Industry-scale finite difference wave modelling on a single GPU using the out-of-core technique

Jon Marius Venstad

Norwegian University of Science and Technology (NTNU)
Department of Petroleum Engineering & Applied Geophysics
E-mail: venstad@gmail.com

The ROSE Meeting, 2015

What is a Graphical Processing Unit?

What is a Graphical Processing Unit?

- Is specialised consumer hardware.

What is a Graphical Processing Unit?

- Is specialised consumer hardware.
- Has $\times 100$ the arithmetic capabilites of one CPU core.

What is a Graphical Processing Unit?

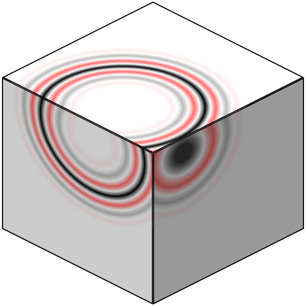
- Is specialised consumer hardware.
- Has $\times 100$ the arithmetic capabilites of one CPU core.
- Requires massively parallel, repeated computations.

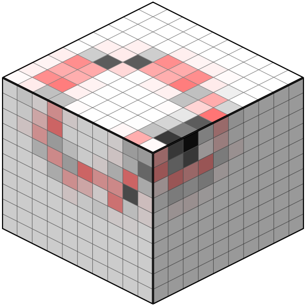
What is a Graphical Processing Unit?

- Is specialised consumer hardware.
- Has $\times 100$ the arithmetic capabilites of one CPU core.
- Requires massively parallel, repeated computations.
- Has only limited storage.

What is Finite Difference Modelling?

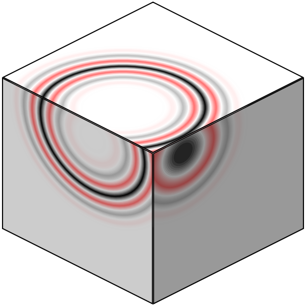
- Used to model physical phenomena, e.g. wave propagation.

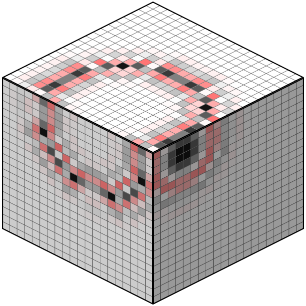




What is Finite Difference Modelling?

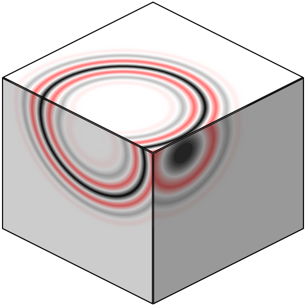
- Used to model physical phenomena, e.g. wave propagation.
- Finer models and more calculations allow for higher accuracy.

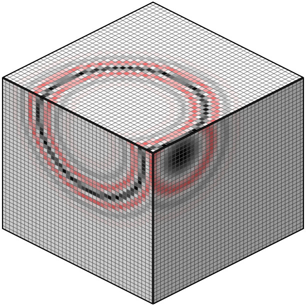




What is Finite Difference Modelling?

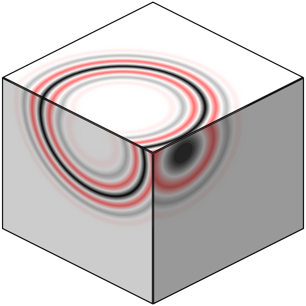
- Used to model physical phenomena, e.g. wave propagation.
- Finer models and more calculations allow for higher accuracy.
- Significant memory and computational requirements for 3D.

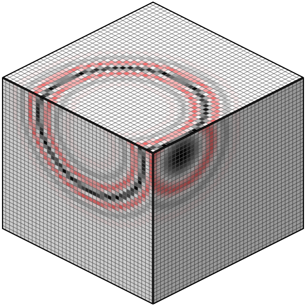




What is Finite Difference Modelling?

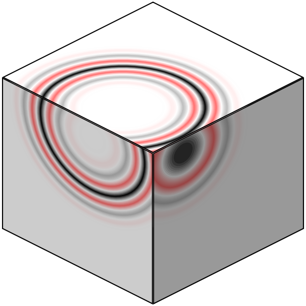
- Used to model physical phenomena, e.g. wave propagation.
- Finer models and more calculations allow for higher accuracy.
- Significant memory and computational requirements for 3D.
- Use Graphical Processing Units (GPUs) for computation.

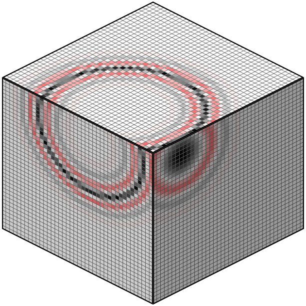




What is Finite Difference Modelling?

- Used to model physical phenomena, e.g. wave propagation.
- Finer models and more calculations allow for higher accuracy.
- Significant memory and computational requirements for 3D.
- Use Graphical Processing Units (GPUs) for computation.
- Circumvent the limitations posed by the size of the GPU.





How to: Finite Difference Modelling

Model a given differential equation, e.g.:

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho}\left(\frac{\partial \tau_{ij}}{\partial j}\right) \tag{1}$$

$$\frac{\partial \tau_{ij}}{\partial t} = \delta_{ij}\lambda\frac{\partial v_k}{\partial k} + \mu\left(\frac{\partial v_i}{\partial j} + \frac{\partial v_j}{\partial i}\right) \tag{2}$$

How to: Finite Difference Modelling

Model a given differential equation, e.g.:

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho}\left(\frac{\partial \tau_{ij}}{\partial j}\right) \tag{1}$$

$$\frac{\partial \tau_{ij}}{\partial t} = \delta_{ij}\lambda\frac{\partial v_k}{\partial k} + \mu\left(\frac{\partial v_i}{\partial j} + \frac{\partial v_j}{\partial i}\right) \tag{2}$$

1. Discretise each parameter and variable onto a 3D cube.

How to: Finite Difference Modelling

Model a given differential equation, e.g.:

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho}\left(\frac{\partial \tau_{ij}}{\partial j}\right) \tag{1}$$

$$\frac{\partial \tau_{ij}}{\partial t} = \delta_{ij}\lambda\frac{\partial v_k}{\partial k} + \mu\left(\frac{\partial v_i}{\partial j} + \frac{\partial v_j}{\partial i}\right) \tag{2}$$

1. Discretise each parameter and variable onto a 3D cube.
2. Approximate derivatives by weighted sums.

How to: Finite Difference Modelling

Model a given differential equation, e.g.:

$$\frac{\partial v_i}{\partial t} = \frac{1}{\rho}\left(\frac{\partial \tau_{ij}}{\partial j}\right) \tag{1}$$

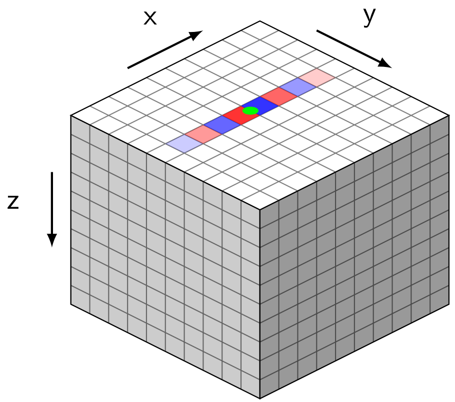
$$\frac{\partial \tau_{ij}}{\partial t} = \delta_{ij}\lambda\frac{\partial v_k}{\partial k} + \mu\left(\frac{\partial v_i}{\partial j} + \frac{\partial v_j}{\partial i}\right) \tag{2}$$

1. Discretise each parameter and variable onto a 3D cube.
2. Approximate derivatives by weighted sums.
3. Update each variable across a small Δt, many times.

For example ...

$$\frac{\partial}{\partial x} u_{i+\frac{1}{2},j,k} \approx$$

$$\begin{array}{rl}
0.0038 & u_{i-3,j,k} \\
-0.0211 & u_{i-2,j,k} \\
+0.1049 & u_{i-1,j,k} \\
-1.2327 & u_{i,j,k} \\
+1.2327 & u_{i+1,j,k} \\
-0.1049 & u_{i+2,j,k} \\
+0.0211 & u_{i+3,j,k} \\
-0.0038 & u_{i+4,j,k}
\end{array}$$



$$\tau_{xy}^{n+\frac{1}{2}} = \tau_{xy}^{n-\frac{1}{2}} + \mu \Delta t \left(\frac{\partial}{\partial x} v_y^n + \frac{\partial}{\partial y} v_x^n \right)$$

Industry scale Finite Difference Modelling?

Model a seismic shot:

- Wavelengths down to 10m.
- 4m$\times$4m$\times$4m cells.
- $1000 \times 500 \times 2000$ grid cells.

Industry scale Finite Difference Modelling?

Model a seismic shot:

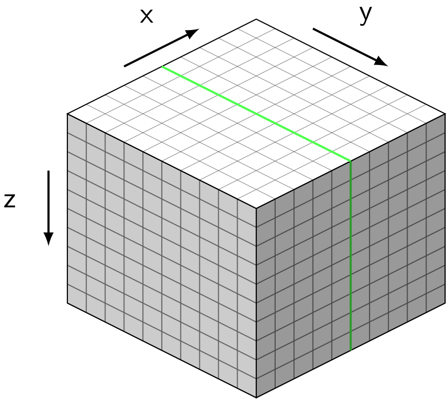
- Wavelengths down to 10m.
- 4m$\times$4m$\times$4m cells.
- $1000 \times 500 \times 2000$ grid cells.
- 48GB of data.

Industry scale Finite Difference Modelling?

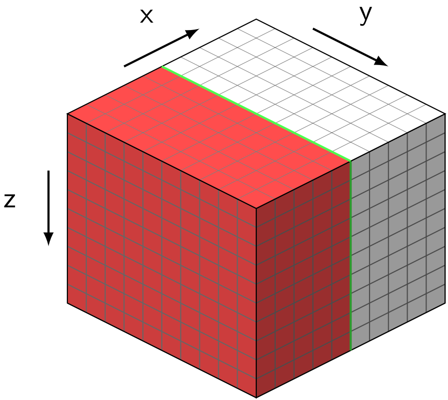
Model a seismic shot:

- Wavelengths down to 10m.
- 4m$\times$4m$\times$4m cells.
- $1000 \times 500 \times 2000$ grid cells.
- 48GB of data.
- A consumer GPU typically fits 4GB-8GB of data.

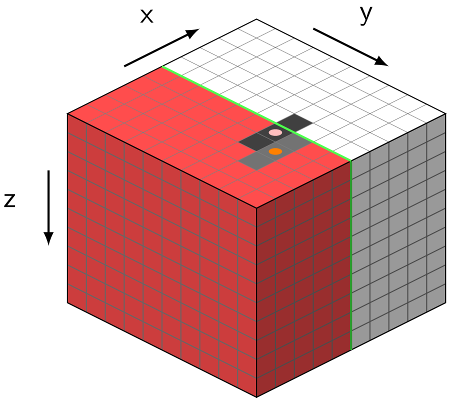
The Memory Barrier – Fitting data onto the GPU



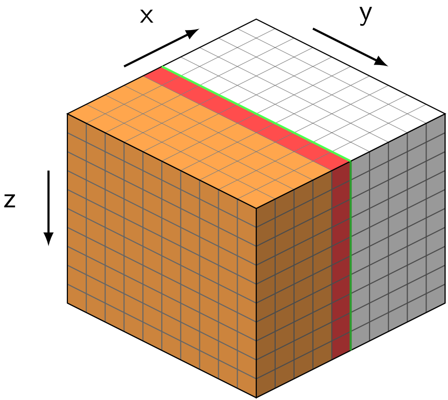
The Memory Barrier – Fitting data onto the GPU



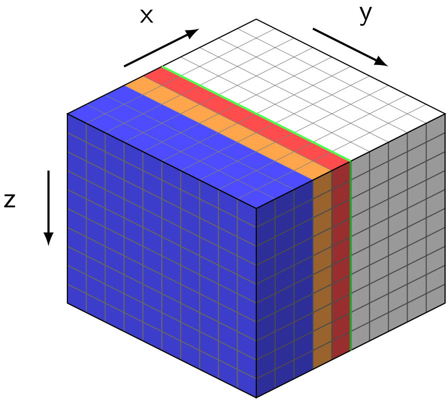
The Memory Barrier – Fitting data onto the GPU



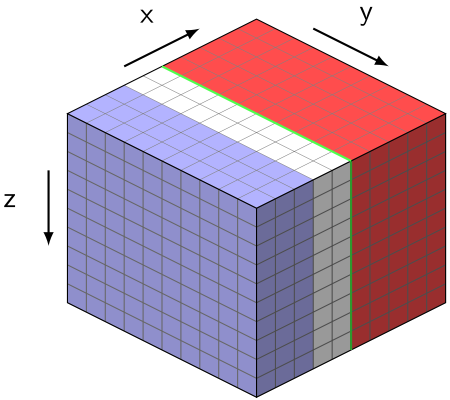
The Memory Barrier – Fitting data onto the GPU



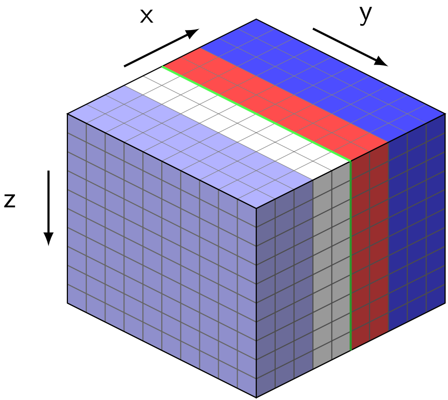
The Memory Barrier – Fitting data onto the GPU



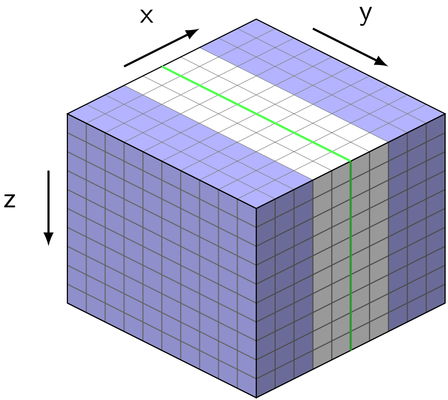
The Memory Barrier – Fitting data onto the GPU



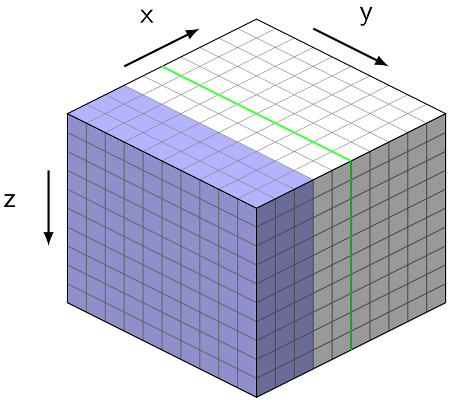
The Memory Barrier – Fitting data onto the GPU



The Memory Barrier – Fitting data onto the GPU

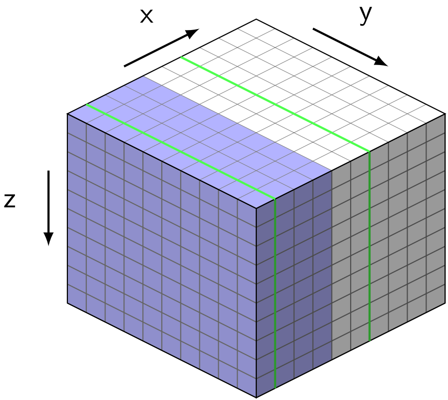


The Memory Barrier – Fitting data onto the GPU



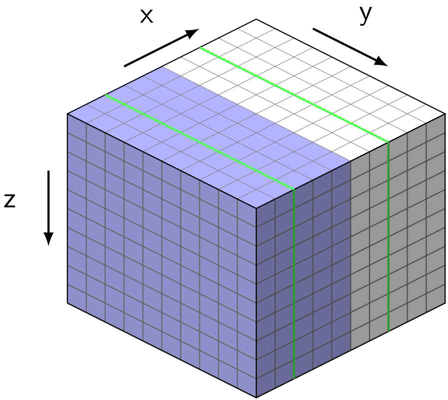
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



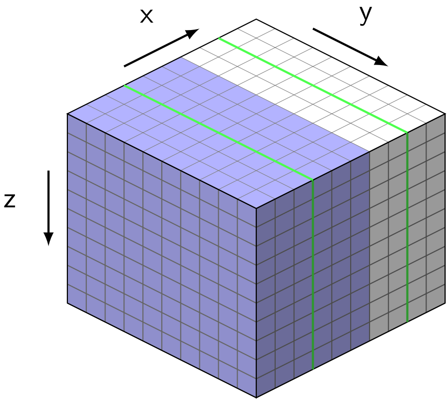
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



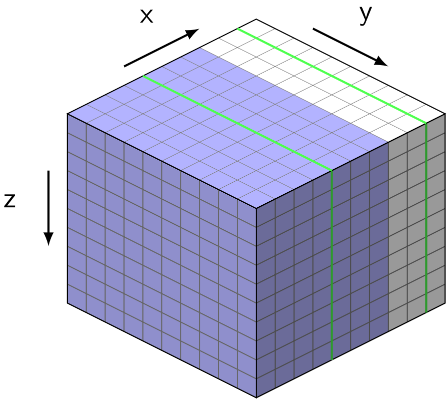
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



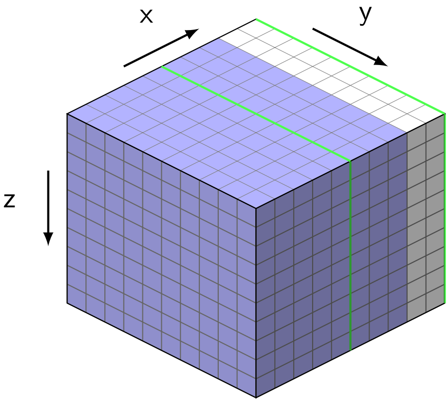
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



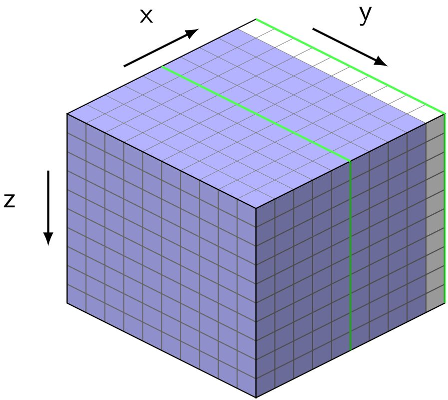
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



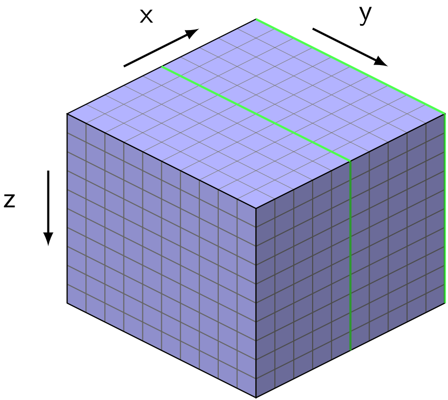
- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



- Use overlapping, sliding model blocks.

The Memory Barrier – Fitting data onto the GPU



- Use overlapping, sliding model blocks.

The Memory Barrier – Data Transfer Slowdown

	Theoretical		Measured	
Work	Speed	Time	Speed	Time
48Gb	16Gb/s	3s	5.3Gb/s	9.2s
$5 \cdot 10^{11}$flop	$4 \cdot 10^{12}$flop/s	0.1s	($4 \cdot 10^{11}$flop/s)	(1.3s)
180Gb	288Gb/s	0.6s	140Gb/s	1.3s

The Memory Barrier – Data Transfer Slowdown

	Theoretical		Measured	
Work	Speed	Time	Speed	Time
48Gb	16Gb/s	3s	5.3Gb/s	9.2s
$5 \cdot 10^{11}$flop	$4 \cdot 10^{12}$flop/s	0.1s	($4 \cdot 10^{11}$flop/s)	(1.3s)
180Gb	288Gb/s	0.6s	140Gb/s	1.3s

- Need computational time $\geq$ IO time to hide tranfsers.

The Memory Barrier – Data Transfer Slowdown

	Theoretical		Measured	
Work	Speed	Time	Speed	Time
48Gb	16Gb/s	3s	5.3Gb/s	9.2s
$5 \cdot 10^{11}$flop	$4 \cdot 10^{12}$flop/s	0.1s	$(4 \cdot 10^{11}$flop/s)	(1.3s)
180Gb	288Gb/s	0.6s	140Gb/s	1.3s

- Need computational time $\geq$ IO time to hide tranfsers.
- This is satisfied when we do 7 time steps per pass.

The Memory Barrier – Data Transfer Slowdown

	Theoretical		Measured	
Work	Speed	Time	Speed	Time
48Gb	16Gb/s	3s	5.3Gb/s	9.2s
$5 \cdot 10^{11}$flop	$4 \cdot 10^{12}$flop/s	0.1s	($4 \cdot 10^{11}$flop/s)	(1.3s)
180Gb	288Gb/s	0.6s	140Gb/s	1.3s

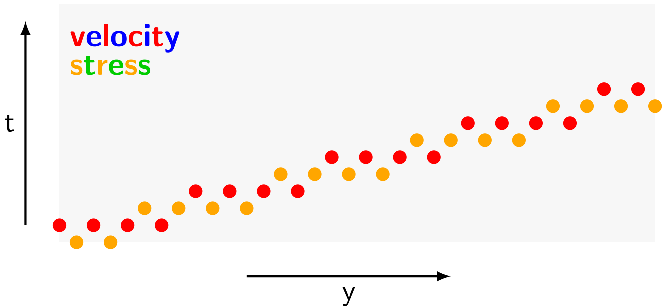
- Need computational time $\geq$ IO time to hide tranfsers.
- This is satisfied when we do 7 time steps per pass.
- $2L(Q+1)$ slices with differentiator length of $2L$ and Q steps.

The Memory Barrier – Data Transfer Slowdown

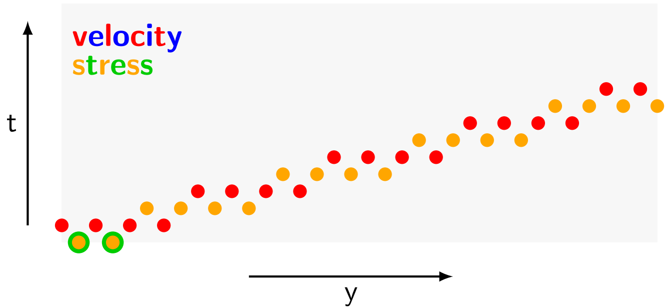
	Theoretical		Measured	
Work	Speed	Time	Speed	Time
48Gb	16Gb/s	3s	5.3Gb/s	9.2s
$5 \cdot 10^{11}$flop	$4 \cdot 10^{12}$flop/s	0.1s	($4 \cdot 10^{11}$flop/s)	(1.3s)
180Gb	288Gb/s	0.6s	140Gb/s	1.3s

- Need computational time $\geq$ IO time to hide tranfsers.
- This is satisfied when we do 7 time steps per pass.
- $2L(Q+1)$ slices with differentiator length of $2L$ and Q steps.
- $L = 8$ and $Q = 7$ gives $<$ 3GB for the reference model.

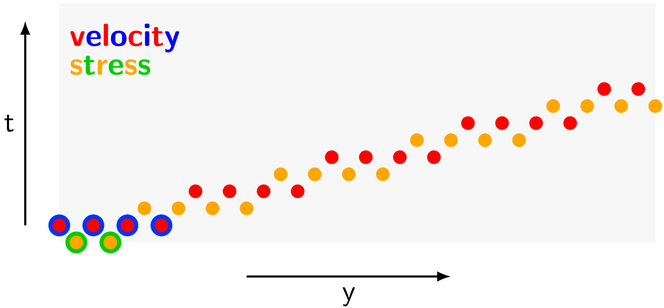
Breaking the Memory Barrier



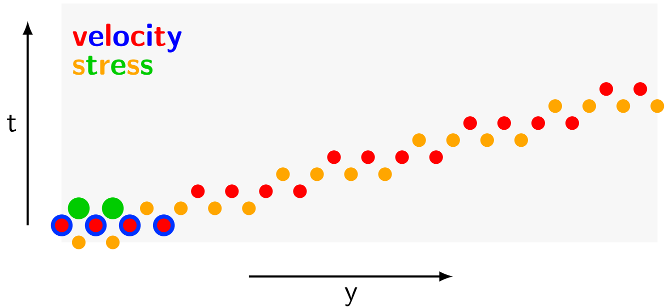
Breaking the Memory Barrier



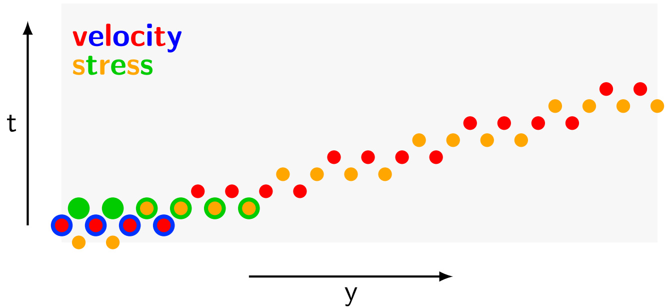
Breaking the Memory Barrier



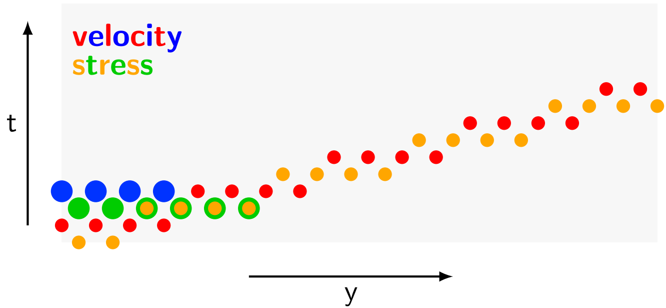
Breaking the Memory Barrier



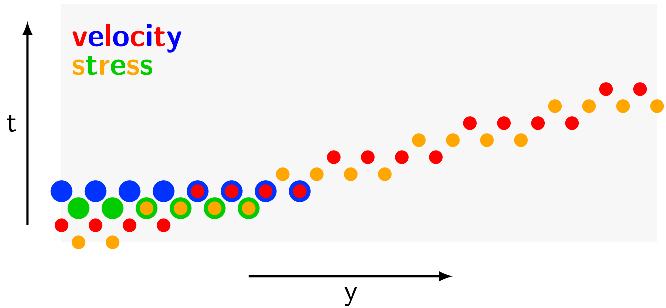
Breaking the Memory Barrier



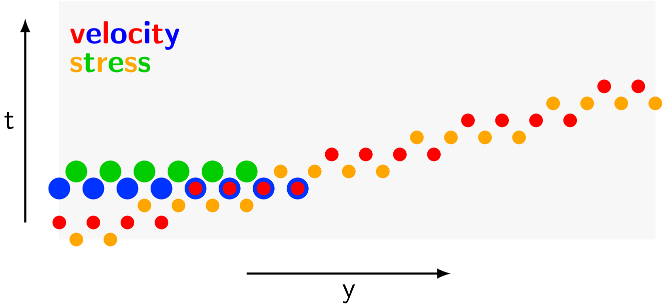
Breaking the Memory Barrier



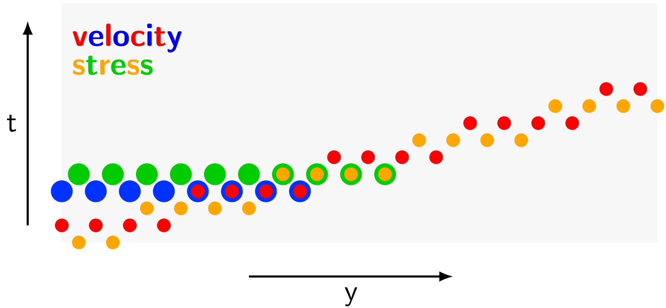
Breaking the Memory Barrier



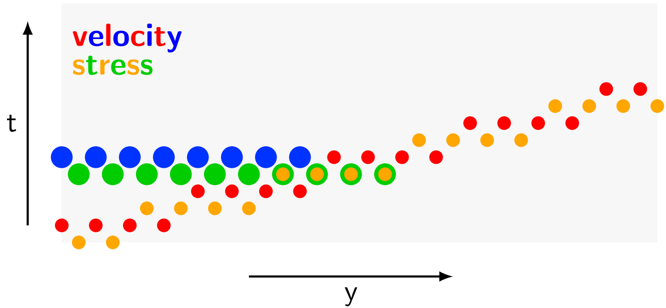
Breaking the Memory Barrier



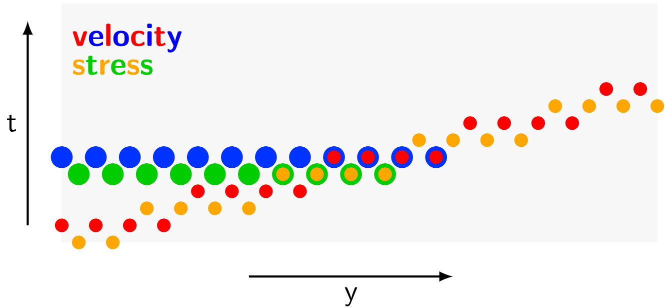
Breaking the Memory Barrier



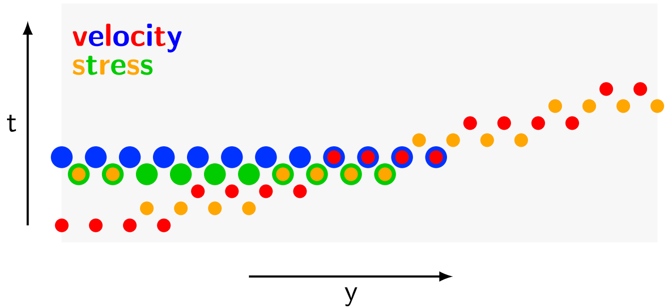
Breaking the Memory Barrier



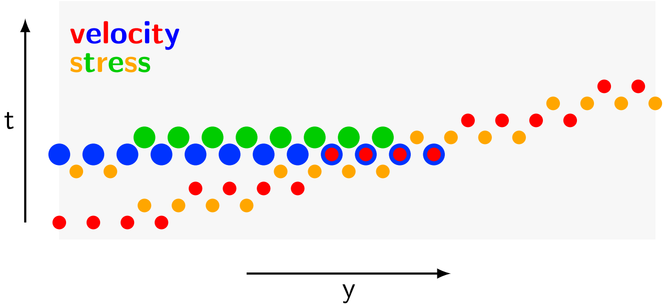
Breaking the Memory Barrier



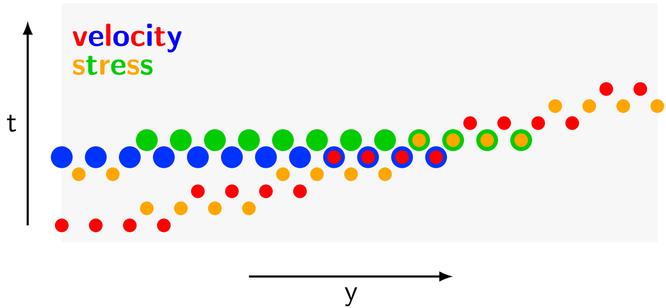
Breaking the Memory Barrier



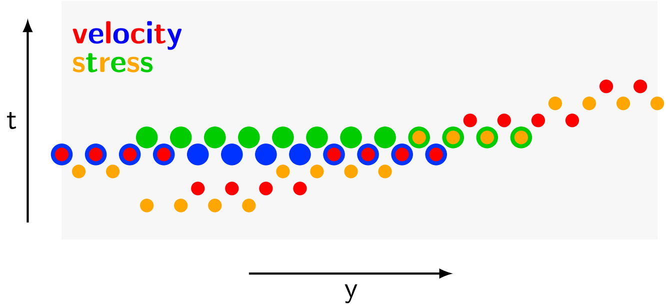
Breaking the Memory Barrier



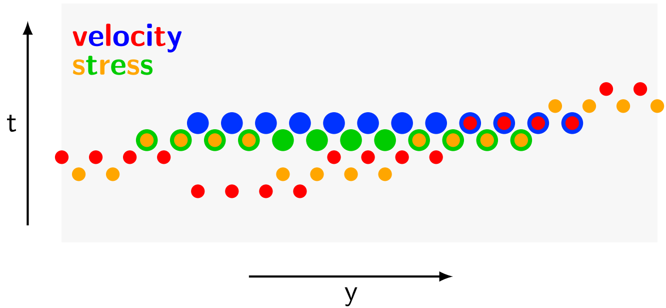
Breaking the Memory Barrier



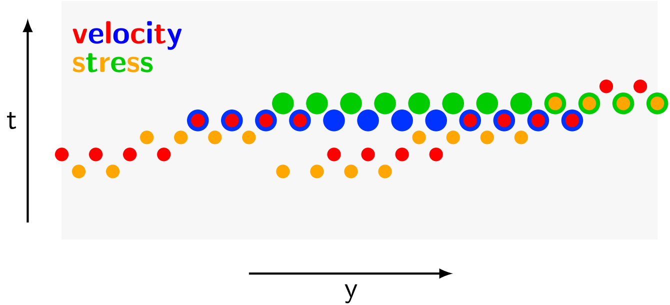
Breaking the Memory Barrier



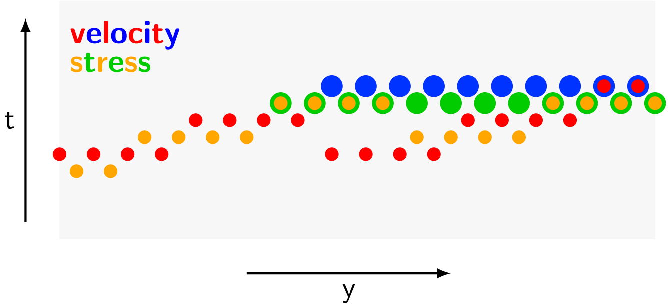
Breaking the Memory Barrier



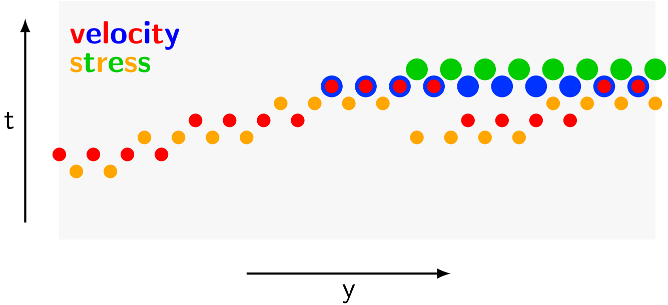
Breaking the Memory Barrier



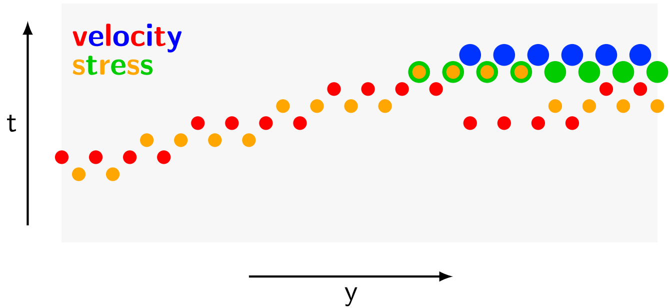
Breaking the Memory Barrier



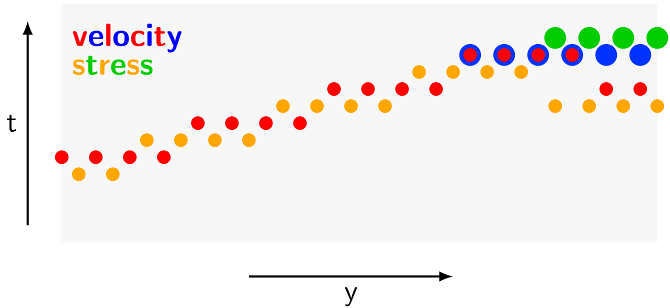
Breaking the Memory Barrier



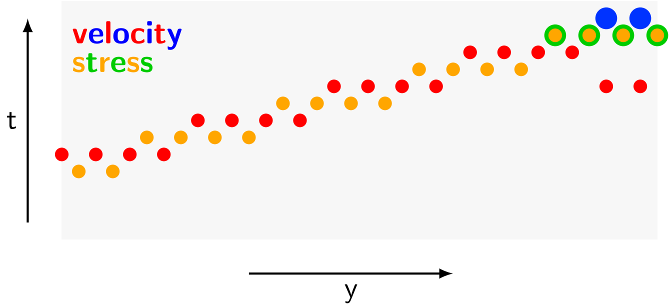
Breaking the Memory Barrier



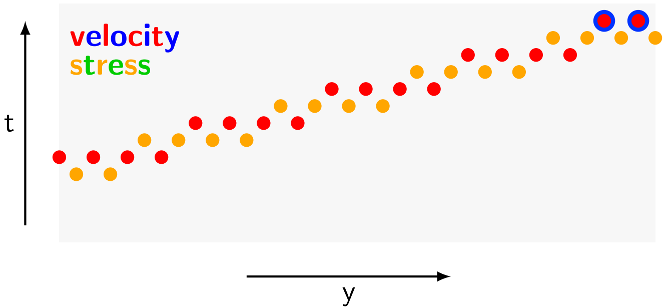
Breaking the Memory Barrier



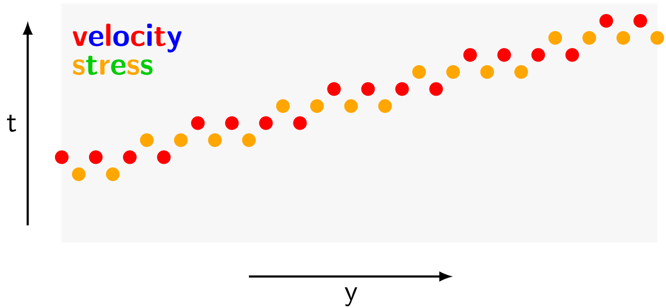
Breaking the Memory Barrier



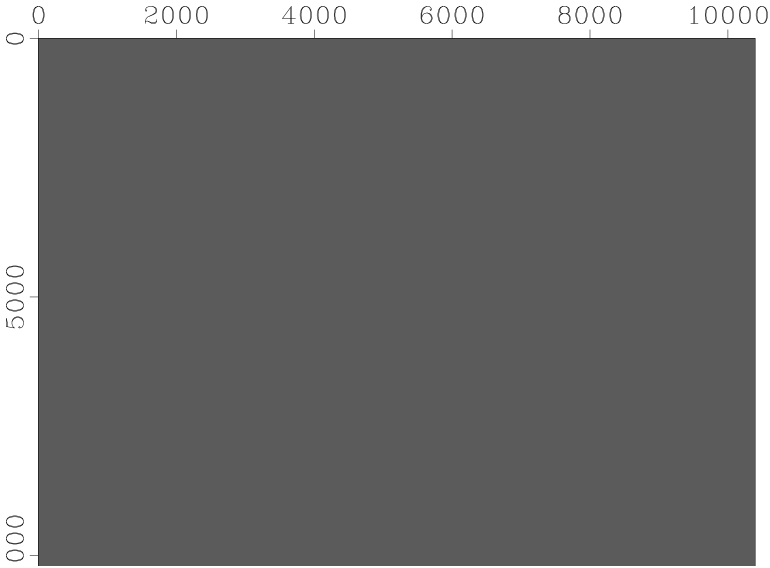
Breaking the Memory Barrier



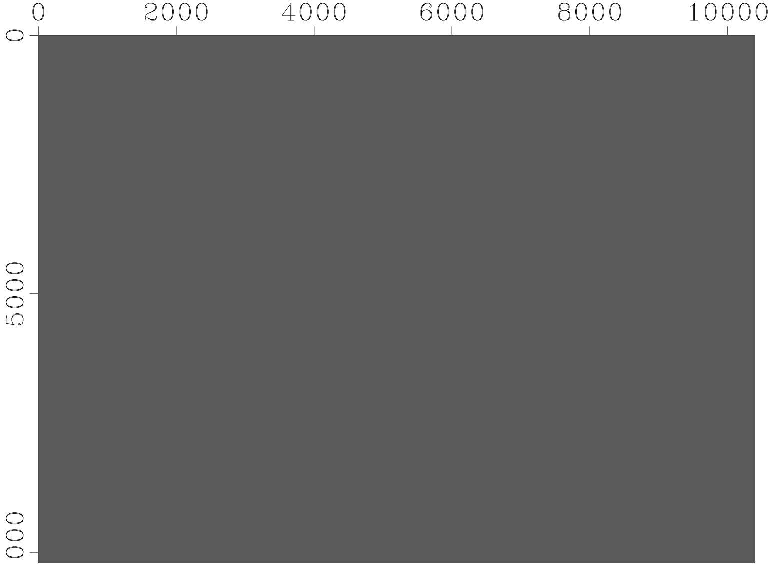
Breaking the Memory Barrier



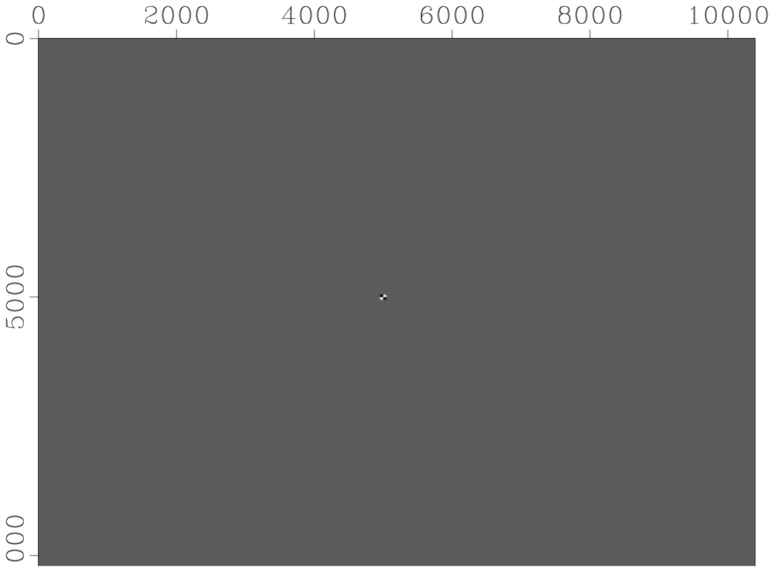
A small example from the Overthrust model



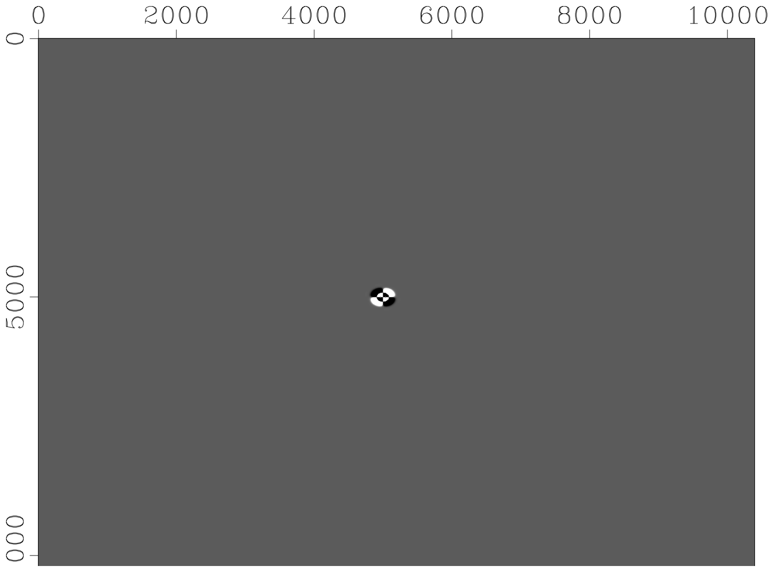
A small example from the Overthrust model



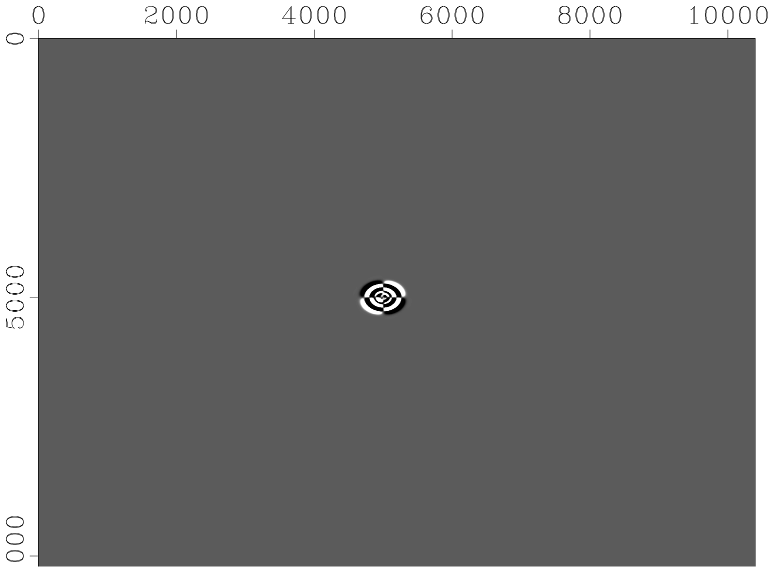
A small example from the Overthrust model



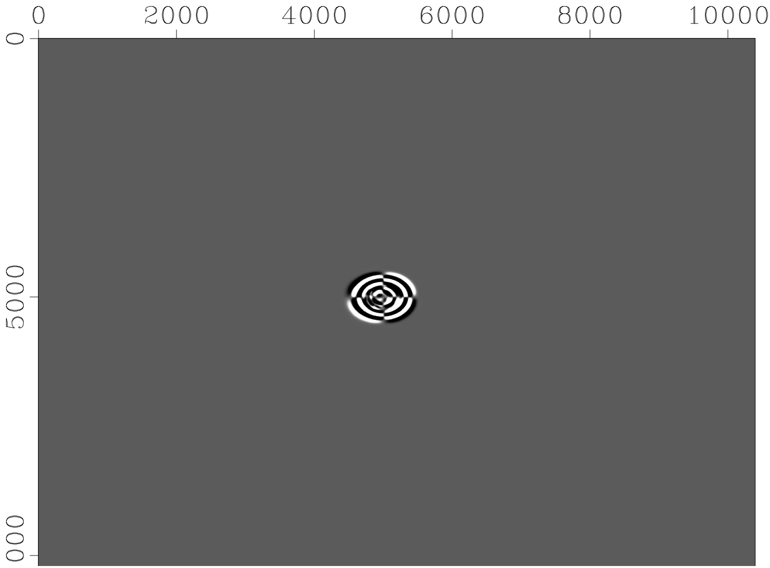
A small example from the Overthrust model



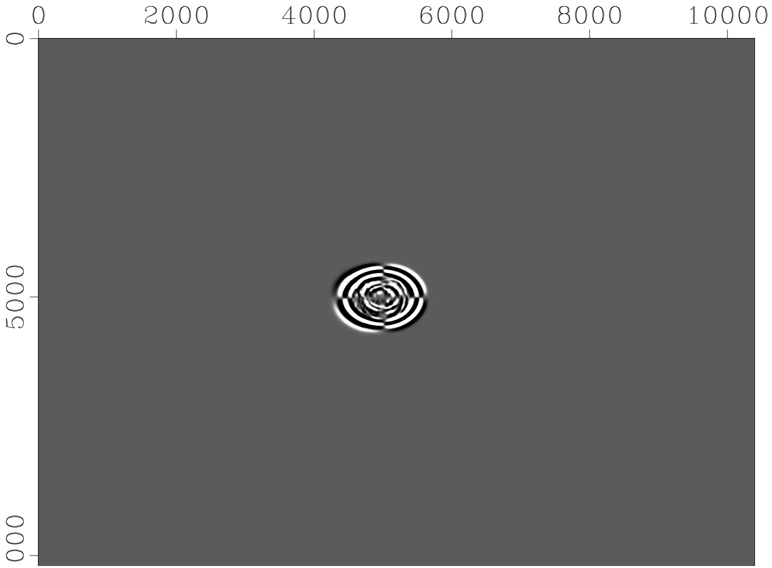
A small example from the Overthrust model



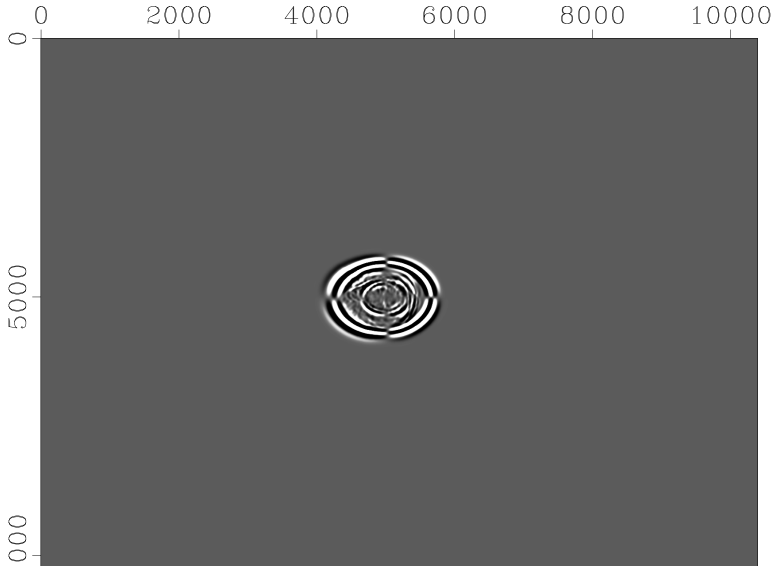
A small example from the Overthrust model



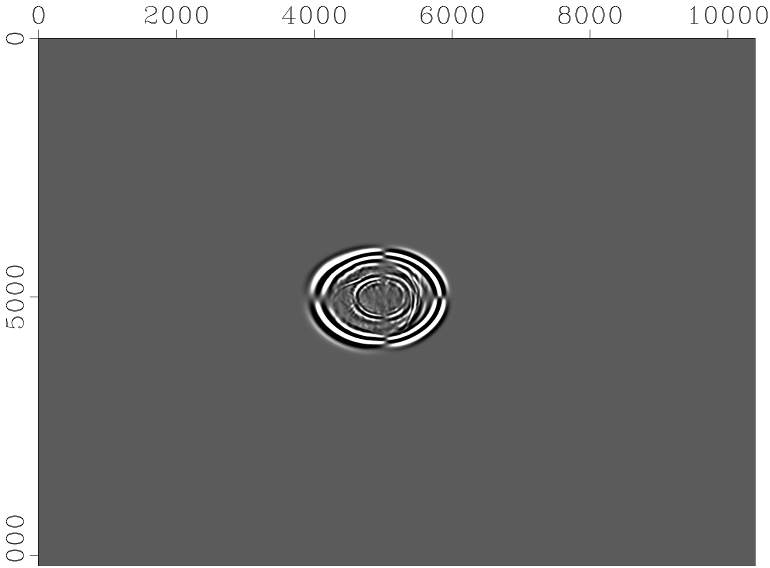
A small example from the Overthrust model



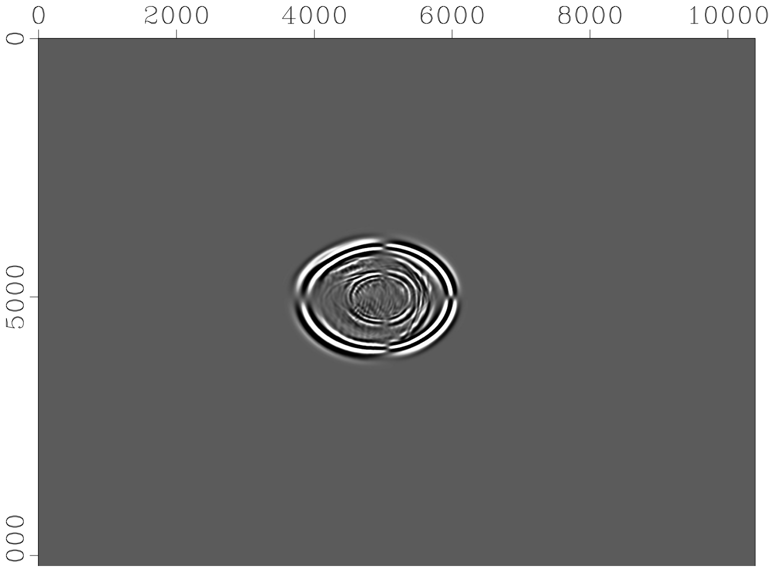
A small example from the Overthrust model



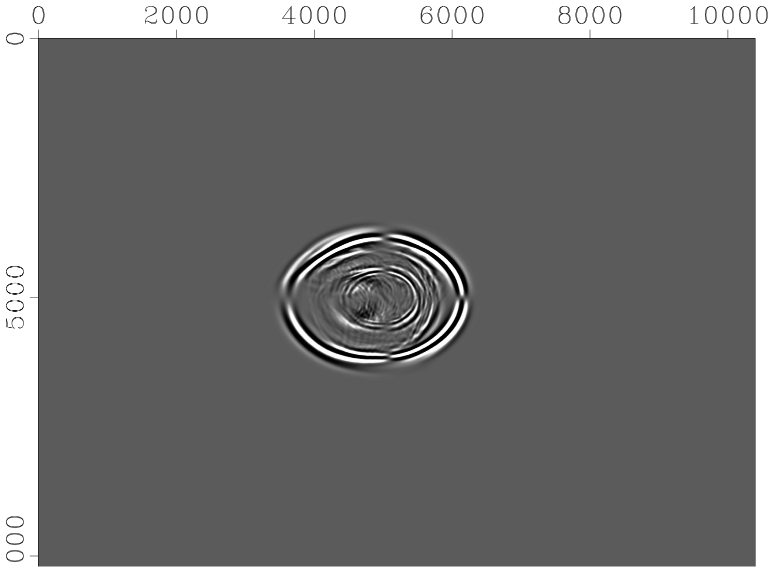
A small example from the Overthrust model



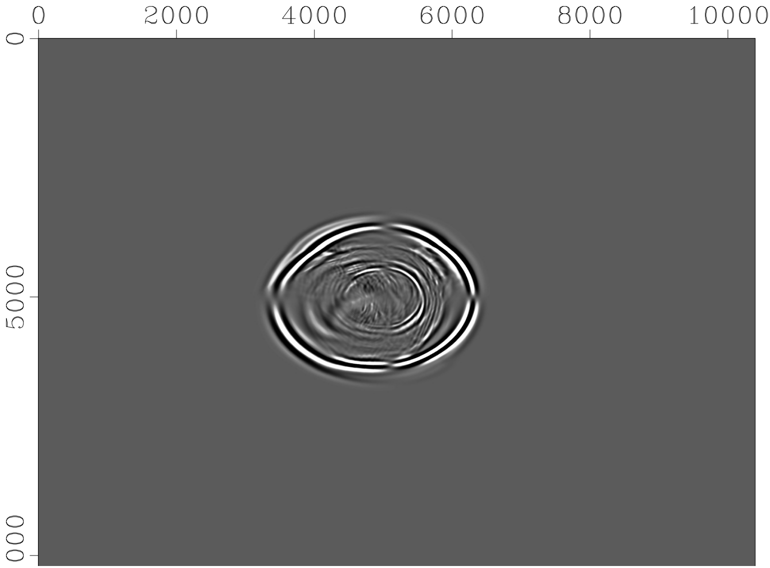
A small example from the Overthrust model



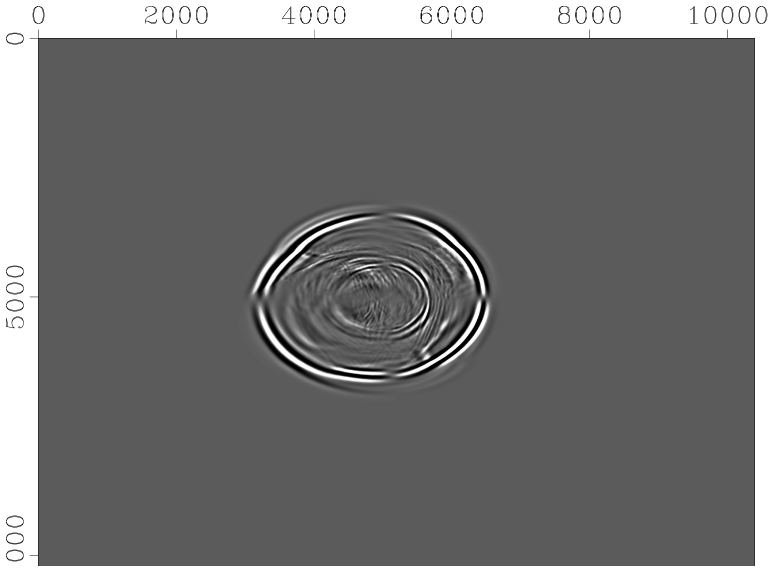
A small example from the Overthrust model



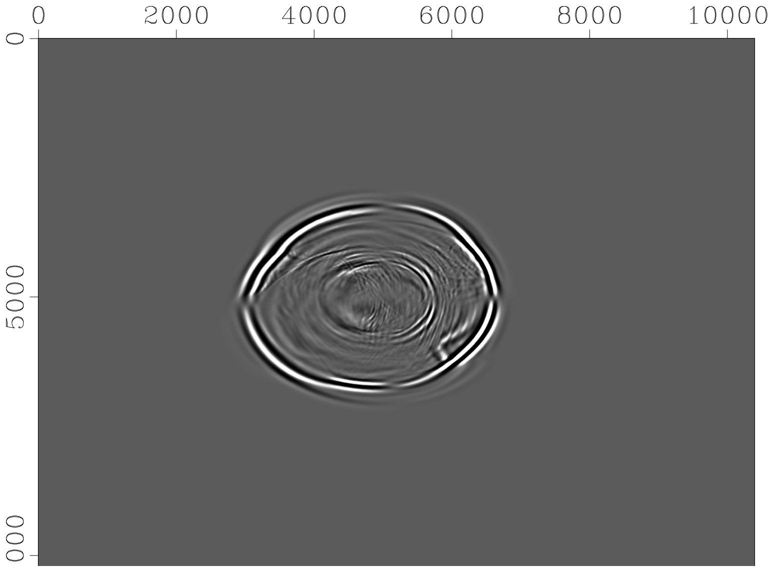
A small example from the Overthrust model



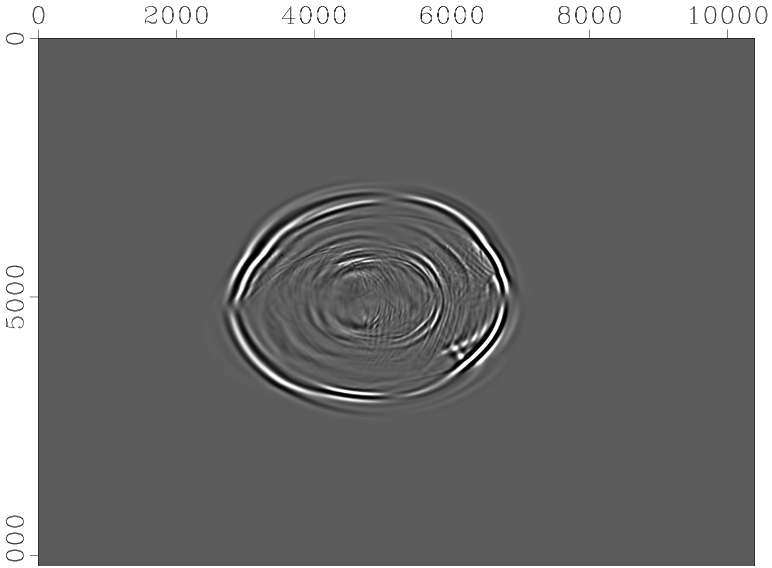
A small example from the Overthrust model



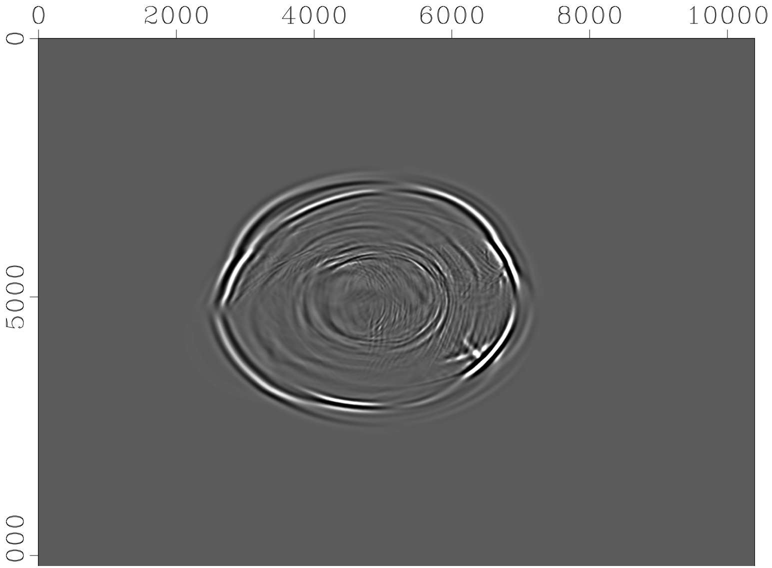
A small example from the Overthrust model



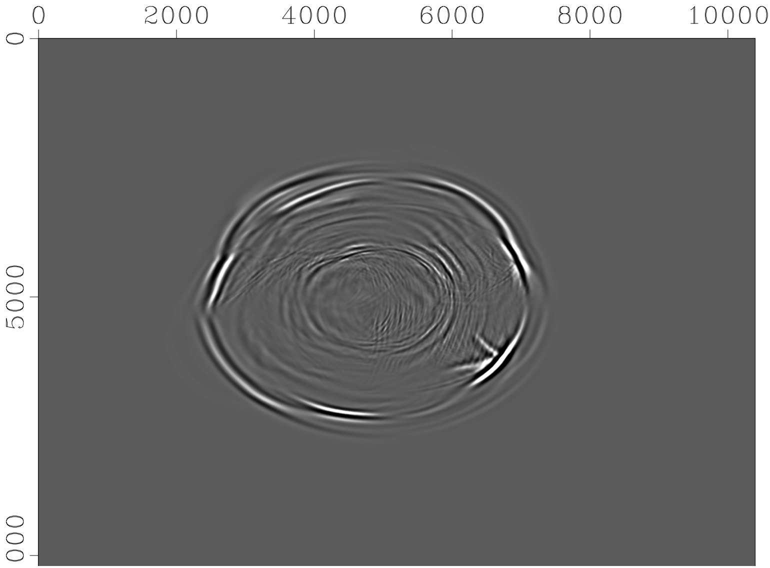
A small example from the Overthrust model



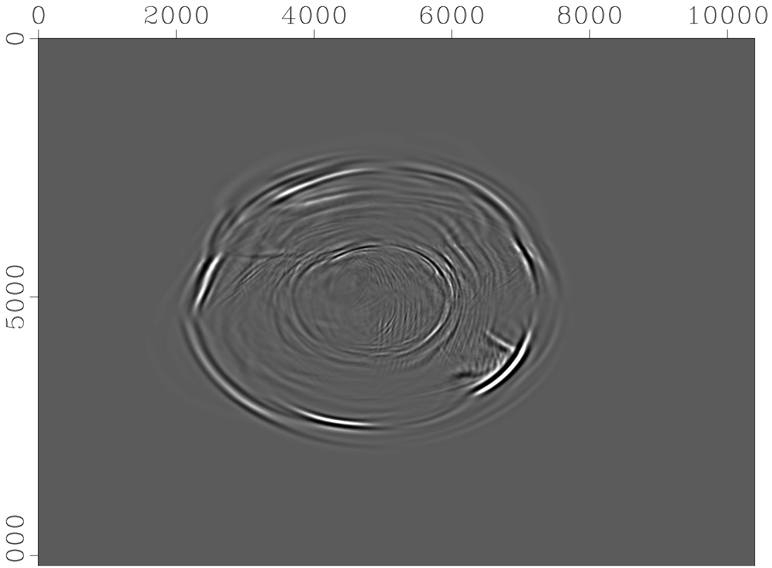
A small example from the Overthrust model



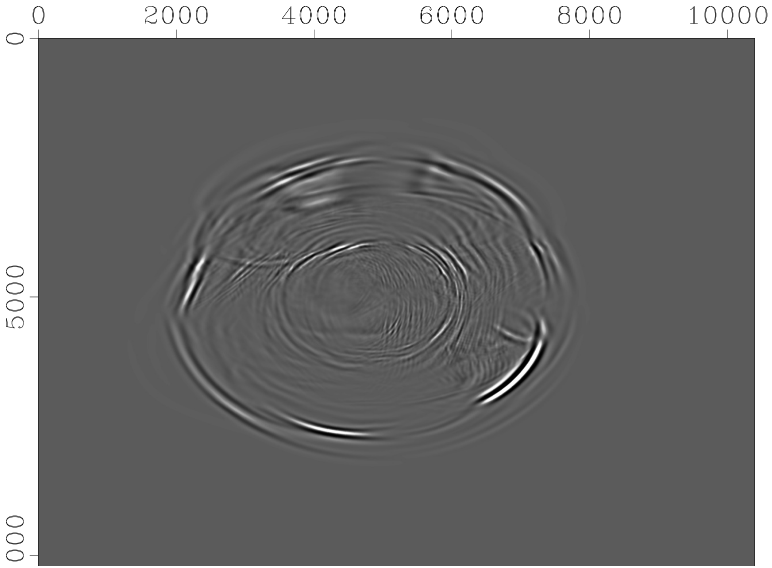
A small example from the Overthrust model



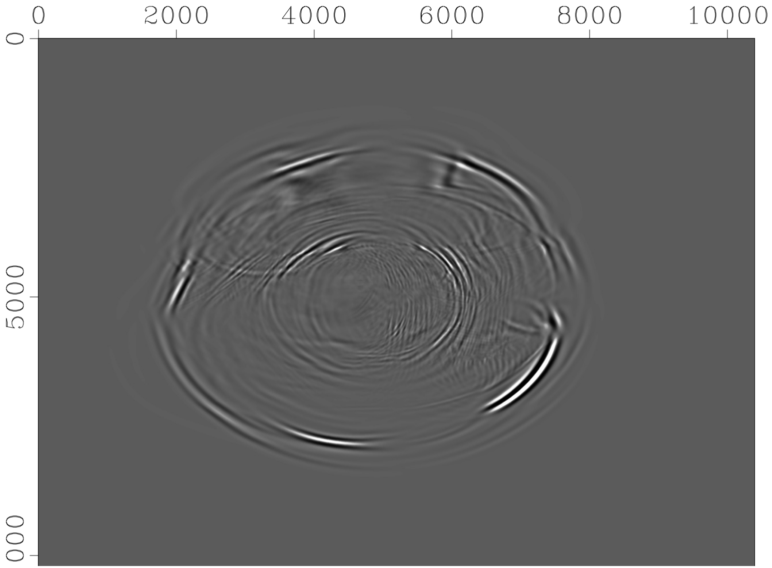
A small example from the Overthrust model



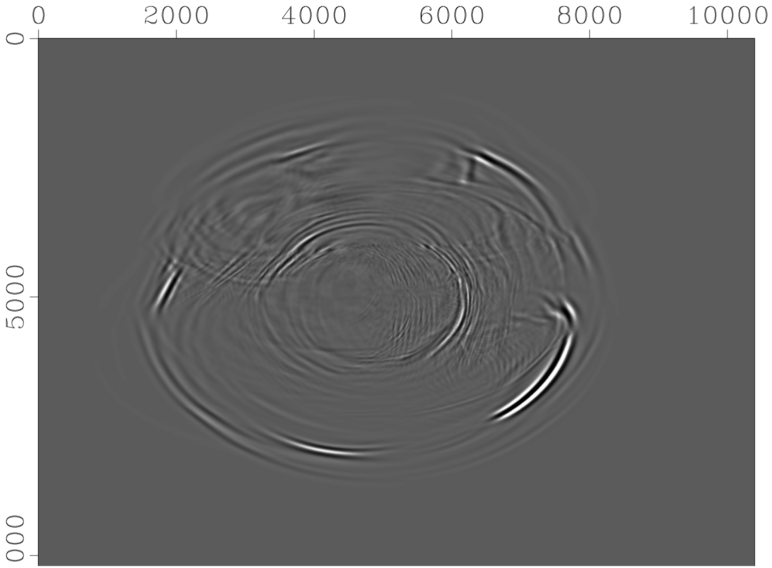
A small example from the Overthrust model



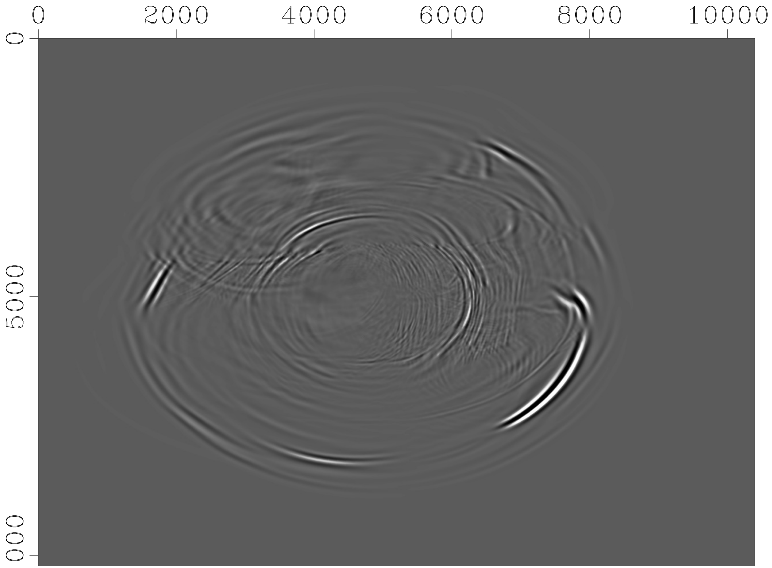
A small example from the Overthrust model



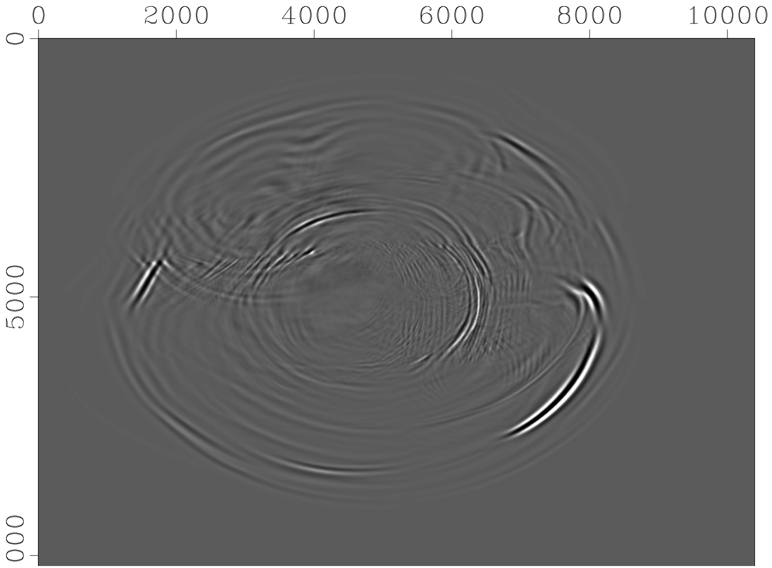
A small example from the Overthrust model



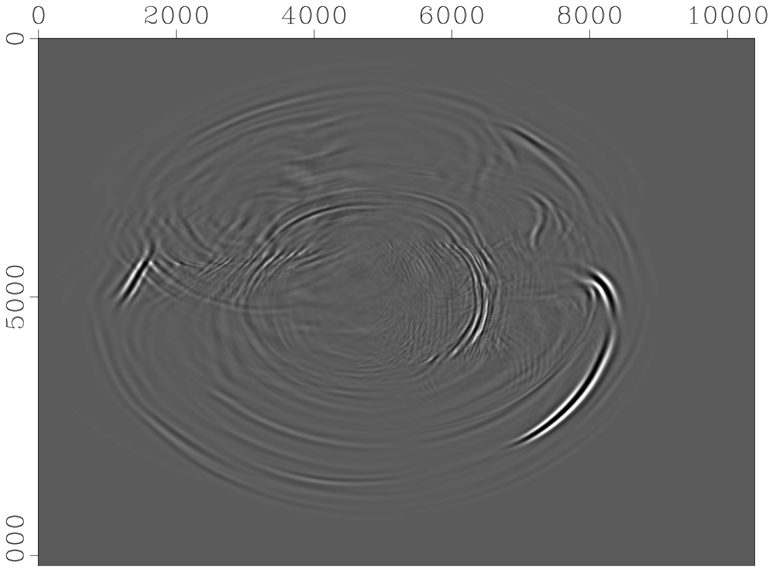
A small example from the Overthrust model



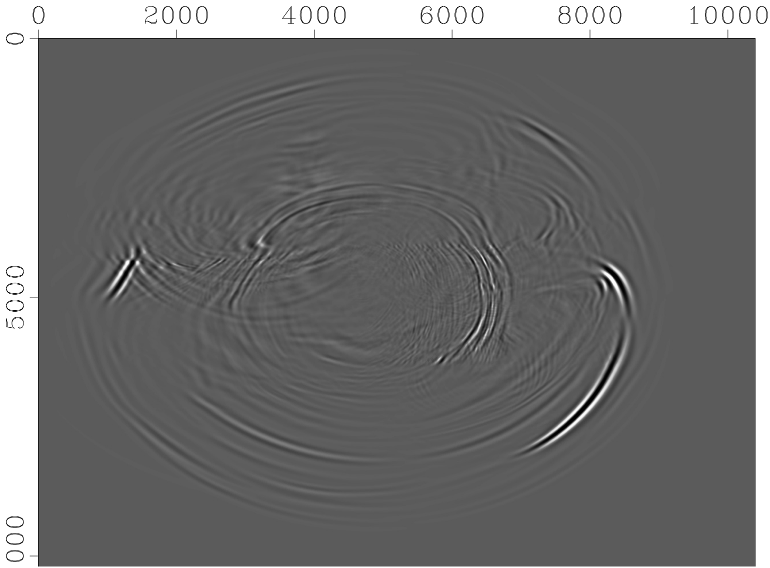
A small example from the Overthrust model



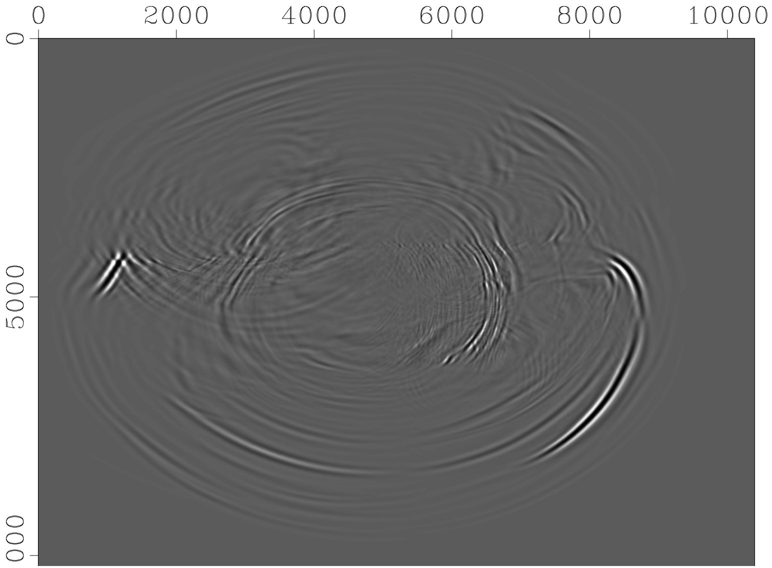
A small example from the Overthrust model



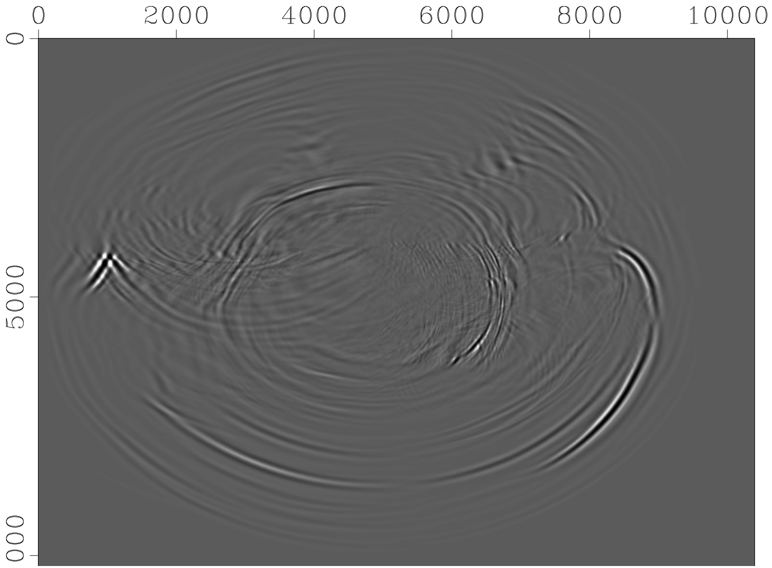
A small example from the Overthrust model



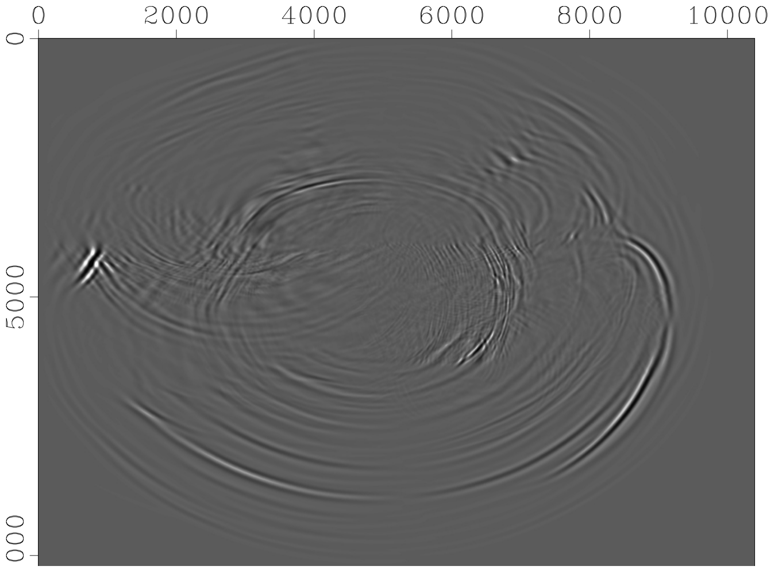
A small example from the Overthrust model



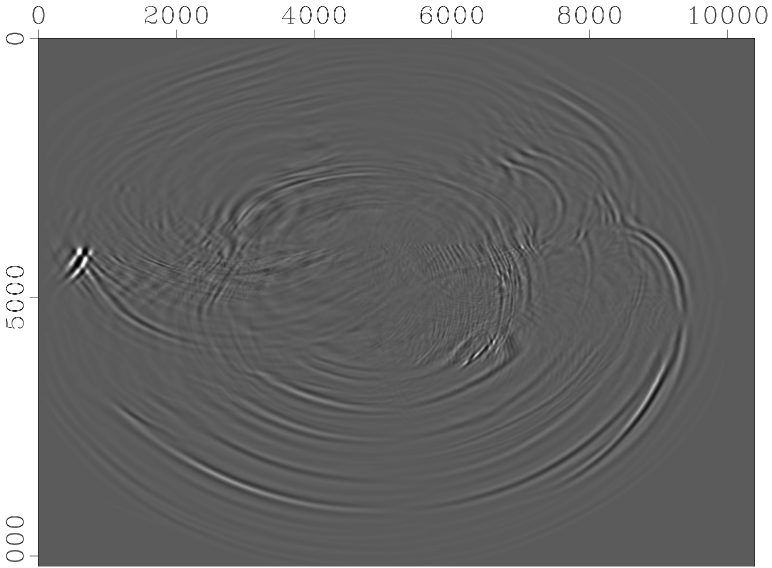
A small example from the Overthrust model



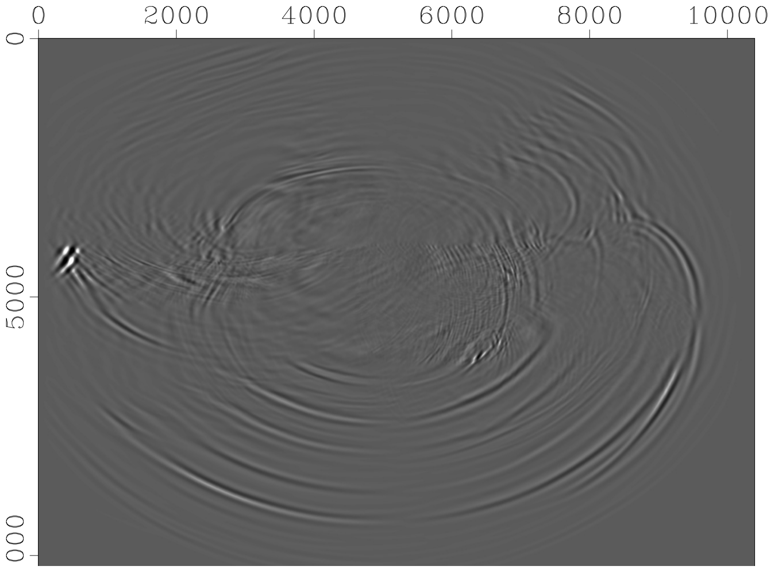
A small example from the Overthrust model



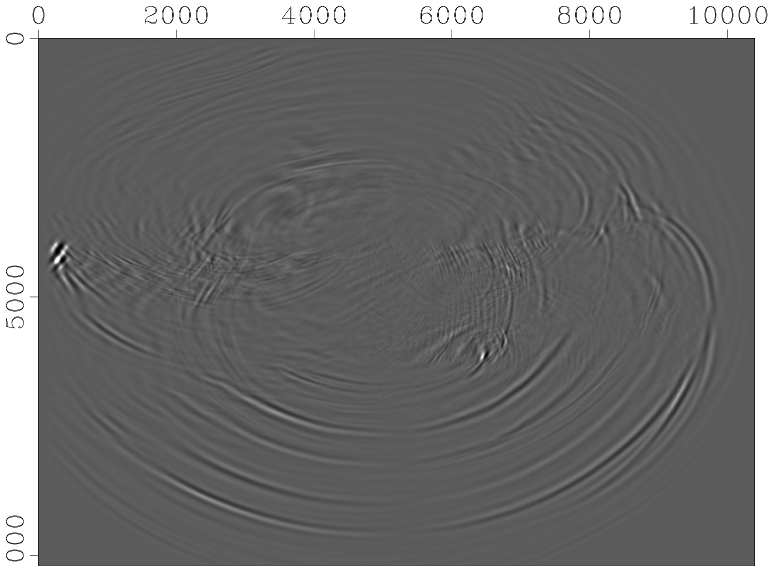
A small example from the Overthrust model



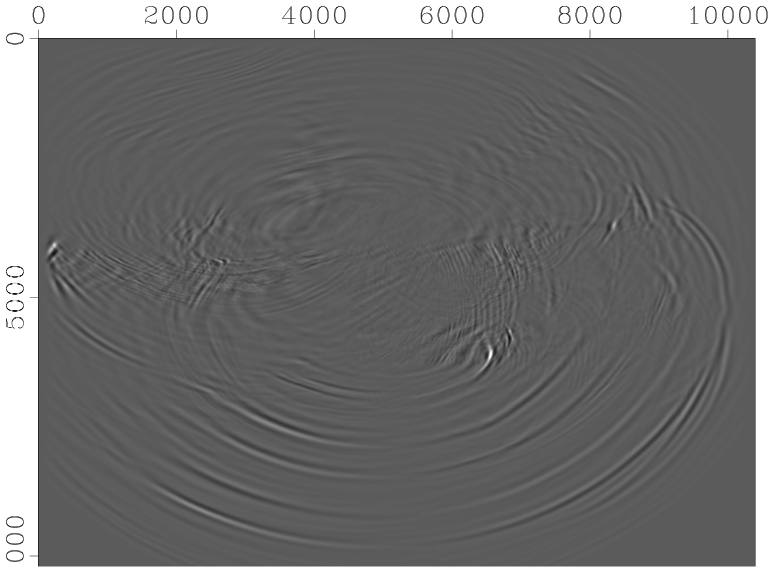
A small example from the Overthrust model



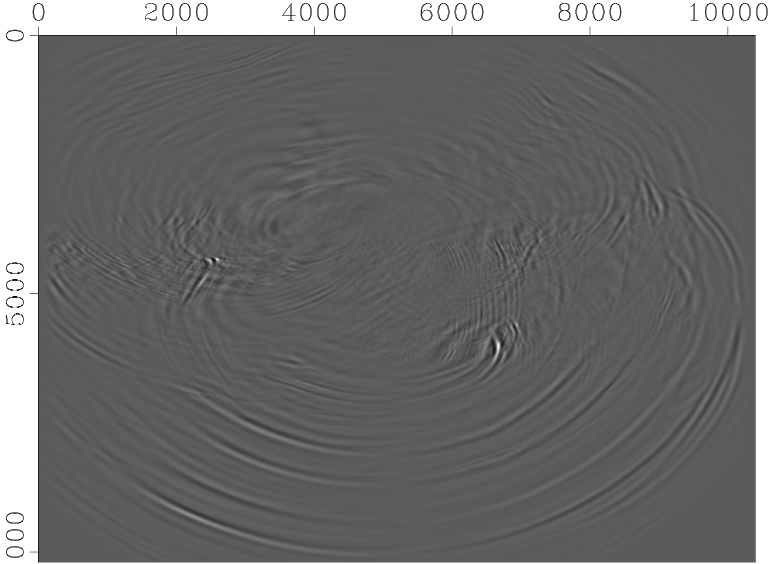
A small example from the Overthrust model



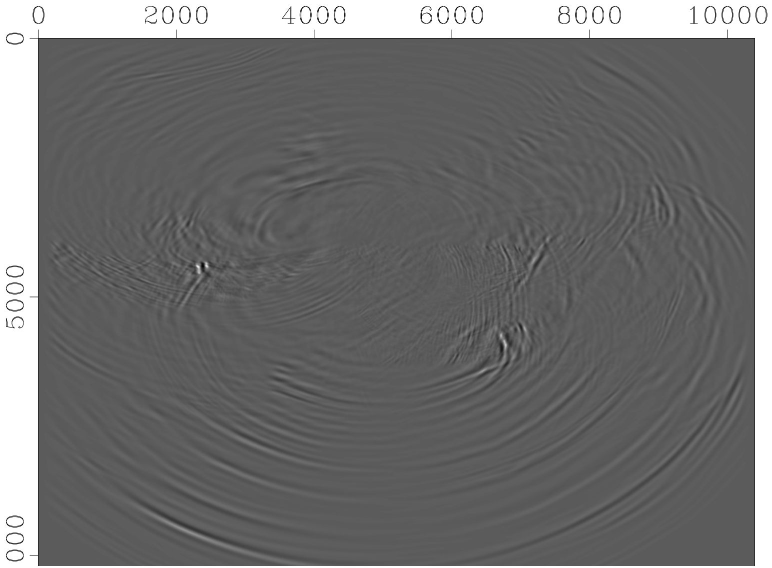
A small example from the Overthrust model



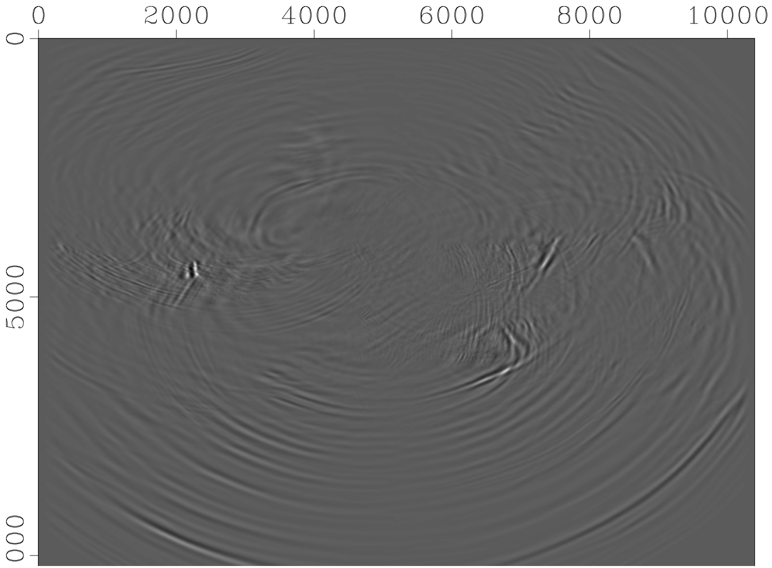
A small example from the Overthrust model



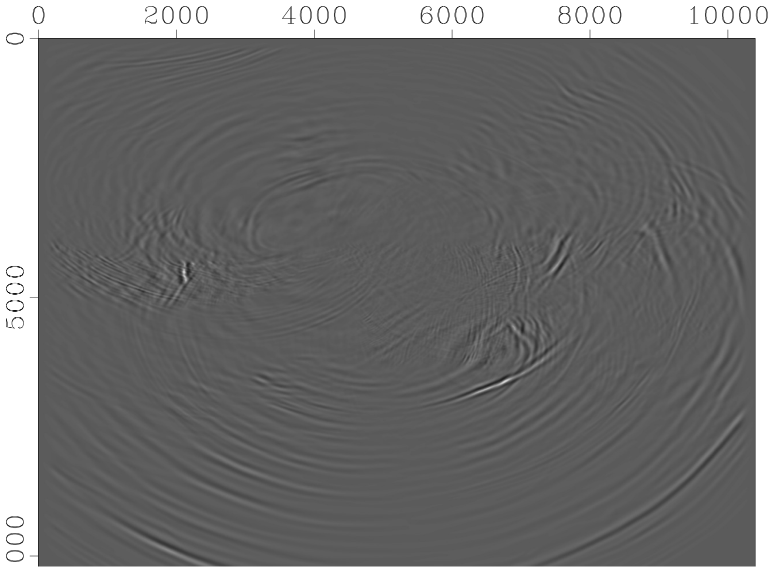
A small example from the Overthrust model



A small example from the Overthrust model



A small example from the Overthrust model



What is the running time?

... for the example model on a three year old consumer GPU ($700)?

What is the running time?

... for the example model on a three year old consumer GPU ($700)?

- $\approx 15\%$ slowdown compared to in-core.

What is the running time?

... for the example model on a three year old consumer GPU (\$700)?

- $\approx 15\%$ slowdown compared to in-core.
- 1.46s per time step. 20000 time steps in 8 hours.

What is the running time?

... for the example model on a three year old consumer GPU ($700)?

- $\approx 15\%$ slowdown compared to in-core.
- 1.46s per time step. 20000 time steps in 8 hours.
- In-house CPU code would use 6 weeks on a single core ...

What is the running time?

... for the example model on a three year old consumer GPU ($700)?

- $\approx 15\%$ slowdown compared to in-core.
- 1.46s per time step. 20000 time steps in 8 hours.
- In-house CPU code would use 6 weeks on a single core ...
- ... or 1 week in parallel on an 8-core CPU.

For your consideration

- GPUs favour computation-heavy, accurate numerical schemes.

For your consideration

- GPUs favour computation-heavy, accurate numerical schemes.
- Computational cost proportional to $\frac{\lambda_{min}}{\Delta x}^4$.

For your consideration

- GPUs favour computation-heavy, accurate numerical schemes.
- Computational cost proportional to ${\frac{\lambda_{min}}{\Delta x}}^4$.
- Shot-parallelism is easily exploited for migration and inversion.

For your consideration

- GPUs favour computation-heavy, accurate numerical schemes.
- Computational cost proportional to ${\frac{\lambda_{min}}{\Delta x}}^4$.
- Shot-parallelism is easily exploited for migration and inversion.
- Arrange GPUs in a pipeline for a lower-level parallelism.

And the verdict is ...

The out-of-core implementation:

And the verdict is ...

The out-of-core implementation:

- Makes large-scale wave modelling feasible on a single GPU.

And the verdict is ...

The out-of-core implementation:

- Makes large-scale wave modelling feasible on a single GPU.
- Gives flexibility and good utilisation of the GPU hardware.

And the verdict is ...

The out-of-core implementation:

- Makes large-scale wave modelling feasible on a single GPU.
- Gives flexibility and good utilisation of the GPU hardware.
- Is easily adopted in a multi-GPU setting.

A big thanks to

I wish to acknowledge Statoil and the sponsors of the ROSE consortium for funding my PhD and the Department of Petroleum Engineering & Applied Geophysics for being a great place to work.

venstad@gmail.com