3D elastic seismic modeling using CUDA unified memory and gaming GPUs

Tore S. Bergslid[1], Børge Arntsen[1] and Espen Birger Raknes[2,3]

1: NTNU, Department of Geoscience and Petroleum
2: Aker BP
3: NTNU, Department of Electronic Systems

E-mail: tore.bergslid@ntnu.no

Trondheim
April 24th 2018

NTNU – Trondheim
Norwegian University of Science and Technology

Outline

Introduction

- Efficient and accurate modeling of seismic data is very important for the oil and gas industry today.

Introduction

- Efficient and accurate modeling of seismic data is very important for the oil and gas industry today.
- FD, FEM, SEM, etc.

Introduction

- Graphics processing units can accelerate seismic modeling.

Introduction

- Graphics processing units can accelerate seismic modeling.
- CUDA? OpenCL?

Introduction

- Graphics processing units can accelerate seismic modeling.
- CUDA? OpenCL?
- Venstad, out-of-core.

Theory

- The elastic wave equation can be written as:

$$\rho \ddot{u}_i - \partial_j c_{ijkl} \partial_l u_k = f_i, \tag{1a}$$

$$c_{ijkl} \partial_l u_k n_j = T_i. \tag{1b}$$

Theory

- The elastic wave equation can be written as:

$$\rho \ddot{u}_i - \partial_j c_{ijkl} \partial_l u_k = f_i, \tag{1a}$$

$$c_{ijkl} \partial_l u_k n_j = T_i. \tag{1b}$$

- Solved by the FD method, velocity-stress scheme, with non-splitting CFS PML ABC.

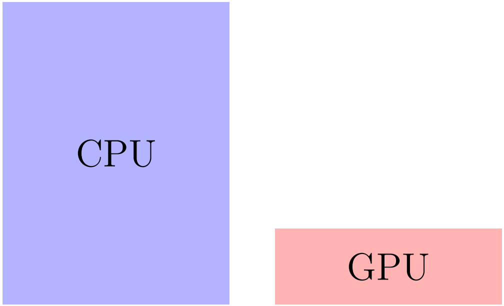
Theory

- CUDA Unified Memory

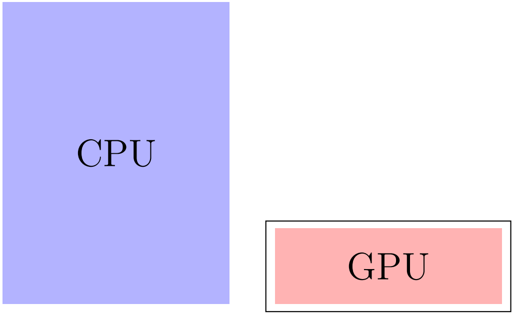
Theory

- CUDA Unified Memory
- Pascal architecture

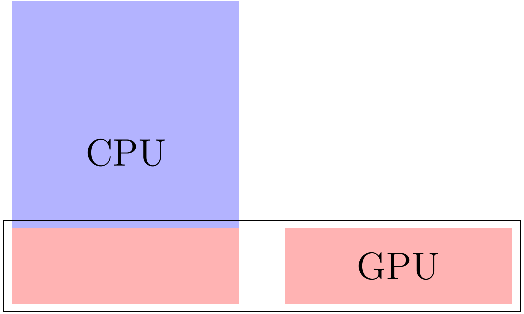
CUDA Unified Memory



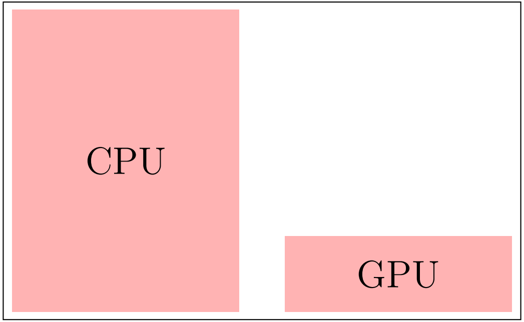
CUDA Unified Memory



CUDA Unified Memory



CUDA Unified Memory

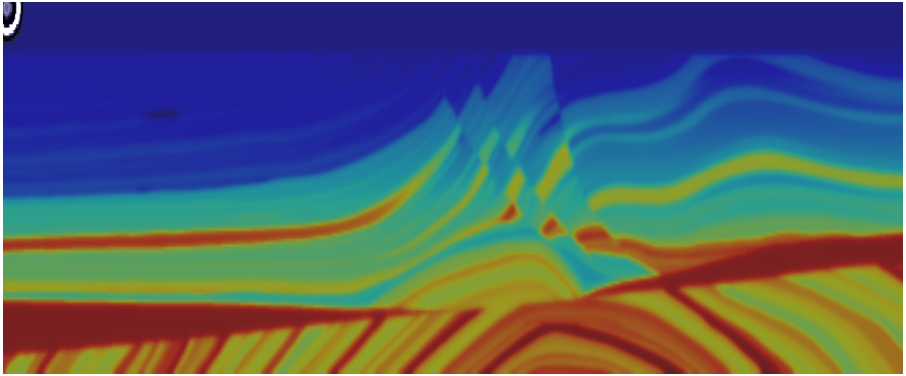


Hardware

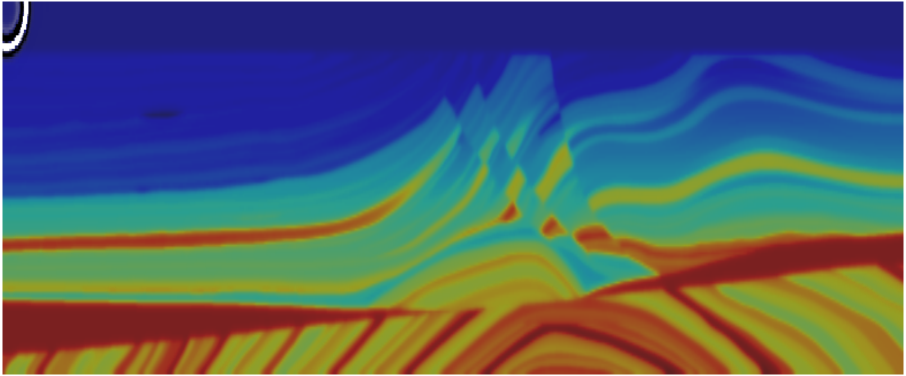
- “Maur” (“Ant”)
- 21 nodes
- 2 × Intel Xeon E5-2660 10-core CPUs
- 2 × Nvidia GTX Titan X GPU
- 128 GB RAM

Photo: NTNU HPC

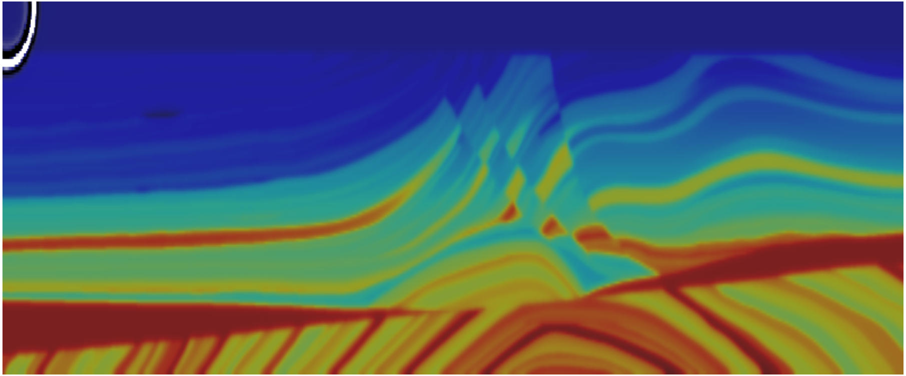
Modeling results



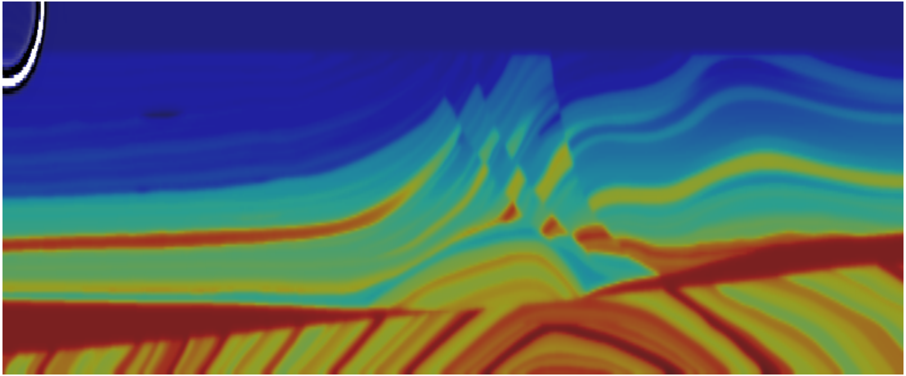
Modeling results



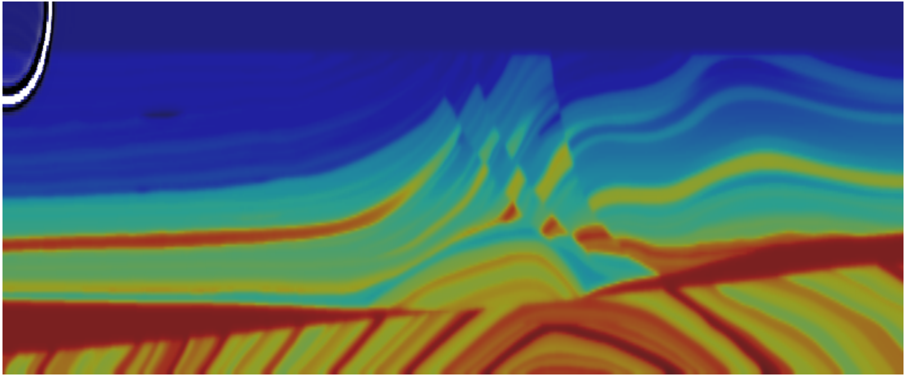
Modeling results



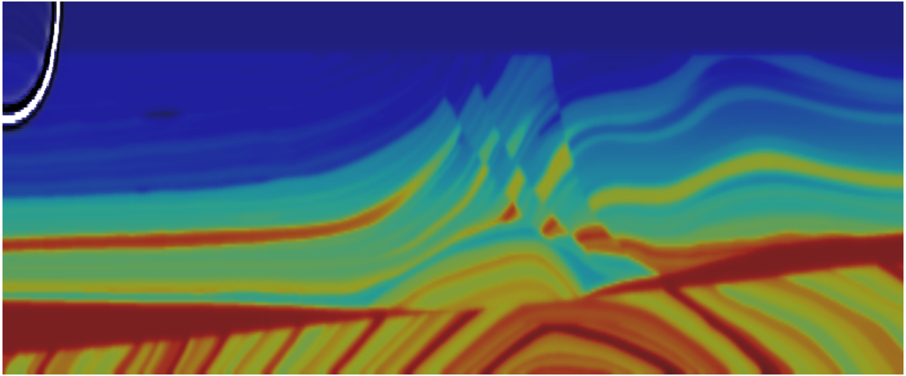
Modeling results



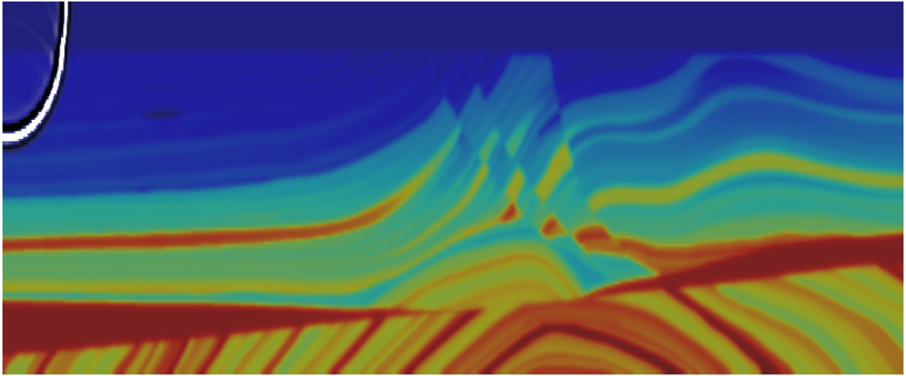
Modeling results



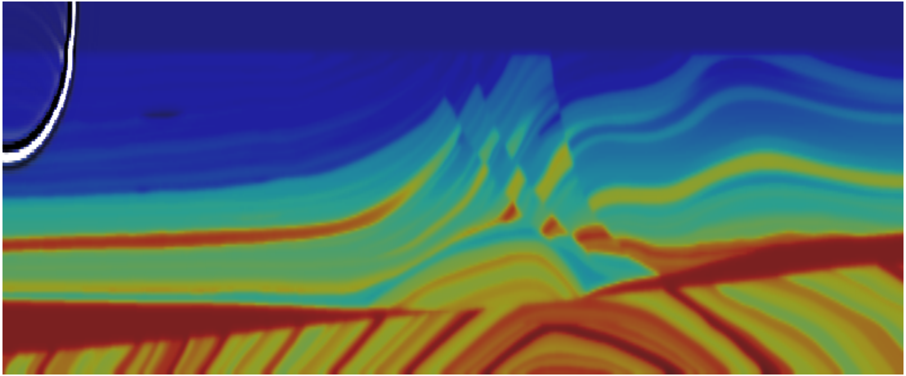
Modeling results



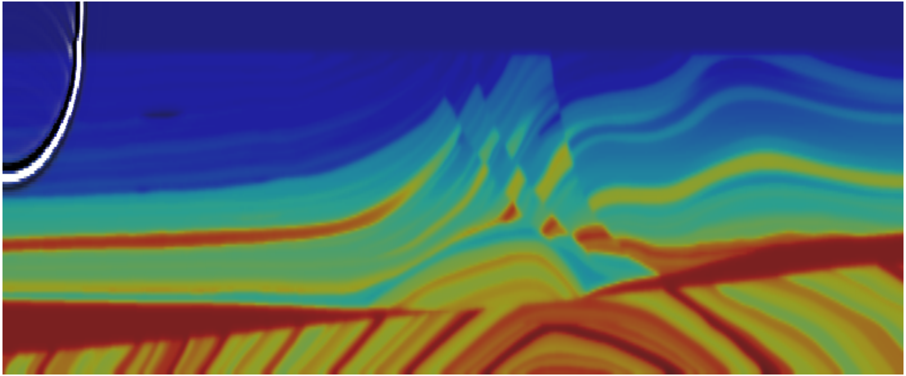
Modeling results



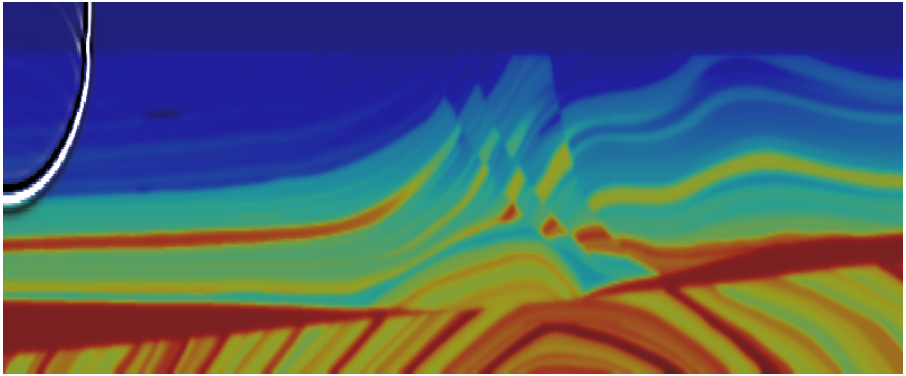
Modeling results



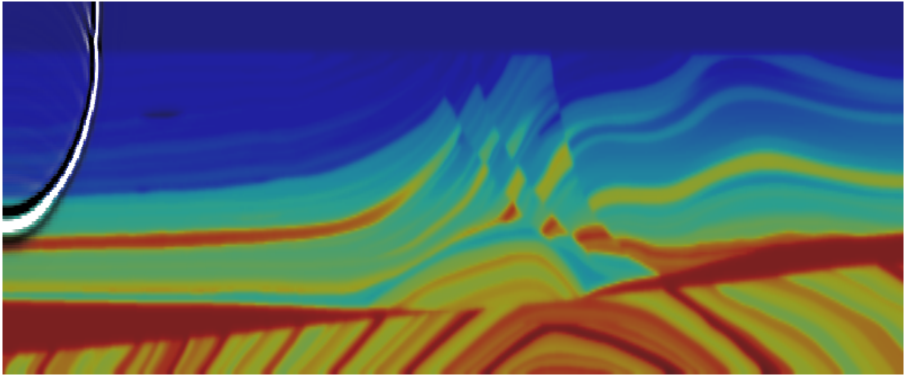
Modeling results



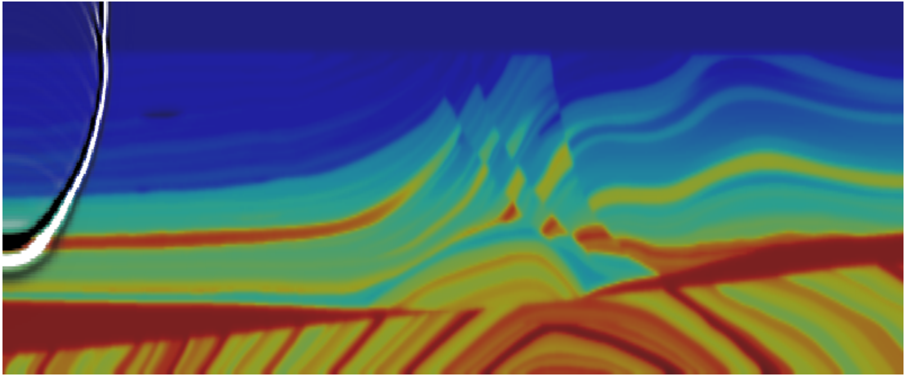
Modeling results



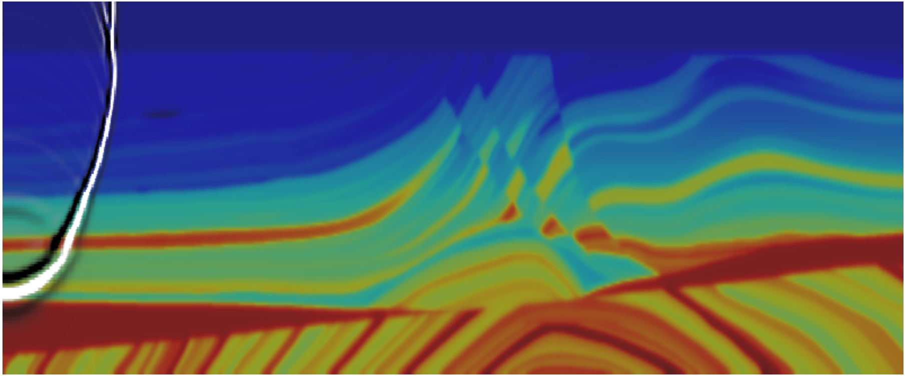
Modeling results



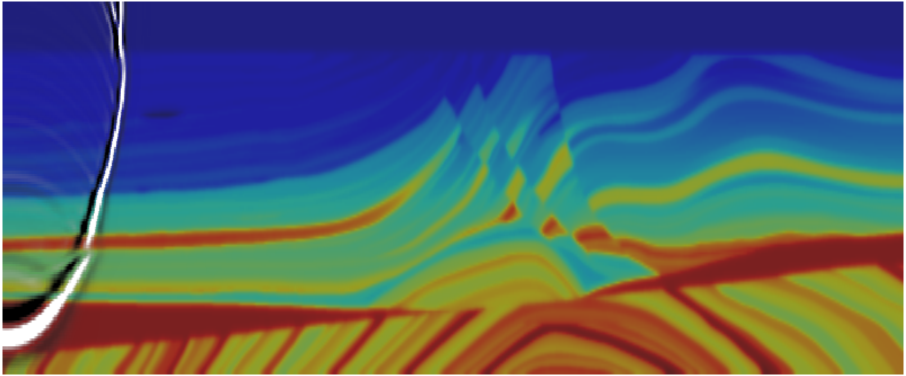
Modeling results



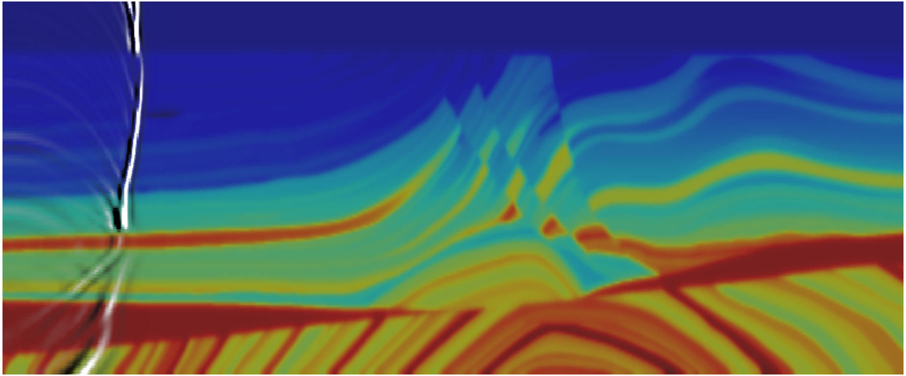
Modeling results



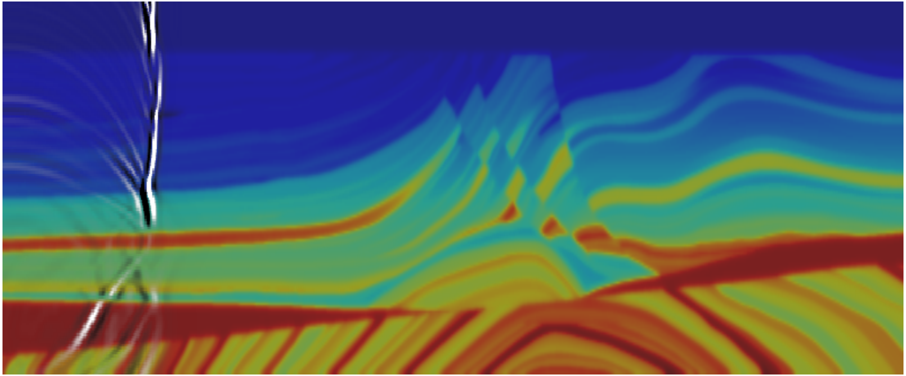
Modeling results



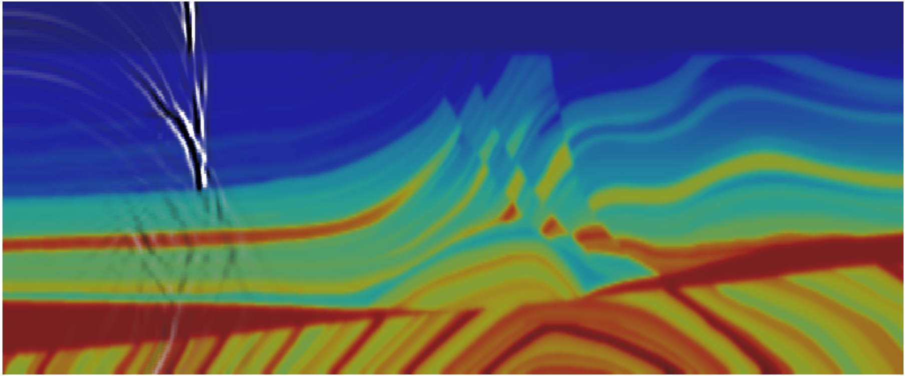
Modeling results



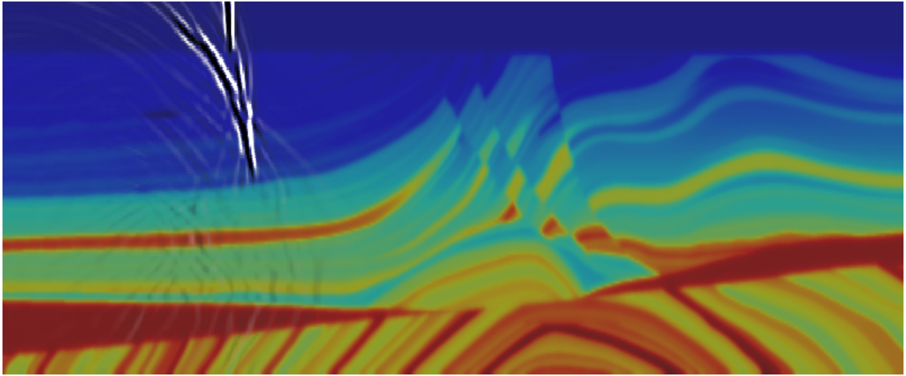
Modeling results



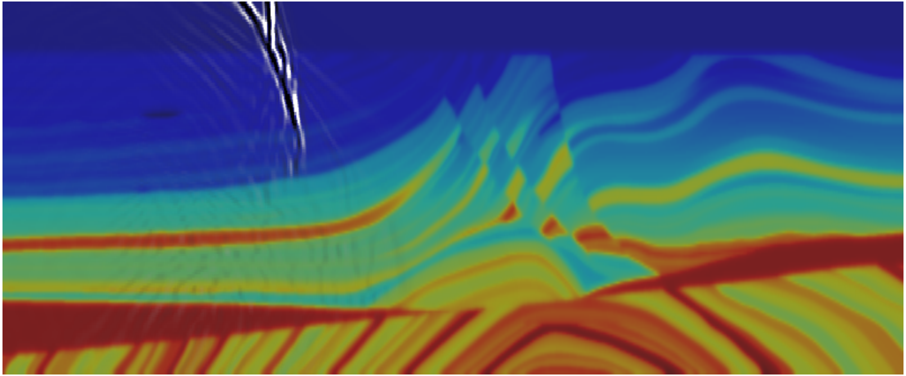
Modeling results



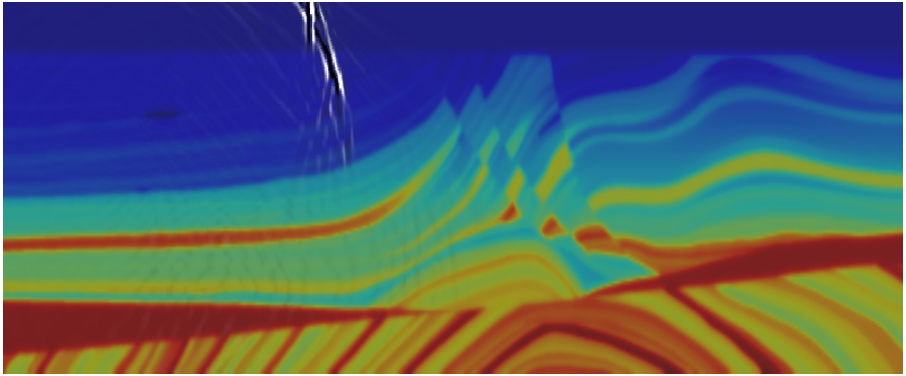
Modeling results



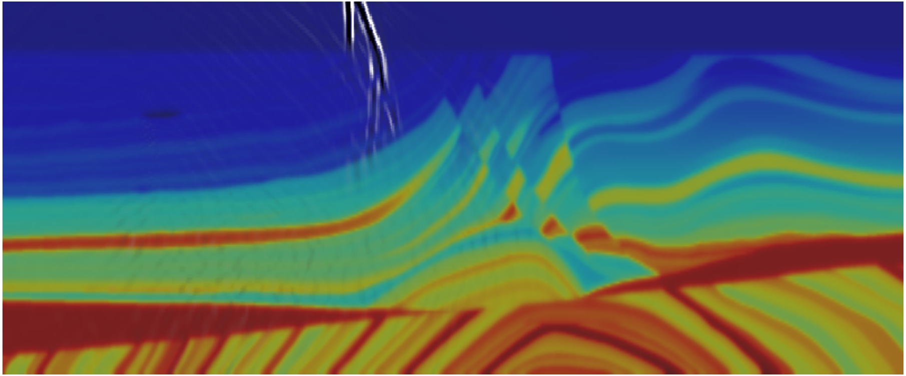
Modeling results



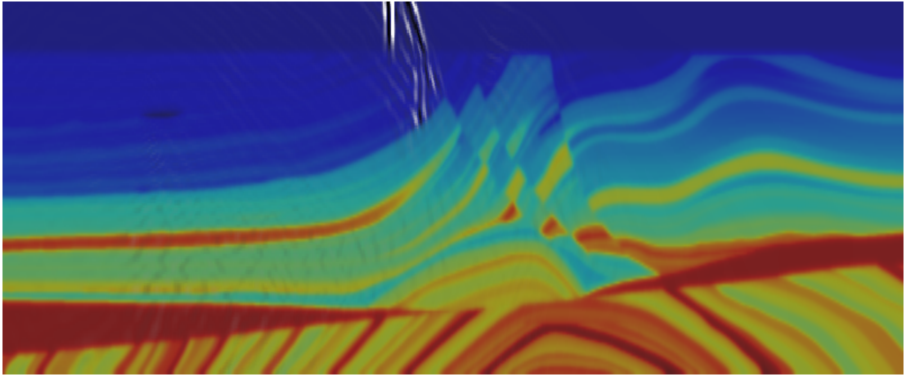
Modeling results



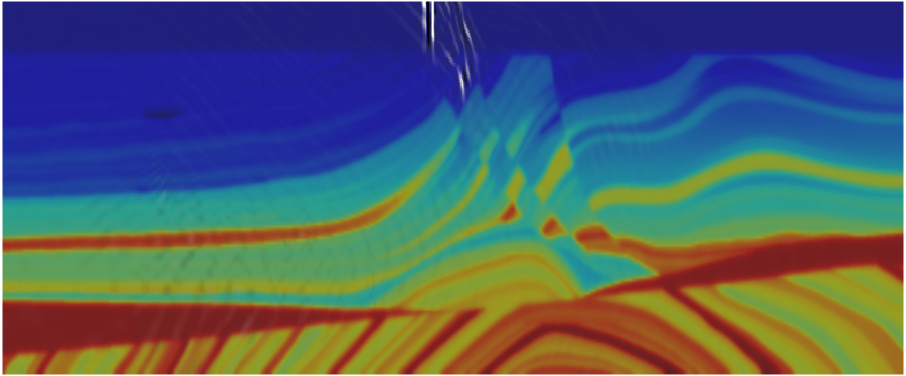
Modeling results



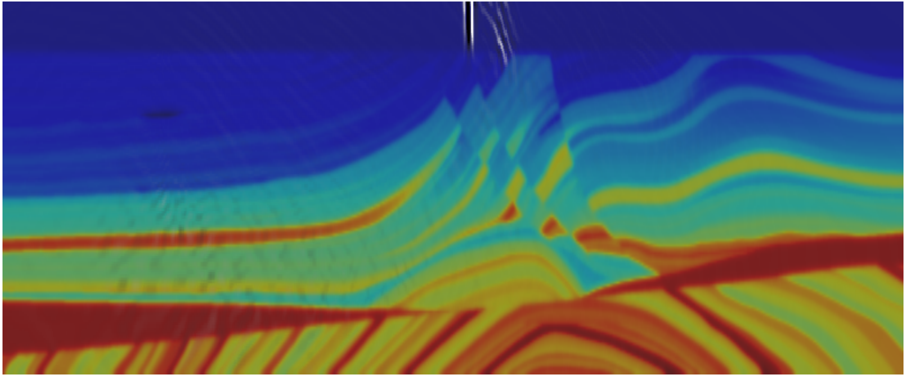
Modeling results



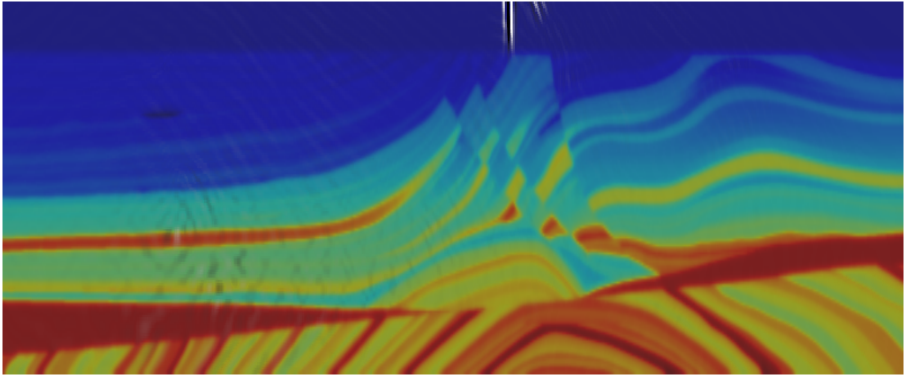
Modeling results



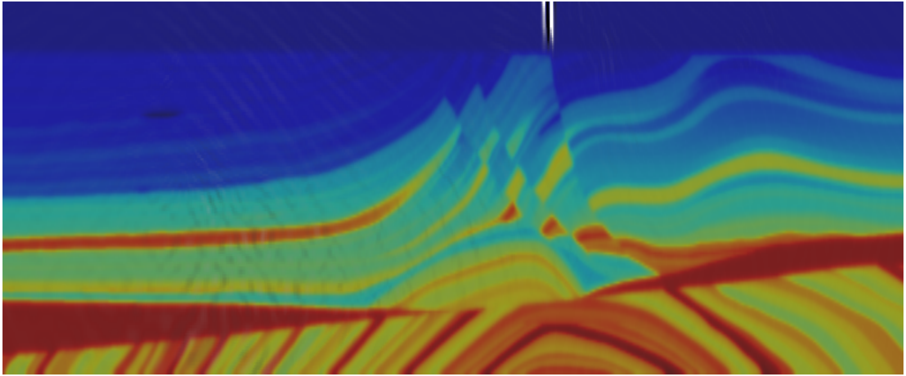
Modeling results



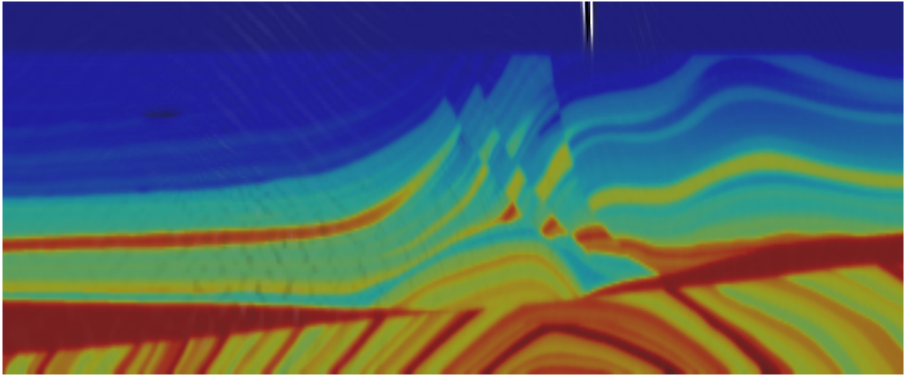
Modeling results



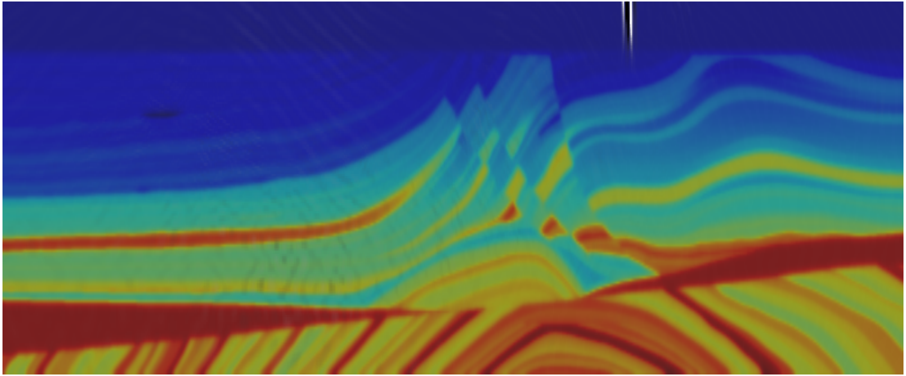
Modeling results



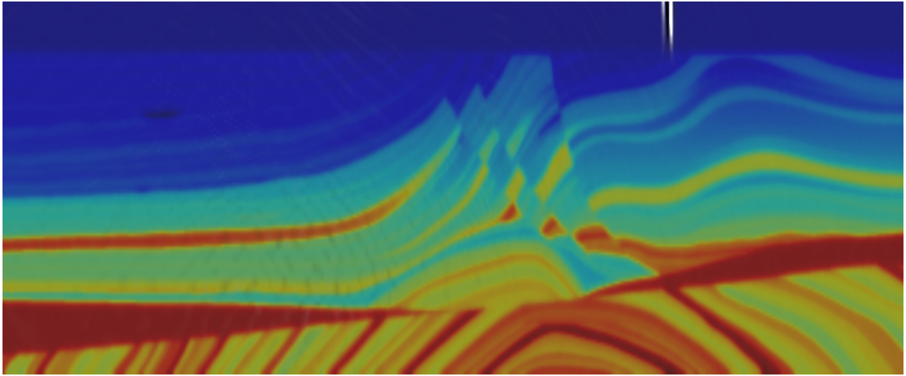
Modeling results



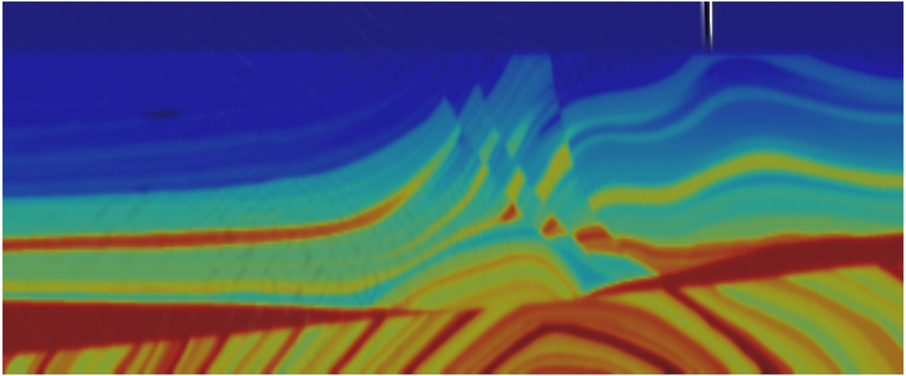
Modeling results



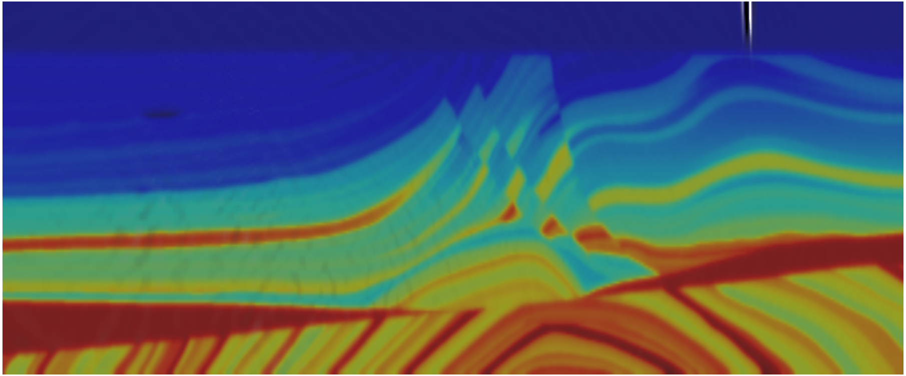
Modeling results



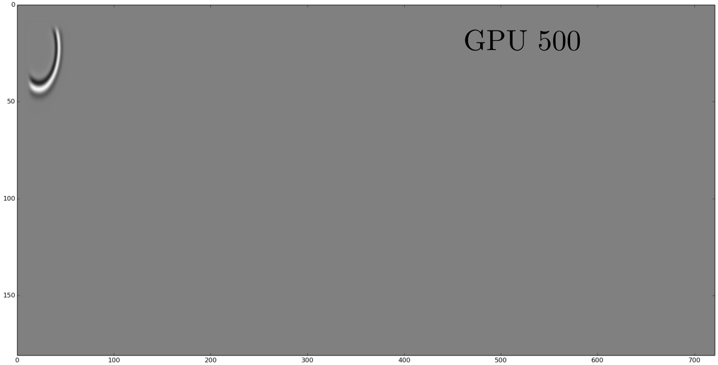
Modeling results



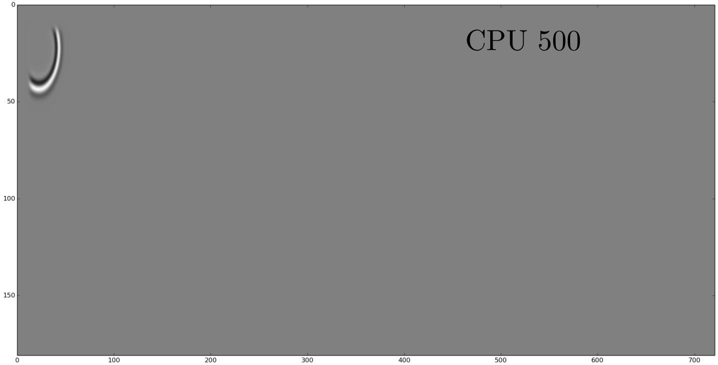
Modeling results



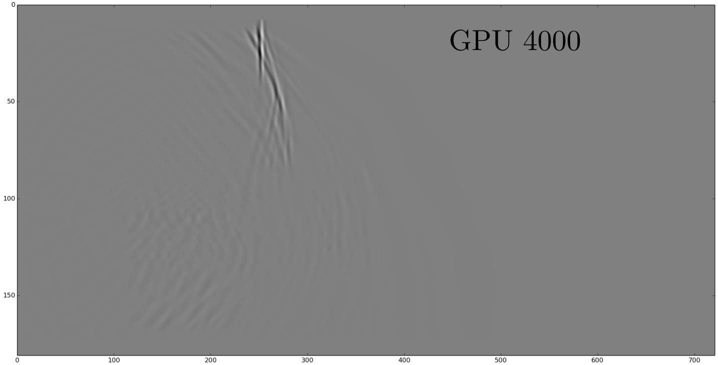
Modeling results



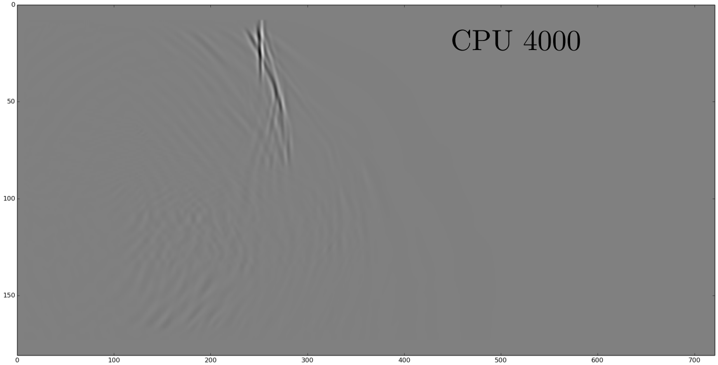
Modeling results



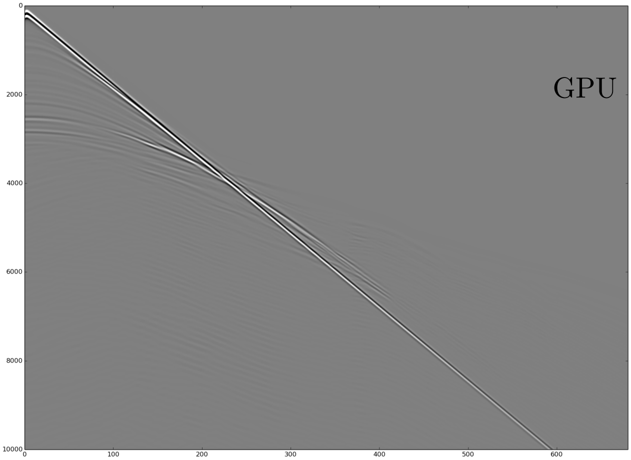
Modeling results



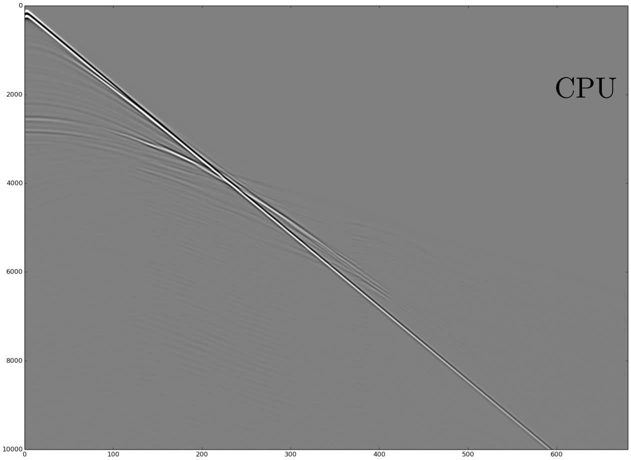
Modeling results



Modeling results



Modeling results



Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]

Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]
GPU	Elastic	627x317x411	5 676 s
GPU	Visco-elastic	627x317x411	99 556 s

Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]
GPU	Elastic	627x317x411	5 676 s
GPU	Visco-elastic	627x317x411	99 556 s
GPU	Elastic	450x450x450	6 348 s[1]

Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]
GPU	Elastic	627x317x411	5 676 s
GPU	Visco-elastic	627x317x411	99 556 s
GPU	Elastic	450x450x450	6 348 s[1]
GPU	Elastic	460x460x460	7 128 s[1]

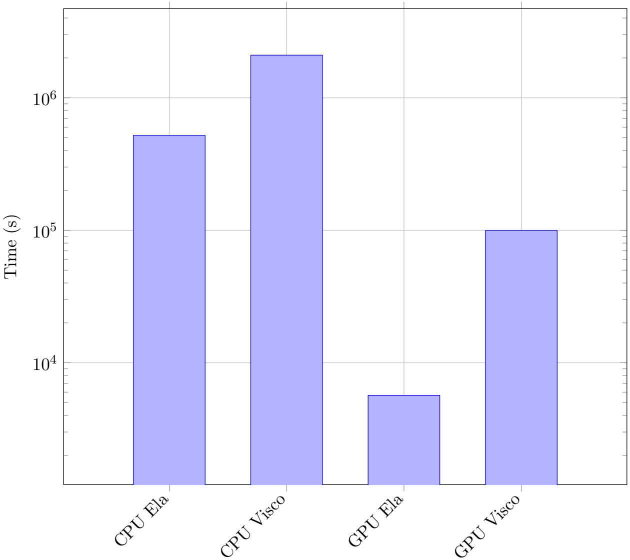
Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]
GPU	Elastic	627x317x411	5 676 s
GPU	Visco-elastic	627x317x411	99 556 s
GPU	Elastic	450x450x450	6 348 s[1]
GPU	Elastic	460x460x460	7 128 s[1]
GPU	Elastic	470x470x470	43 320 s[1]

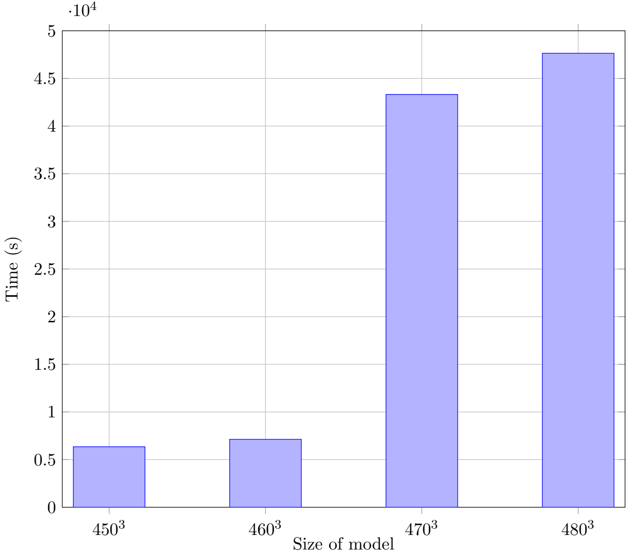
Modeling runtimes

Hardware	Physics	Model size	Runtime
CPU	Acoustic	627x317x411	157 200 s[1]
CPU	Elastic	627x317x411	519 600 s[1]
CPU	Visco-elastic	627x317x411	2 098 800 s[1]
GPU	Elastic	627x317x411	5 676 s
GPU	Visco-elastic	627x317x411	99 556 s
GPU	Elastic	450x450x450	6 348 s[1]
GPU	Elastic	460x460x460	7 128 s[1]
GPU	Elastic	470x470x470	43 320 s[1]
GPU	Elastic	480x480x480	47 640 s[1]

Modeling runtimes

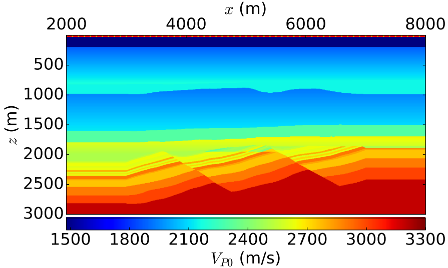


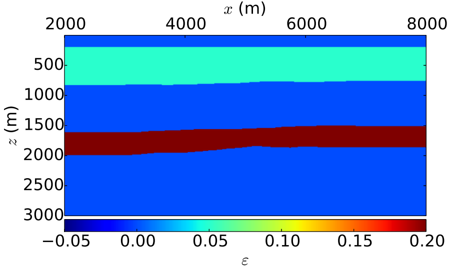
Modeling runtimes



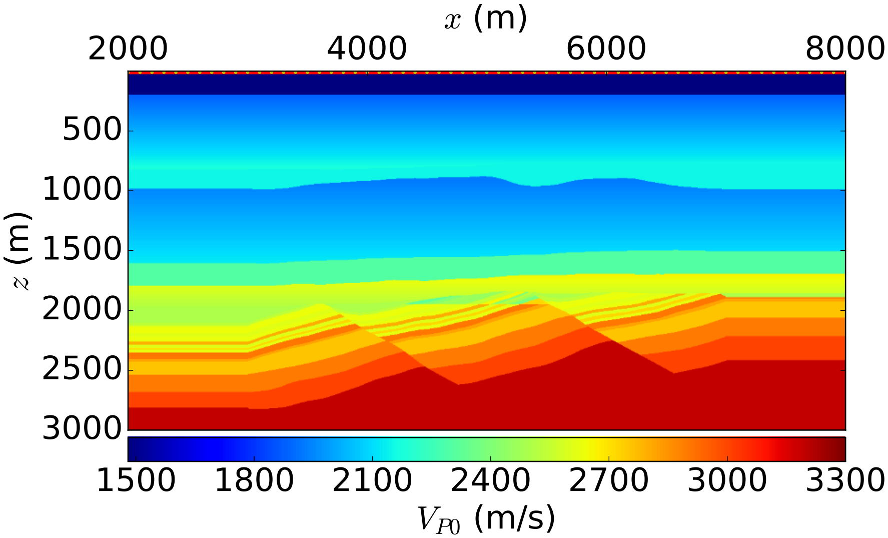
FWI model

- Synthetic model representative of the Gullfaks field
- 10 km long, 3 km deep
- 2001 × 600 grid points
- Total of 101 shots and 2001 receivers
- 3.3 second recording, 5500 time steps
- Source: 15 Hz Ricker wavelet bandpass filtered to 0-7 Hz, 0-10 Hz, and unfiltered.
- Receivers: Pressure
- $\sim$ 50 GB RAM per source



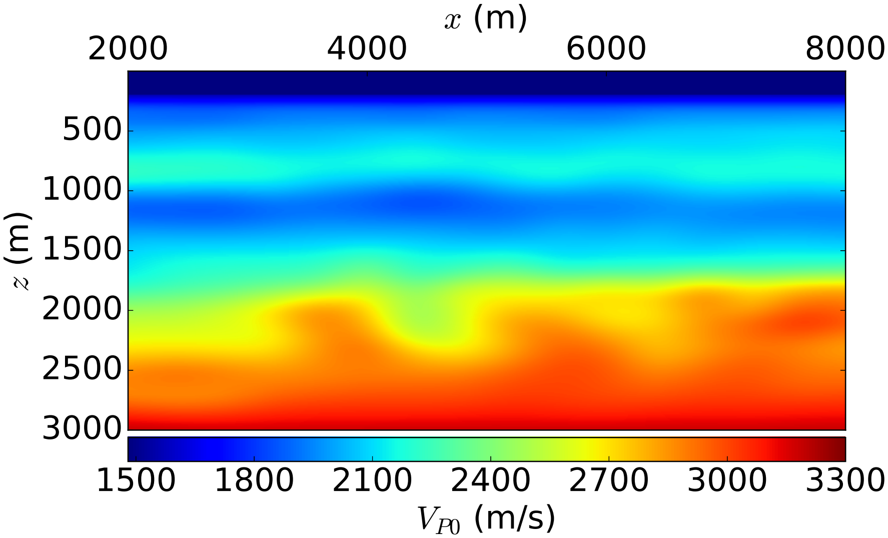


True model

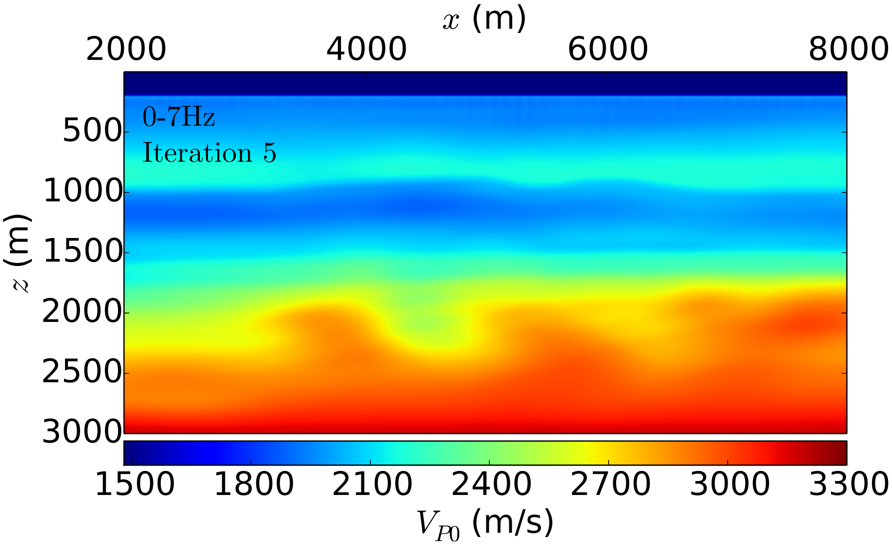


x (m)
2000
4000
6000
8000
500
1000
1500
2000
2500
3000
z (m)
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

Starting model

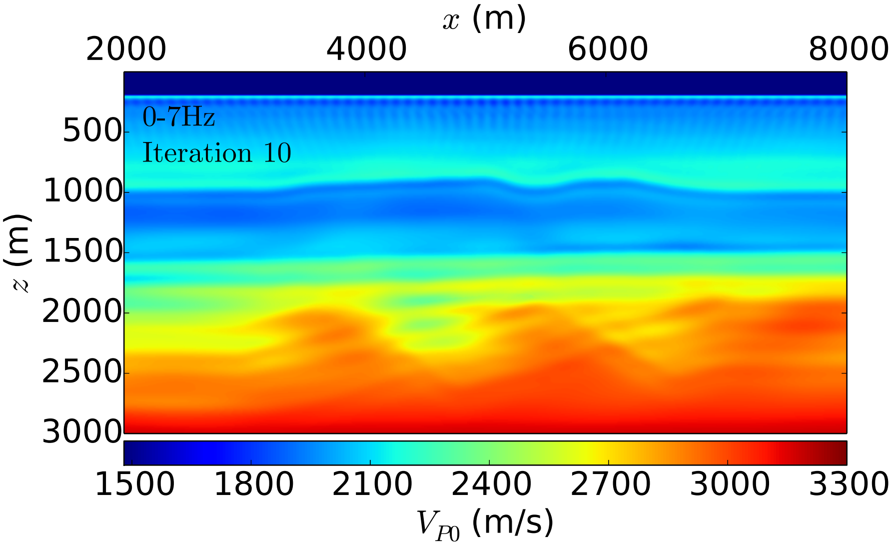


FWI results

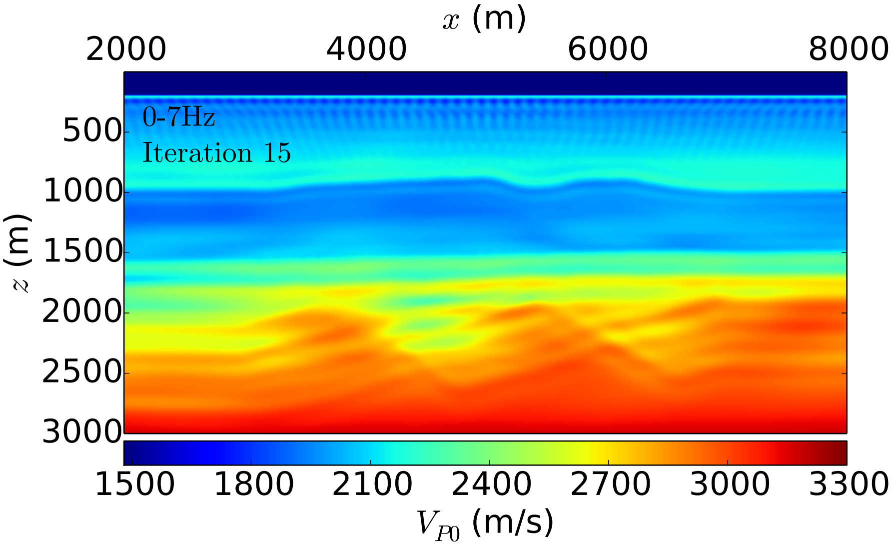


x (m)
2000
4000
6000
8000
0-7Hz
Iteration 5
500
1000
1500
2000
2500
3000
z (m)
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI results

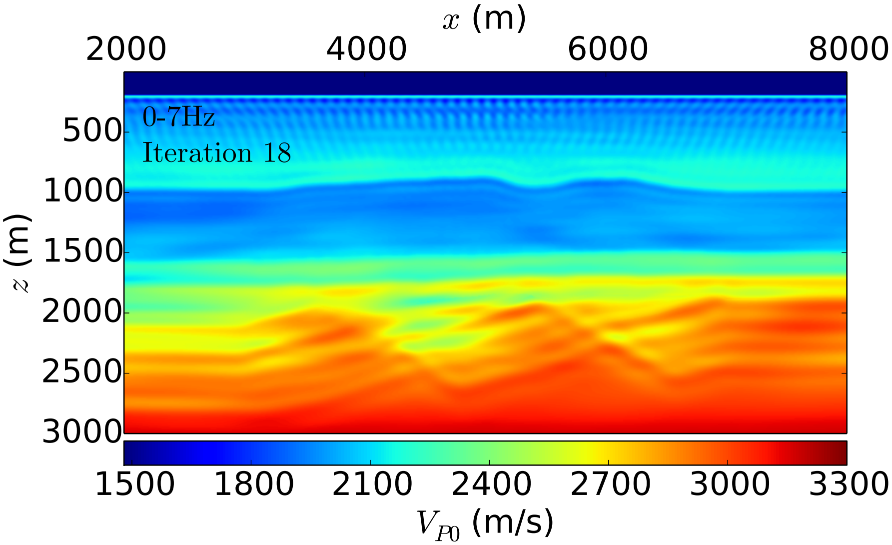

x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
0-7Hz
Iteration 10
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI results



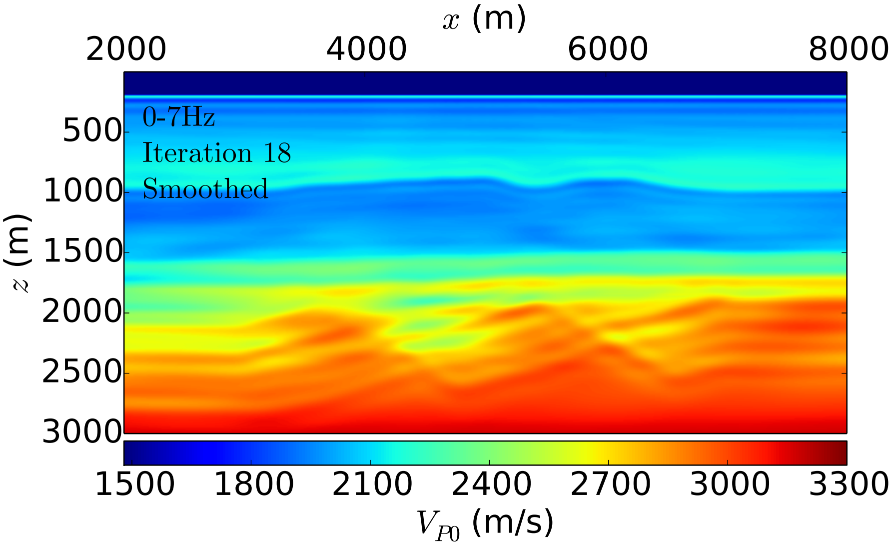
x (m)
2000
4000
6000
8000
500
1000
1500
2000
2500
3000
z (m)
0-7Hz
Iteration 15
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI results

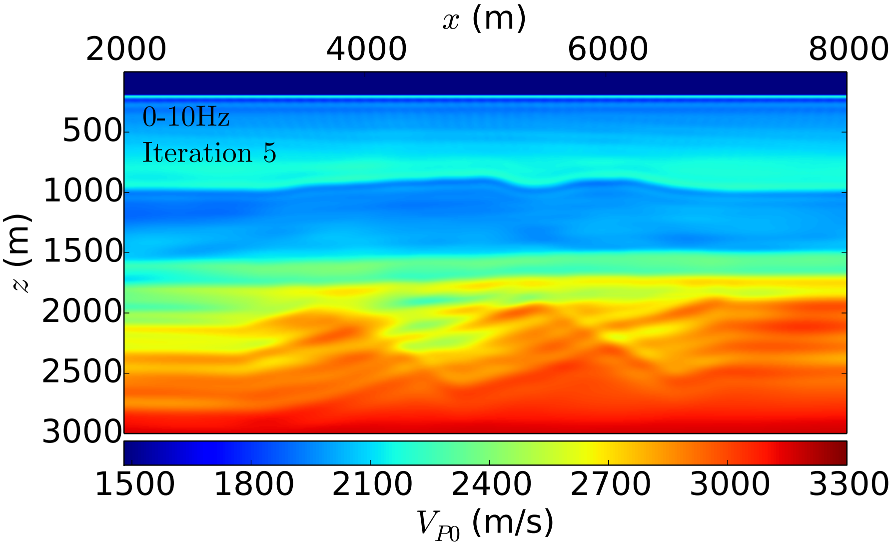


x (m)
2000
4000
6000
8000
500
1000
1500
2000
2500
3000
z (m)
0-7Hz
Iteration 18
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

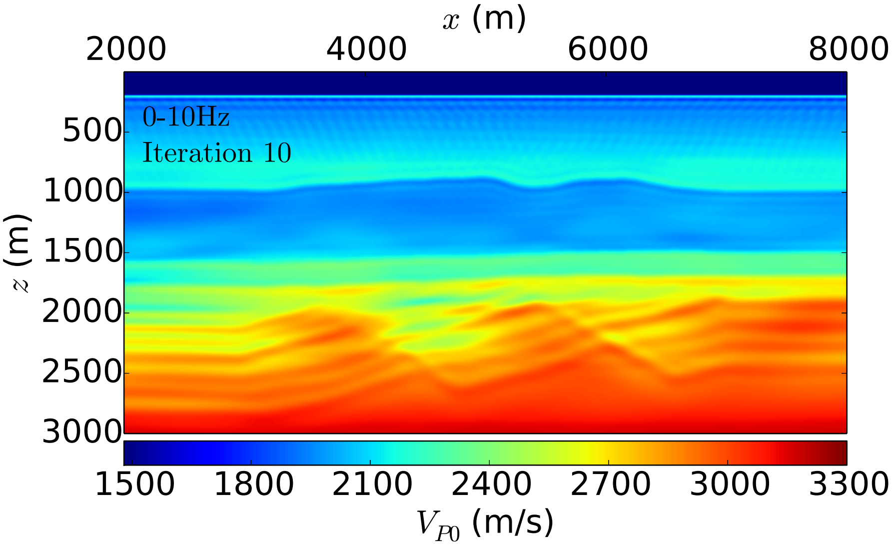
FWI results


x (m)
2000
4000
6000
8000
0-7Hz
Iteration 18
Smoothed
z (m)
500
1000
1500
2000
2500
3000
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

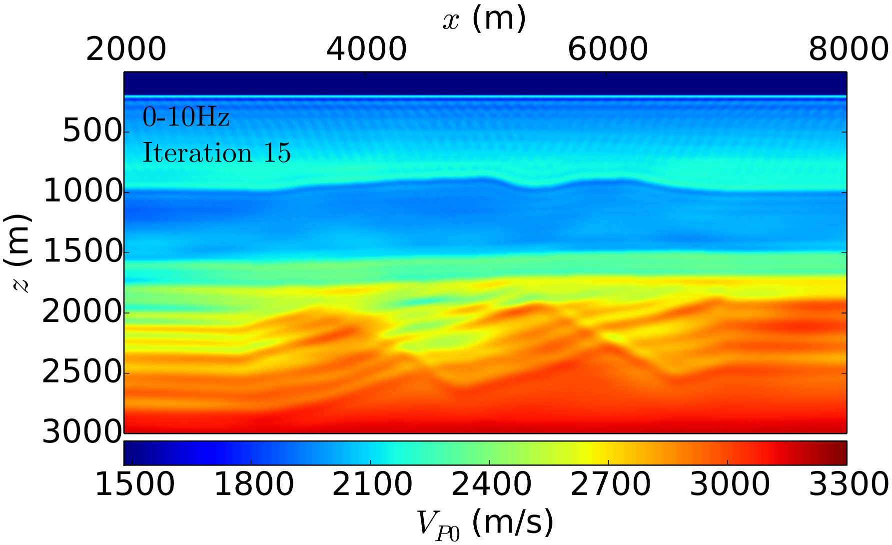
FWI results


x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
0-10Hz
Iteration 5
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI results

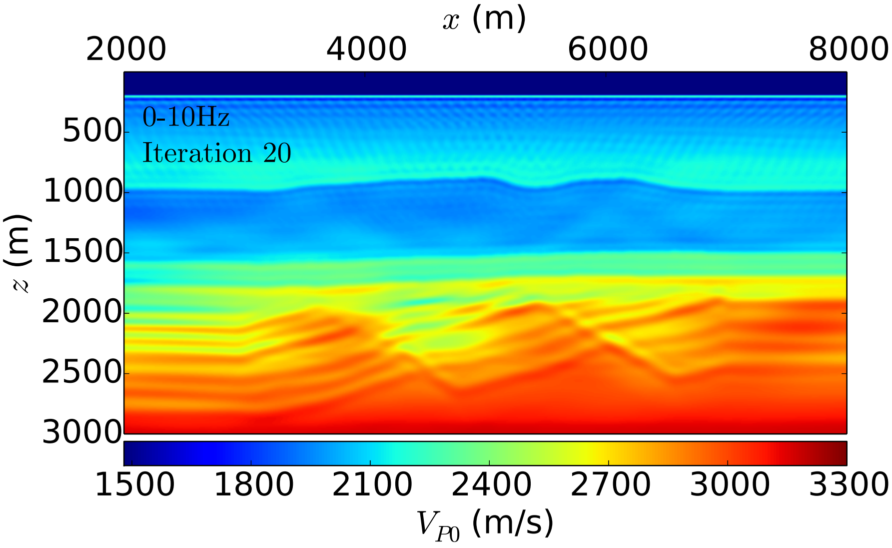

x (m)
2000
4000
6000
8000
0-10Hz
Iteration 10
z (m)
500
1000
1500
2000
2500
3000
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI results

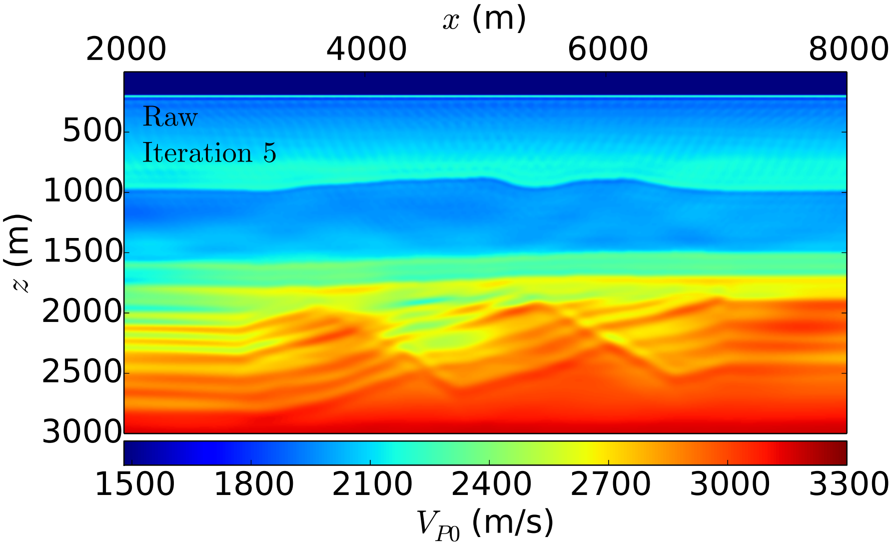


x (m)
2000
4000
6000
8000
500
1000
1500
2000
2500
3000
z (m)
0-10Hz
Iteration 15
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

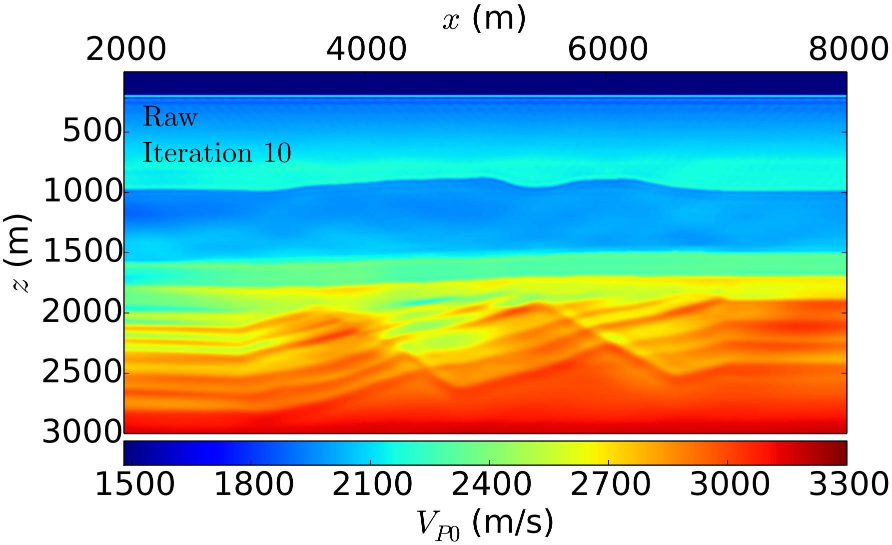
FWI results


x (m)
2000
4000
6000
8000
0-10Hz
Iteration 20
z (m)
500
1000
1500
2000
2500
3000
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

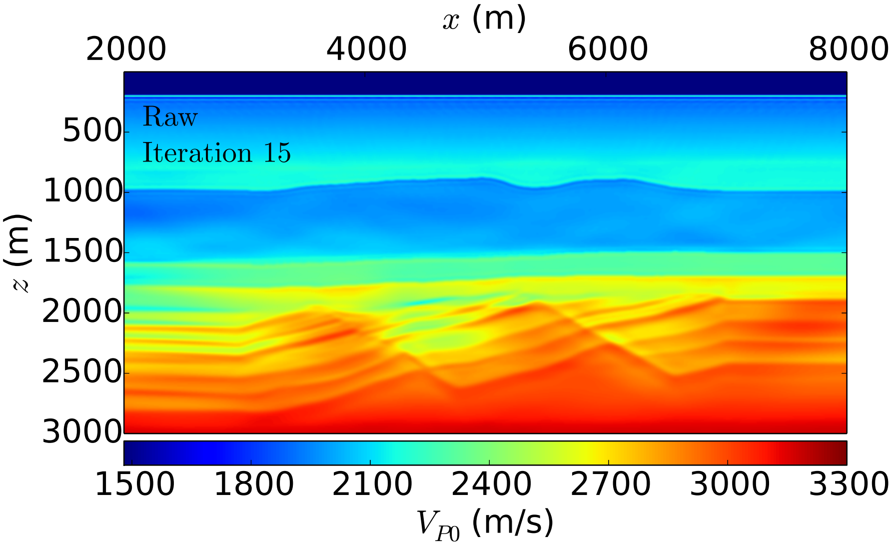
FWI results


x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
Raw
Iteration 5
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

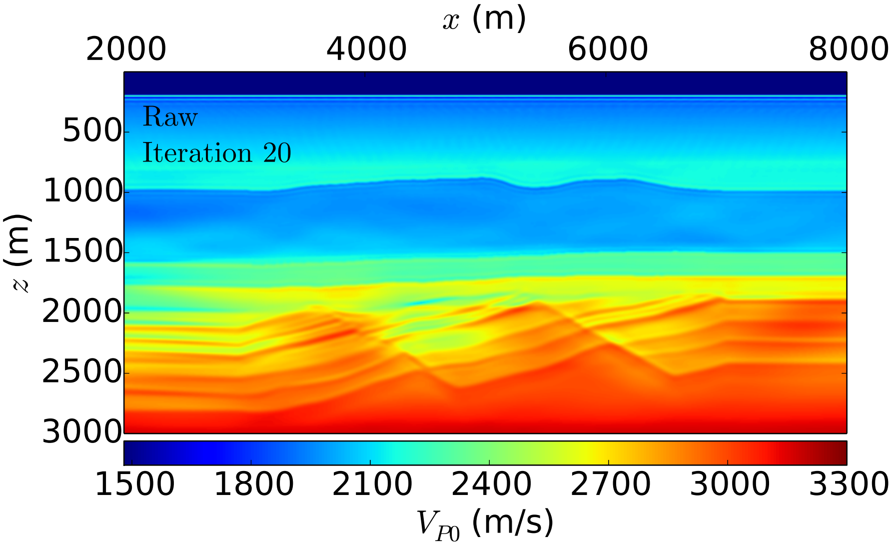
FWI results


x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
Raw
Iteration 10
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

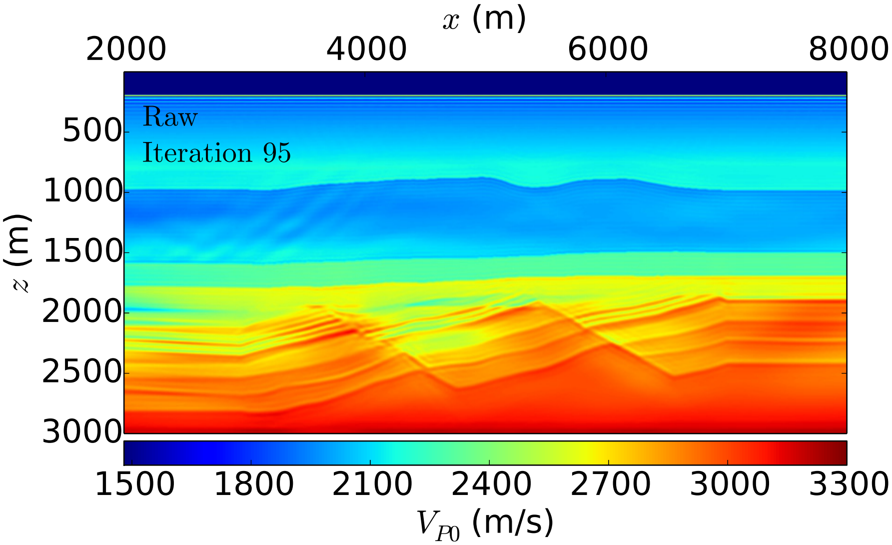
FWI results


x (m)
2000
4000
6000
8000
Raw
Iteration 15
500
1000
1500
2000
2500
3000
z (m)
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

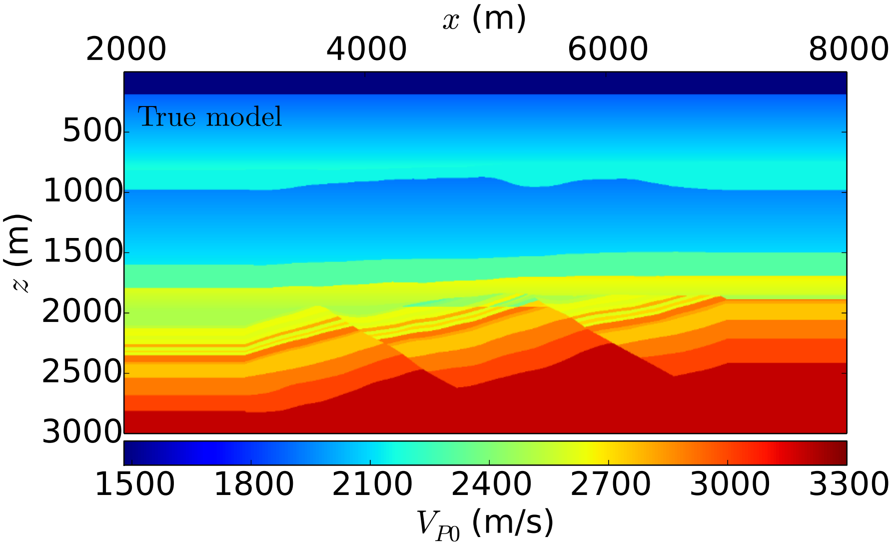
FWI results


x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
Raw
Iteration 20
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

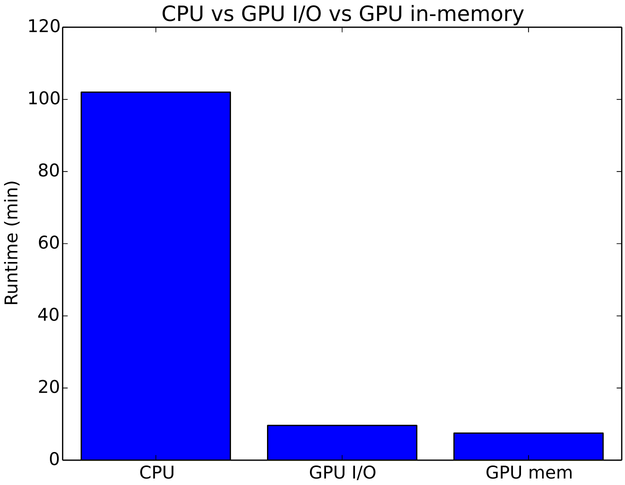
FWI results


x (m)
2000
4000
6000
8000
z (m)
500
1000
1500
2000
2500
3000
Raw
Iteration 95
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

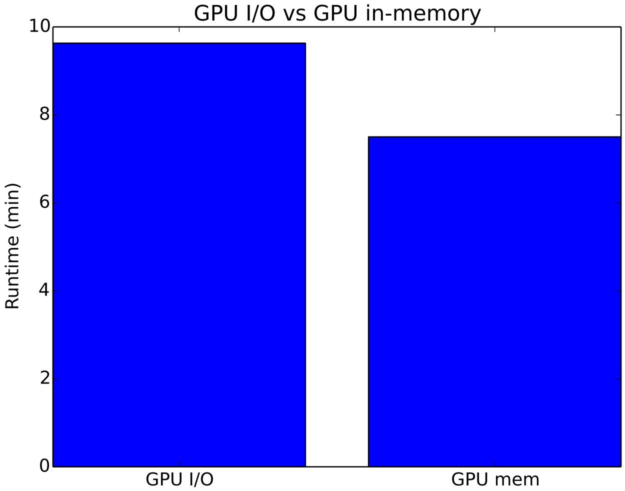
FWI results


x (m)
2000
4000
6000
8000
True model
500
1000
1500
2000
2500
3000
z (m)
1500
1800
2100
2400
2700
3000
3300
V_{P0} (m/s)

FWI runtimes



FWI runtimes



Conclusions

- Achieved approximately 100 times speedup of single source 3D elastic FD modeling.
- Achieved approximately 12 times speedup of FWI.

Conclusions

- Achieved approximately 100 times speedup of single source 3D elastic FD modeling.
- Achieved approximately 12 times speedup of FWI.
- Writing fast GPU code is relatively easy.

Acknowledgments

We thank the ROSE consortium and their sponsors for support. The authors would like to thank NTNU and Aker BP ASA for making the code available through the Codeshare project.