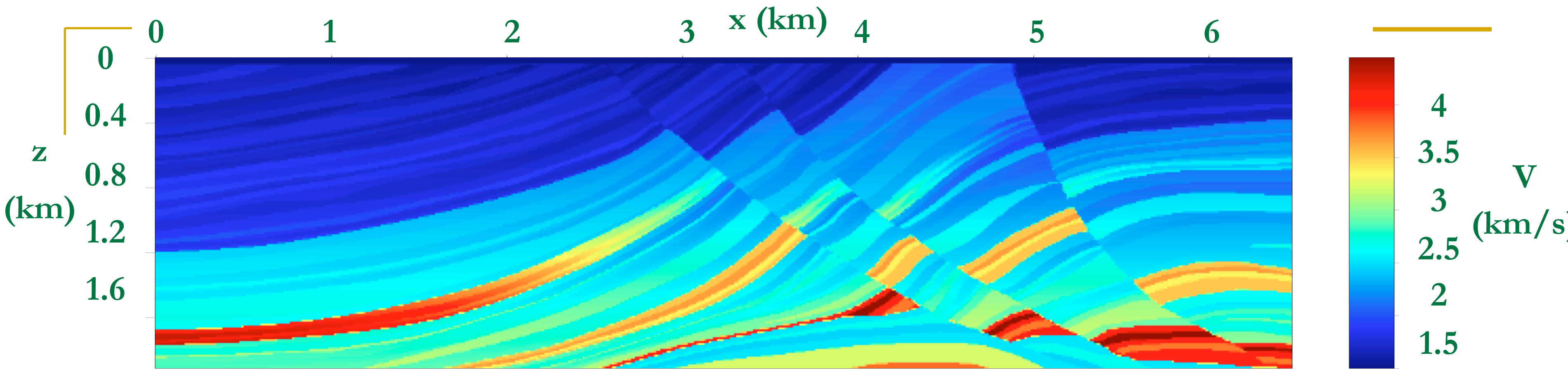
What about the iterative process?

Outline

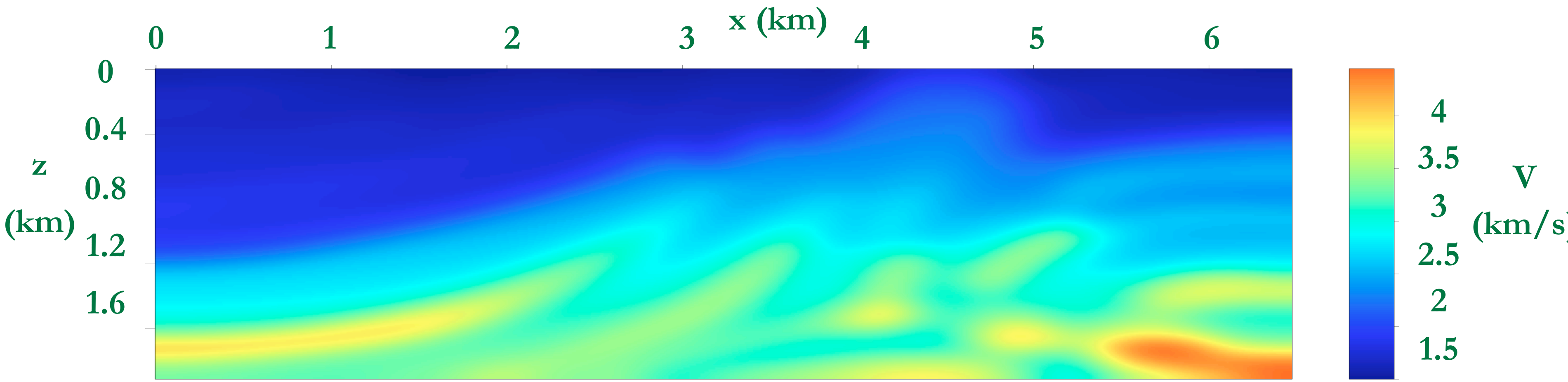
1. Introduction
2. Motivation
3. Synthetic Example
4. Conclusion

Synthetic example with three different boundary conditions:

a. Absorbing boundary condition
b. Random boundary condition
c. Constant (Zero randomness) boundary condition



True model



Initial model

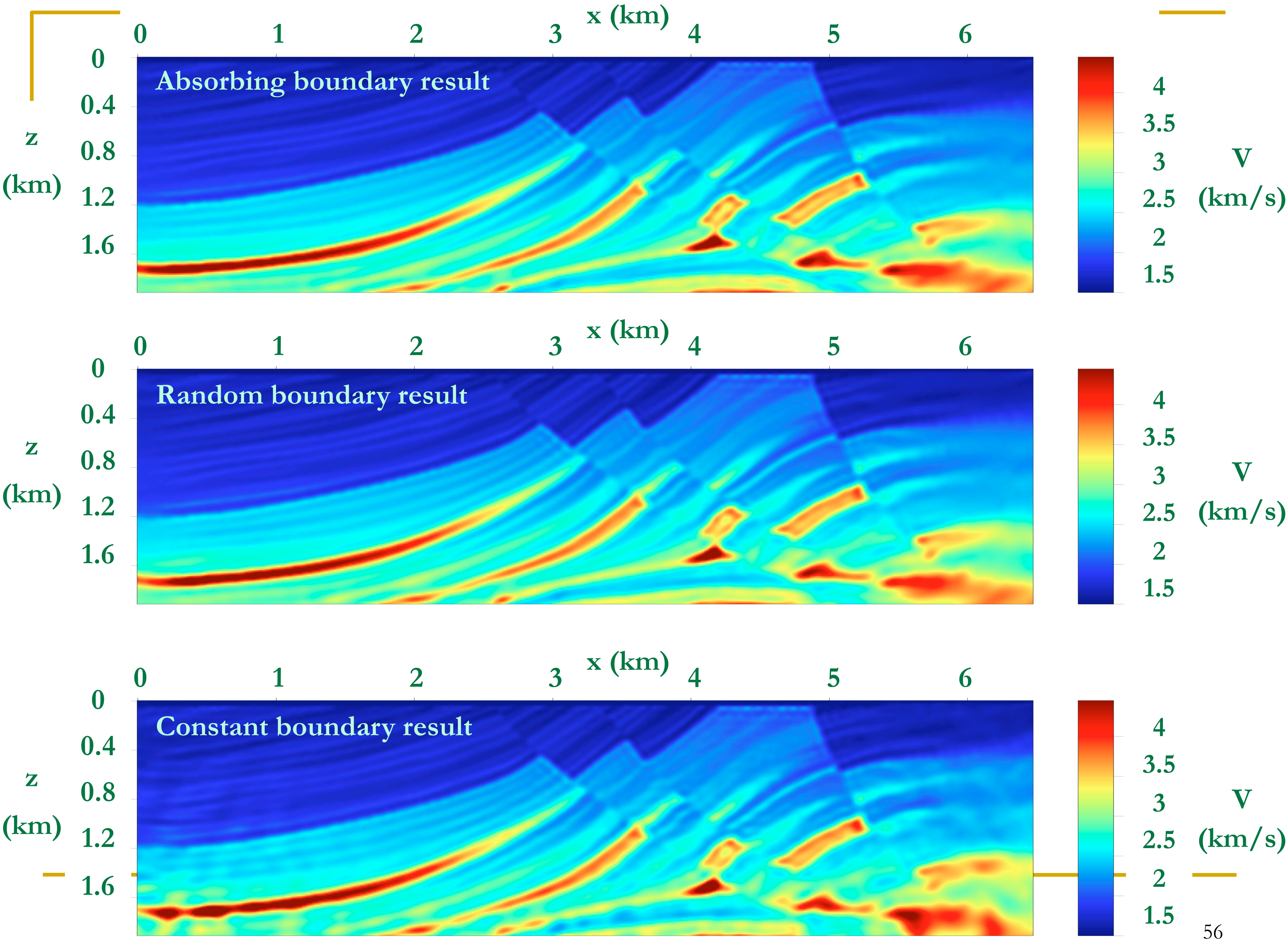
Inversion parameter

160*540 grid points, 12 m spacing

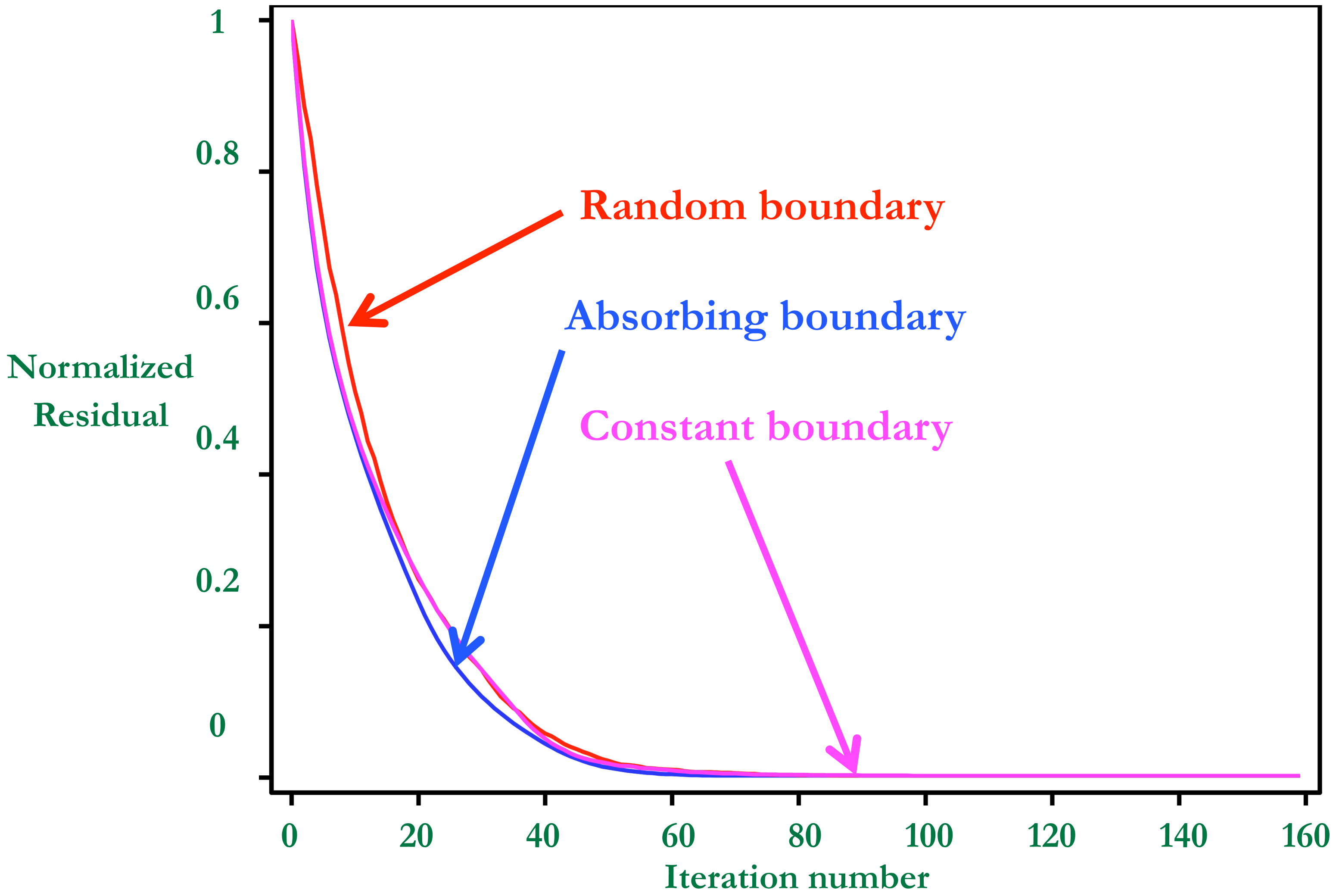
60 shots, 84 meter spacing

7Hz peak frequency source

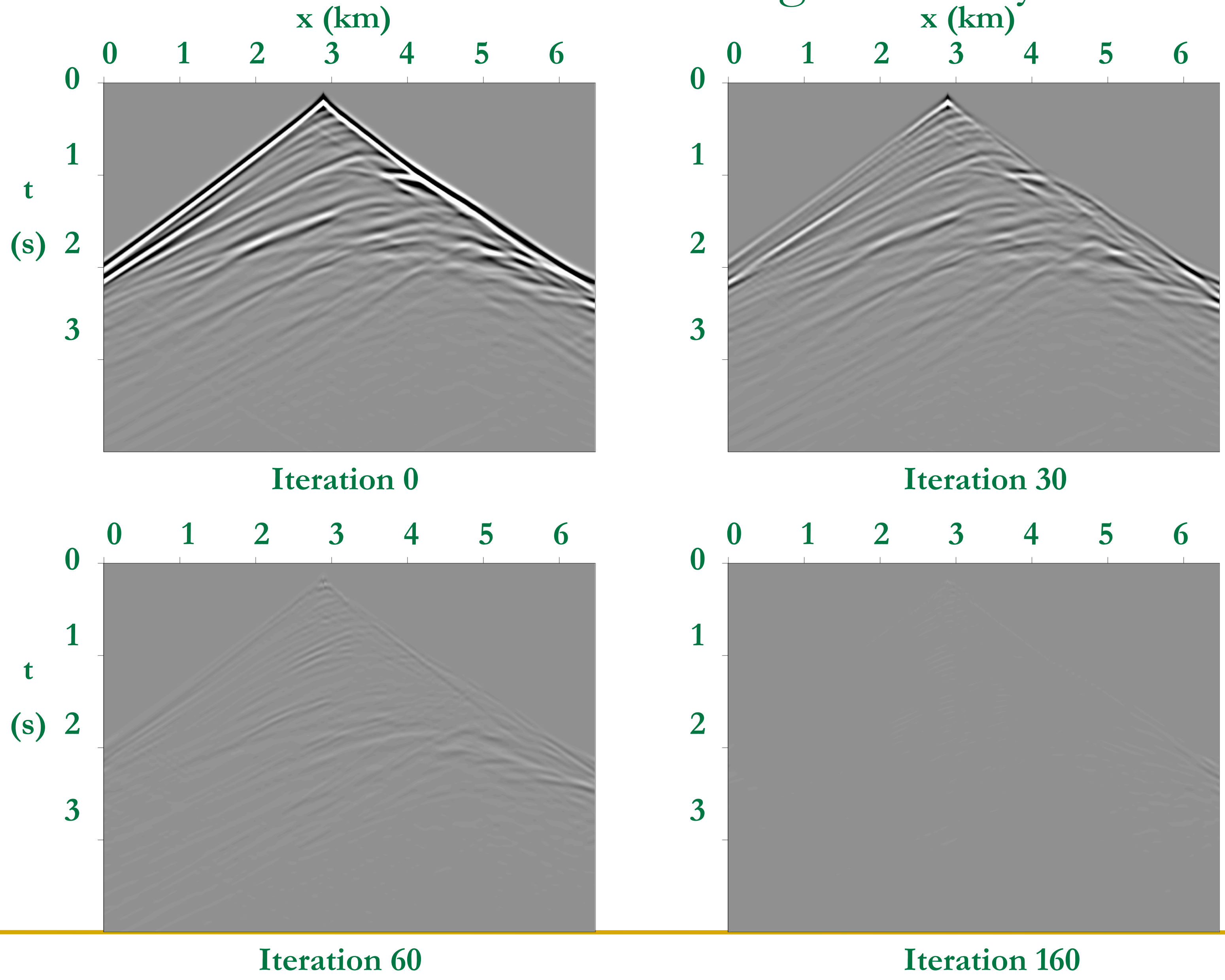
Receivers everywhere on the surface


x (km)
0
1
2
3
4
5
6
0
0.4
0.8
1.2
1.6
z
(km)
Absorbing boundary result
4
3.5
3
2.5
2
1.5
V
(km/s)
x (km)
Random boundary result
x (km)
Constant boundary result

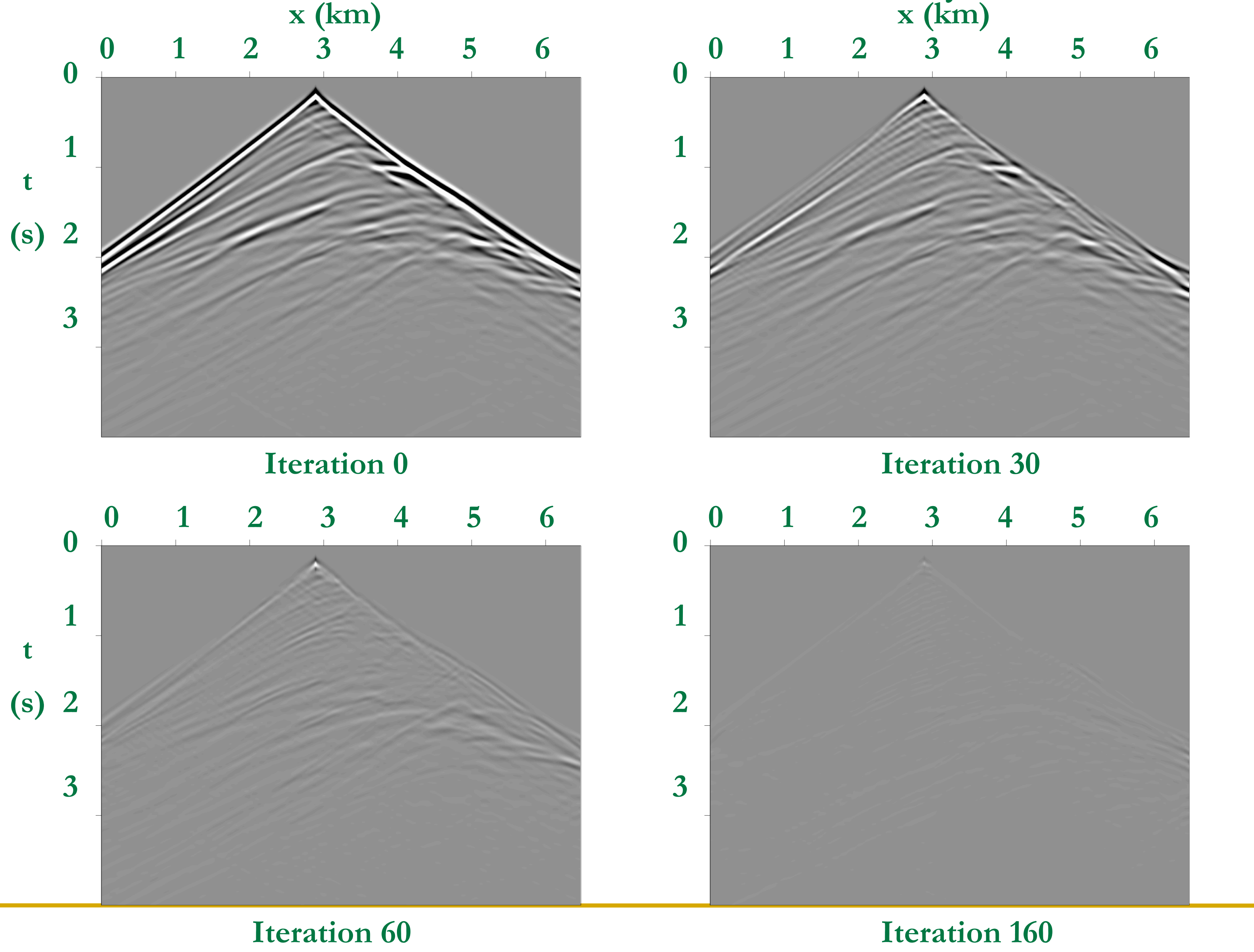
RMS Residual comparison



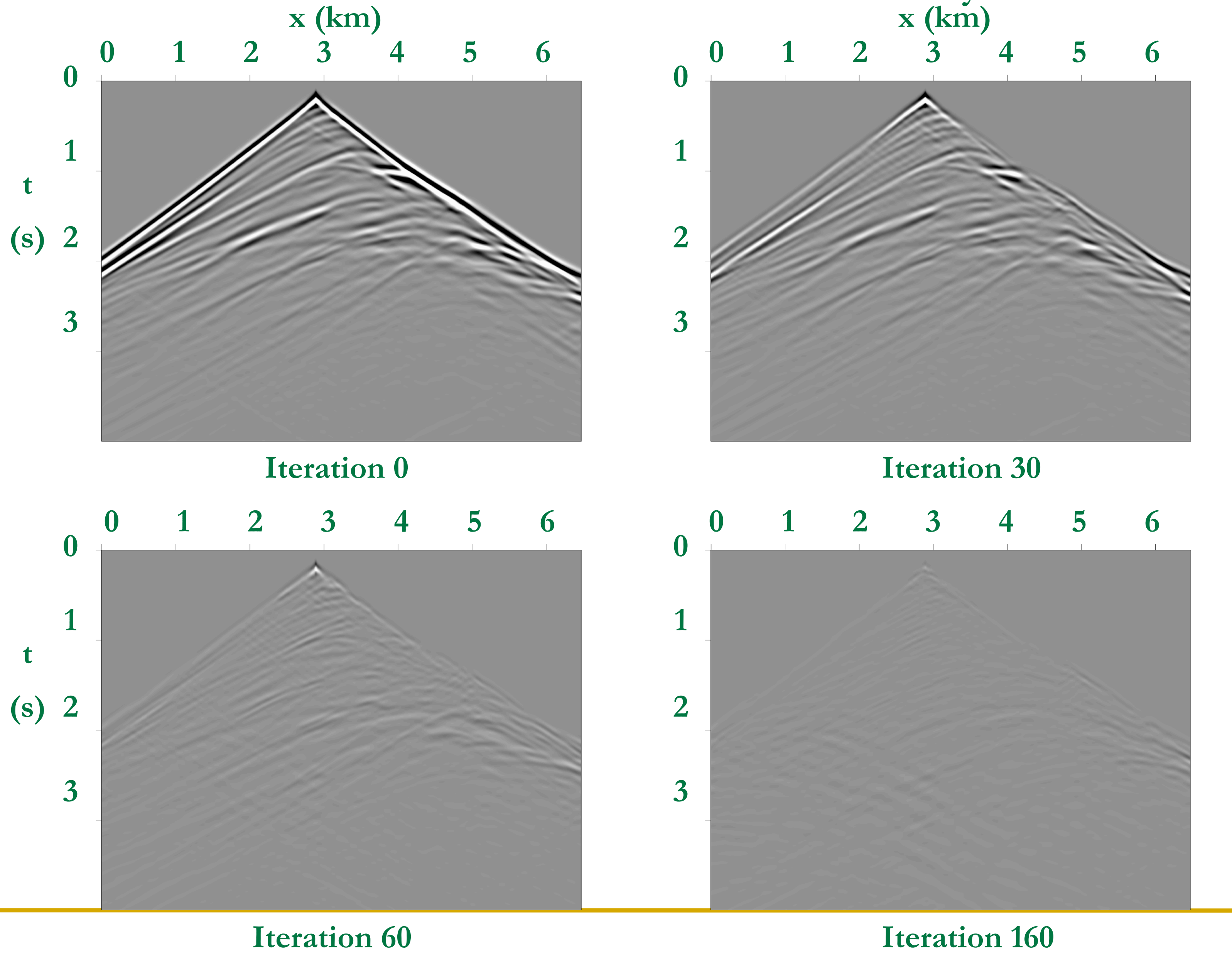
Central shot residual with absorbing boundary



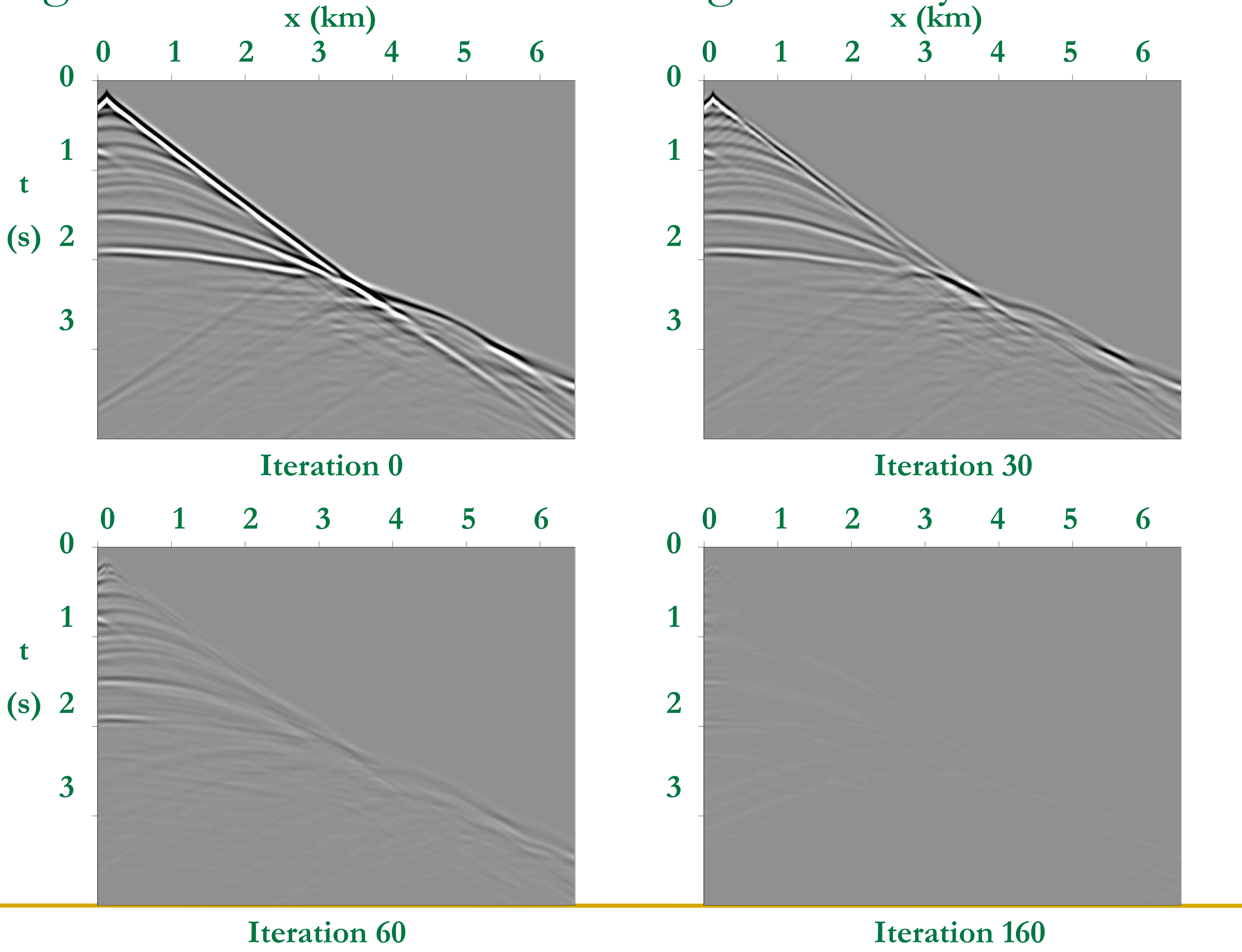
Central shot residual with random boundary



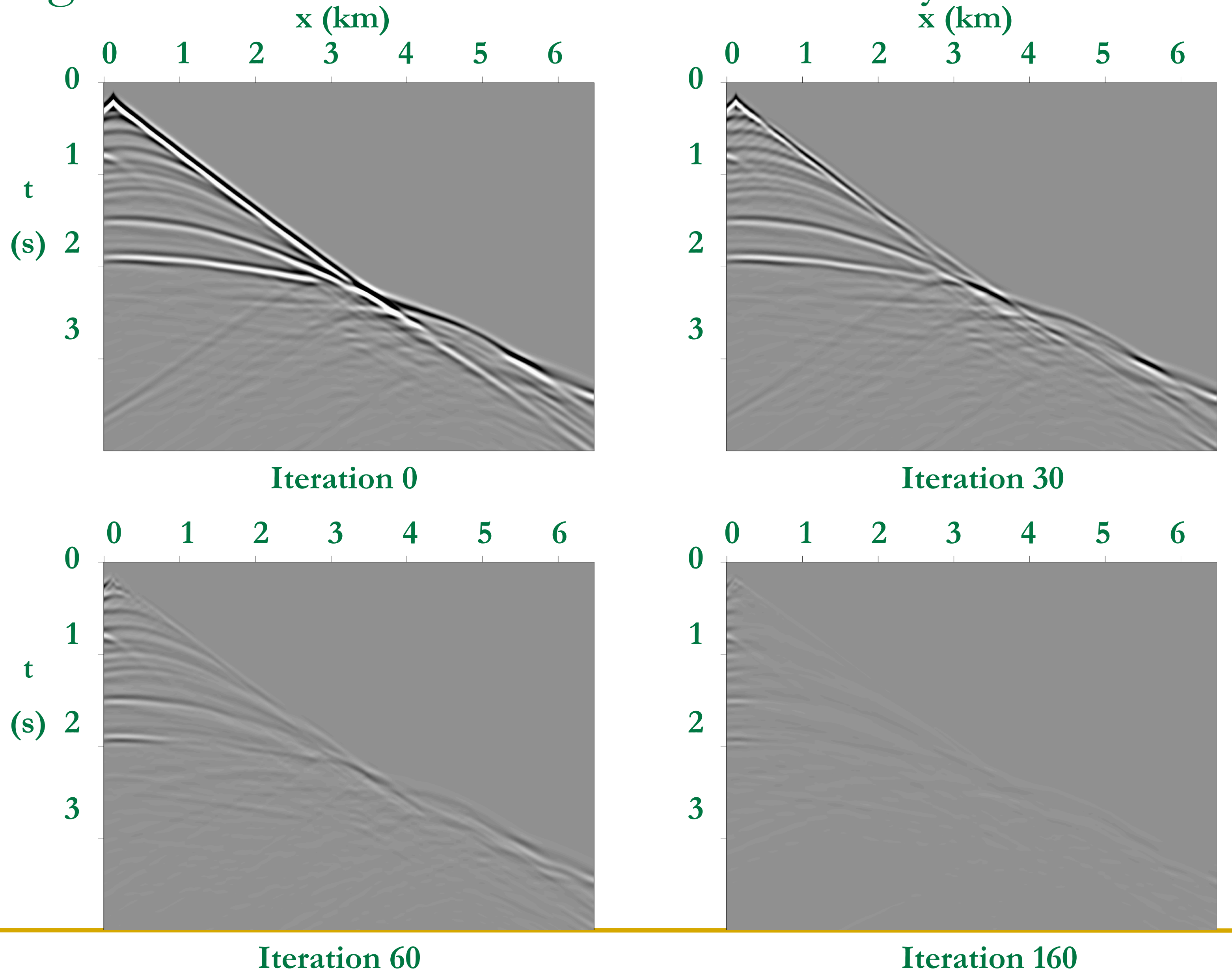
Central shot residual with constant boundary



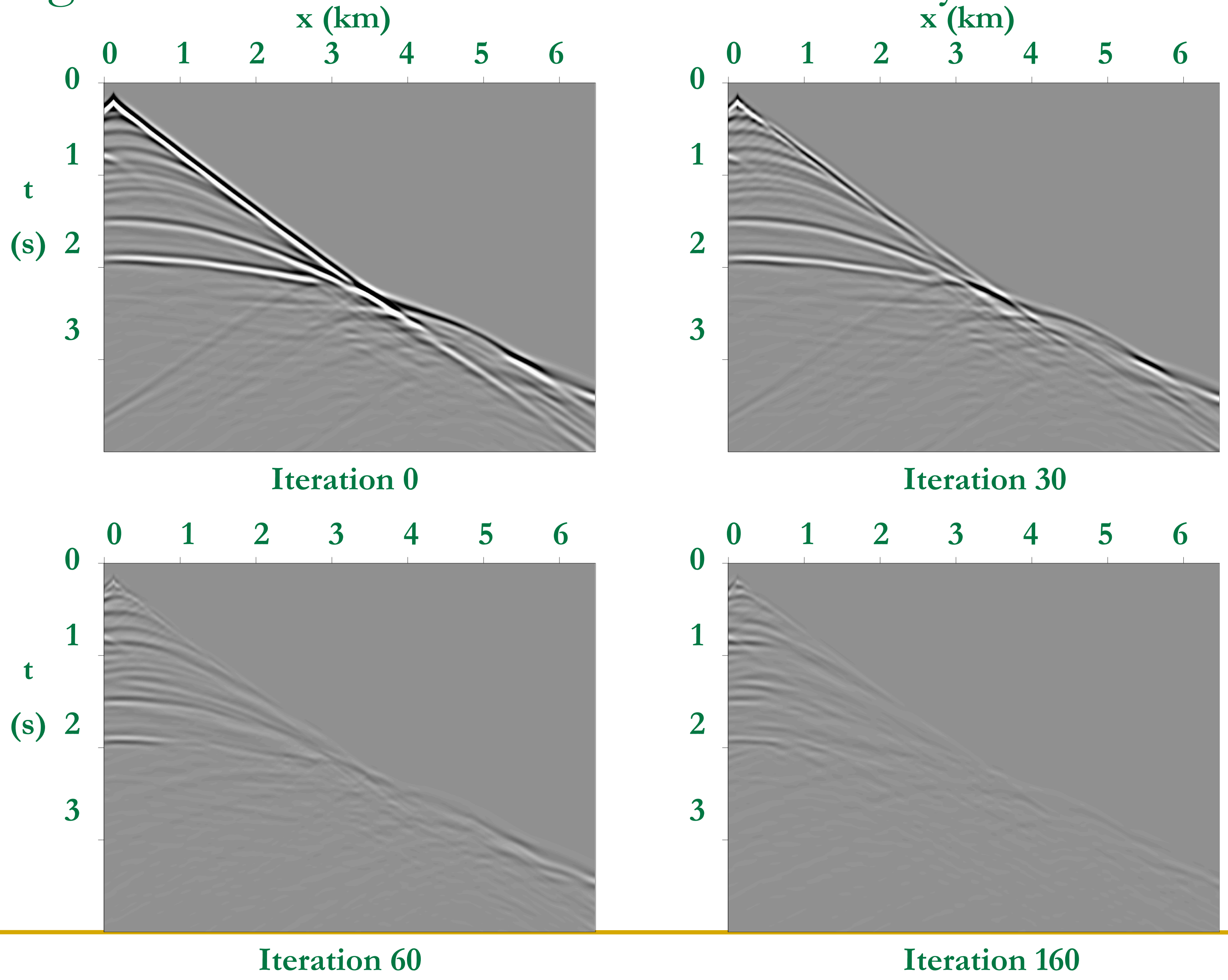
Edge shot residual with absorbing boundary



Edge shot residual with random boundary



Edge shot residual with constant boundary



Outline

1. Introduction
2. Motivation
3. Synthetic Example
4. Conclusion

Conclusions

Waveform inversion using a:

Random boundary requires extra wave propagations, but significantly reduces memory requirements

Conclusions

Waveform inversion using a:

Random boundary requires extra wave propagations, but significantly reduces memory requirements

Random boundary is almost as accurate as when using an absorbing boundary

Conclusions

Waveform inversion using a:

Random boundary requires extra wave propagations, but significantly reduces memory requirements

Random boundary is almost as accurate as when using an absorbing boundary

Constant boundary is almost as accurate as when using an absorbing boundary, provided there are sufficient data constraints

Acknowledgements

- SEP sponsors for the financial support of this research

Thank you

Questions & Suggestions