Pseudospectral Methods for Pseudo Acoustic Wave Propagation

Ben Wards
wardsbd@gmail.com

Overview

- RTM, F-K, and Kirchhoff migration
- Taking a derivative with the FFT
- Dispersion Relations and the wave equation
- Pseudospectral methods
- High-Order Pseudospectral
- Phase-shift time stepping
- Phase-shift time stepping for variable velocity
- Comparison of numerical simulations
- Anisotropy, VTI and TTI
- Aliasing Versus Stability

Depth or Time Migration?

- Time migration:
  - Obeys Snell's law for flat horizontal interfaces.
- Depth migration
  - Obeys Snell's law
- True Amplitude versus ray tracing(eikonal equation)
- Depth continuation algorithms or FK-migration
  - one-way in depth
- Reverse time migration
  - Obeys Snell's law
  - RTM is a depth migration algorithm.

Why RTM

- Accurate propagation through velocity field
- Multipathing
  - Turning waves
  - Free surface reflections
  - Internal multiples
- Multiple Reflections
- True Amplitude

FD versus Pseudospectral

- Finite difference(FD) use local operators and must be oversampled to be accurate
- FD is embarrassingly parallel
- Pseudospectral uses larger grid spacing
  - and so uses less memory
- Pseudospectral uses the whole domain so is not as parallel for GPU

- Taking the derivative in the space domain is the same as multiplying by $2i\pi x_i$ in the wavenumber domain.

$$\frac{\partial U}{\partial x_i}(t,\vec{x}) = \mathrm{FT}_{\vec{k}}^{-1} 2i\pi k_i \, \mathrm{FT}_{\vec{x}} U(t,\vec{x})$$

$$\frac{\partial U}{\partial x_i^2}(t,\vec{x}) = -\mathrm{FT}_{\vec{k}}^{-1} (2\pi)^2 k_i^2 \mathrm{FT}_{\vec{x}} U(t,\vec{x})$$

```python
import numpy as np
import matplotlib.pyplot as plt


xmax  = 2*np.pi*2
dx = 0.05
N  = int(xmax/dx)
x  = np.arange(0.,N, xmax/N)
y  = np.sin(x) + 0.0001 * np.random.random(size=x.shape)
dy_analytical = np.cos(x)


k  = np.fft.fftfreq(N, d = dx) * 2 * np.pi
```

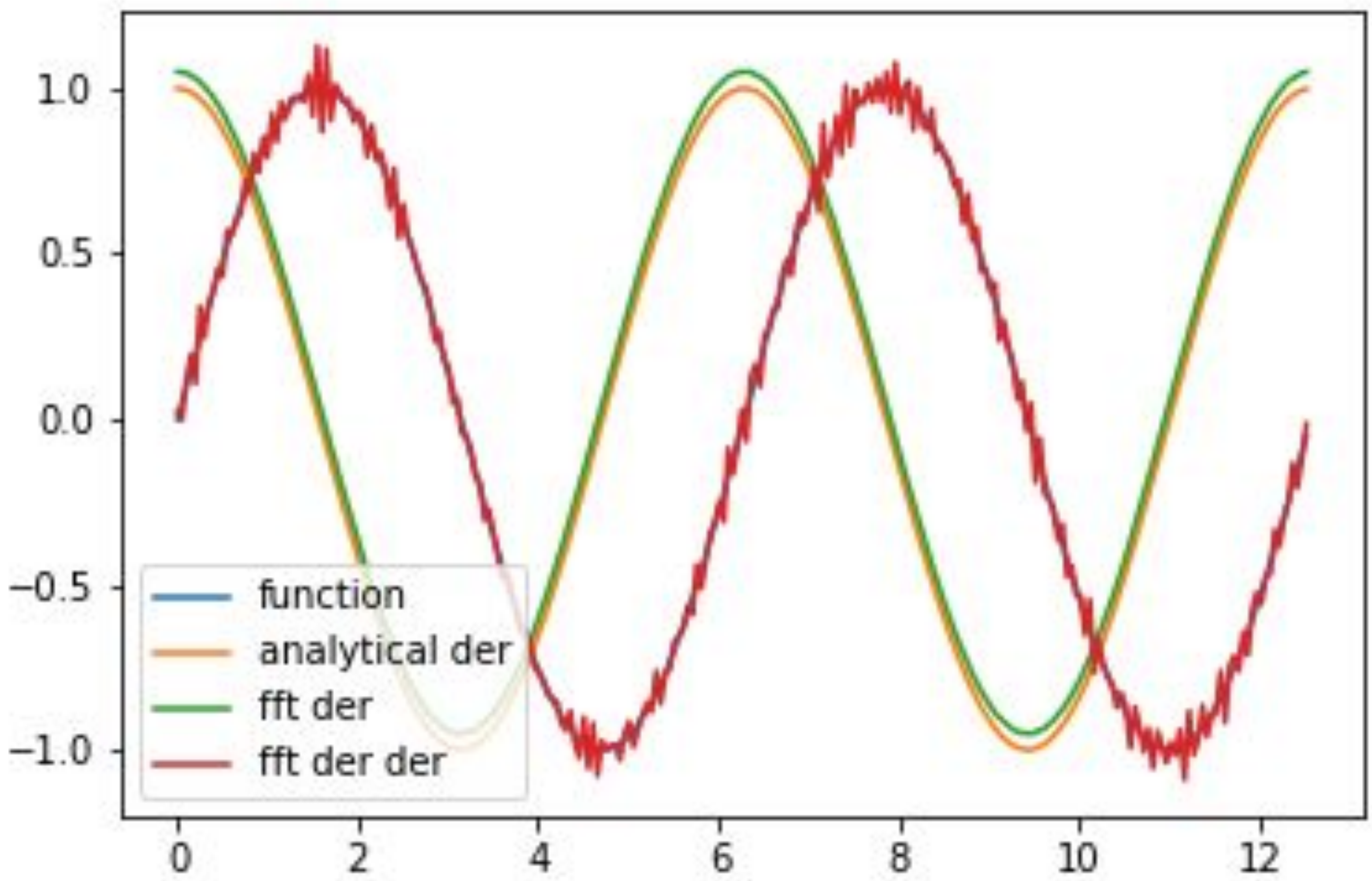
Calculating the first derivative,

```python
fd = np.real(np.fft.ifft(1.0j * k * np.fft.fft(y)))
```

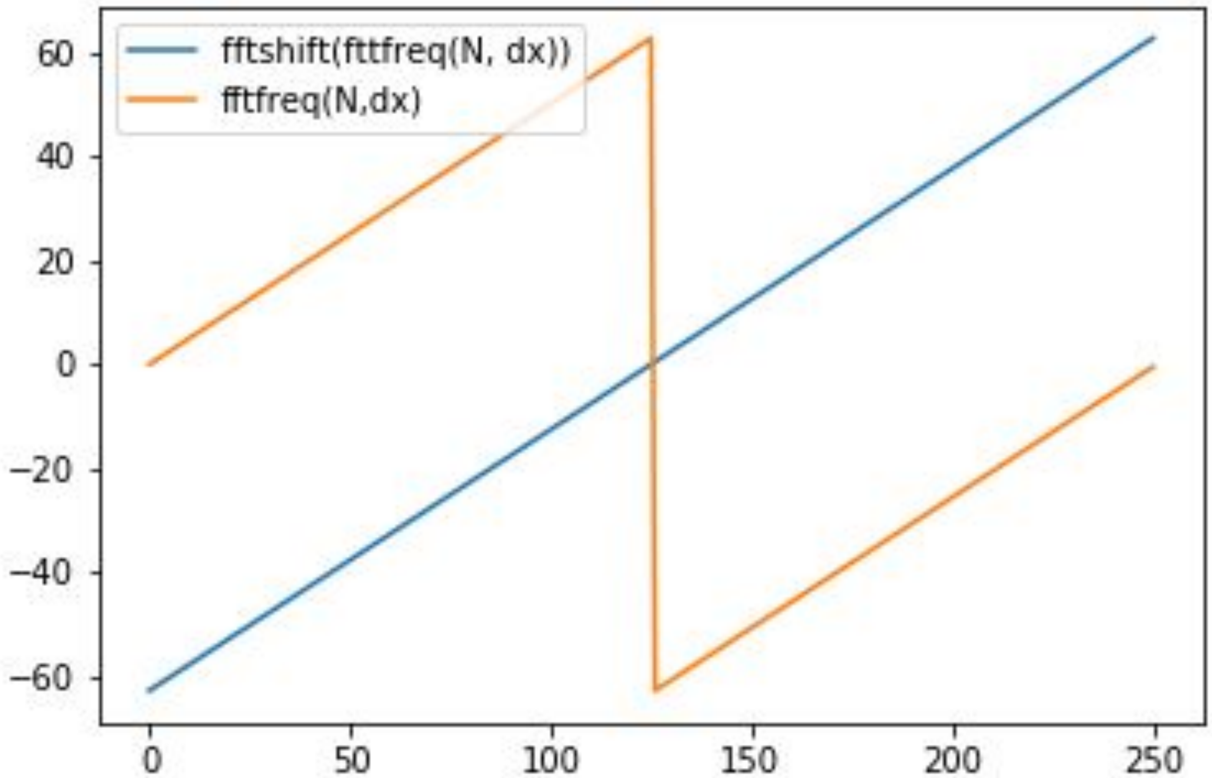
Calculating the second derivative,

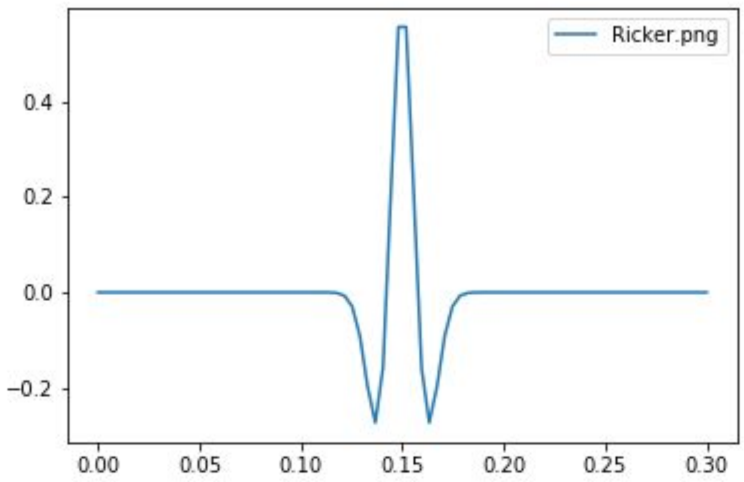
```python
fdd= np.real(np.fft.ifft( -k**2 * np.fft.fft(y)))
```

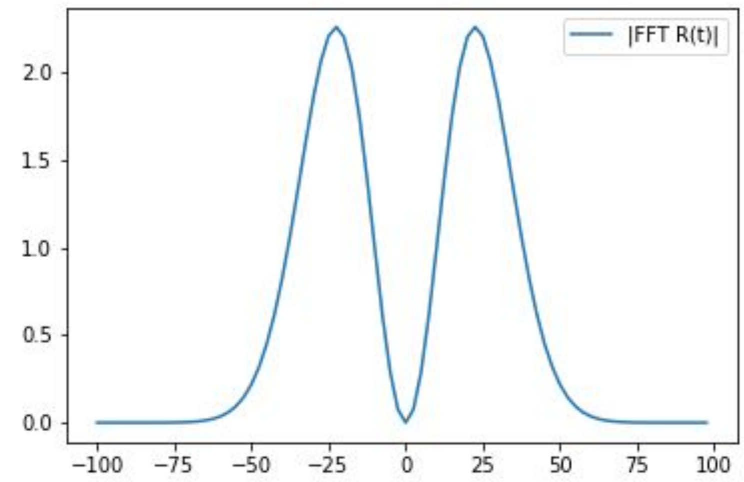
Derivative with FFT

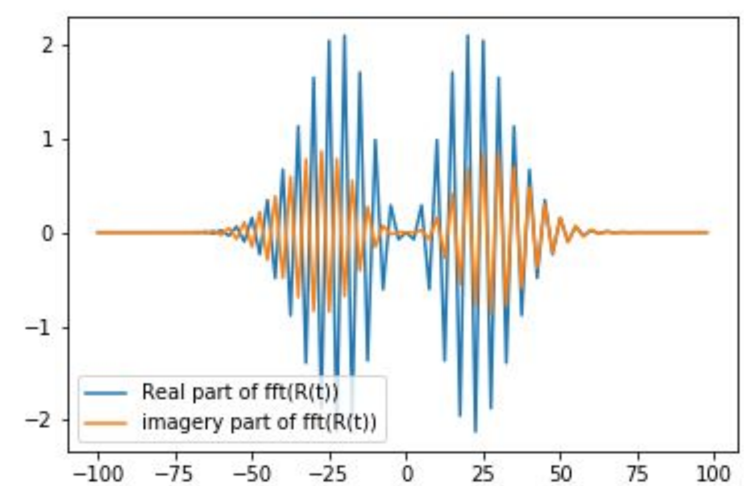


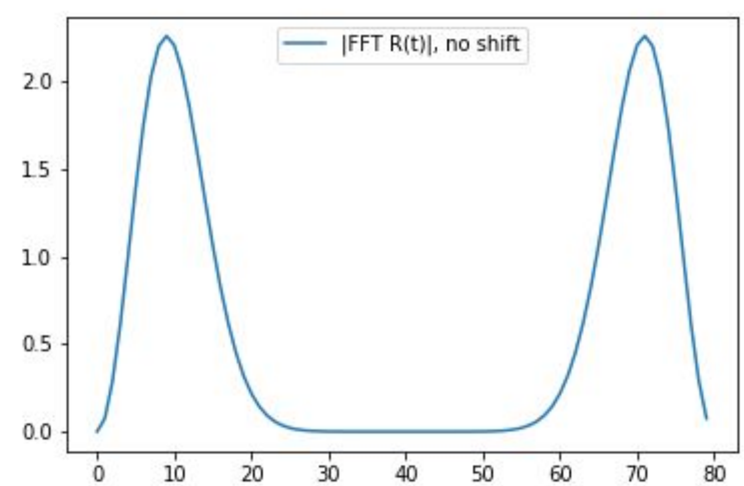
The FFT splits the output:




Ricker.png


|FFT R(t)|


Real part of fft(R(t))
imagery part of fft(R(t))


|FFT R(t)|, no shift

Pseudospectral

$$\Delta = \frac{\partial^2}{\partial x^2} + \ldots + \frac{\partial^2}{\partial z^2}$$

$$\frac{\partial^2 U}{\partial t^2}(t, \vec{x}) = c^2 \Delta U(t, \vec{x})$$

- Calculated Laplacian in Fourier Domain

$$\frac{\partial^2 U}{\partial t^2}(t, \vec{x}) = -c^2 \mathrm{FT}_{\vec{k}}^{-1}\left\{ |\vec{k}|^2 \, \mathrm{FT}_{\vec{x}}\{U(t, \vec{x})\} \right\}$$

- Centered finite difference

$$U(\Delta t + t, \vec{x}) = -U(-\Delta t + t, \vec{x}) + 2U(t, \vec{x})$$
$$-(c\Delta t)^2 \mathrm{FT}_{\vec{k}}^{-1}\left\{ |\vec{k}|^2 \, \mathrm{FT}_{\vec{x}}\{U(t, \vec{x})\} \right\}$$

Dispersions Relations

- Apply a Plane wave to a PDE like the acoustic wave equation

$$U(t,\vec{x}) = A_0 \exp(i(\omega t - \vec{x} \bullet \vec{k}))$$

$$\frac{\partial^2 U}{\partial t^2}(t,\vec{x}) = c^2 \Delta U(t,\vec{x})$$

$$\omega^2 = c^2 |\vec{k}|^2$$

- Given a Dispersion Relation do you get a wave equation?

One-way in time propagation

- Approximate the wave equation by the one-way in time wave equation

$$\omega = \pm c\,|\vec{k}\,|$$

$$\frac{\partial U}{\partial t}(t,\vec{x}) = \pm c\sqrt{-\Delta}\,U(t,\vec{x})$$

- Or with the Fourier transform

$$\frac{\partial \hat{U}}{\partial t}(t,\vec{k}) = \pm 2\pi c i\,|\vec{k}\,|\,\hat{U}(t,\vec{k})\}$$

- Which has the solution

$$U(t+\delta t,\vec{x}) = \mathrm{FT}_{\vec{k}}^{-1}\left\{e^{\pm 2\pi i c\delta t|\vec{k}|}\mathrm{FT}_{\vec{x}}\{U(t,\vec{x})\}\right\}$$

-Zhang & Zhang (2009)

- One-way depth step Extrapolation: $k_z = \pm\sqrt{\omega^2 / c^2 - k_x^2}$

$$U(z+\Delta z, k_x, \omega) = e^{i\Delta z\sqrt{\frac{\omega^2}{v^2} - k_x^2}} U(z, k_x, \omega)$$

- Two-way phase-shift timestepping extrapolator:

$$U(t+\Delta t, k_x, k_z)$$
$$= \left(e^{+i\Delta t v\sqrt{k_x^2+k_z^2}} + e^{-i\Delta t v\sqrt{k_x^2+k_z^2}}\right) U(t, k_x, k_z) - U(t-\Delta t, k_x, k_z)$$

- One-way phase-shift timestepping extrapolator:

$$U(t+\Delta t, k_x, k_z) = \left(e^{\pm i\Delta t v\sqrt{k_x^2+k_z^2}}\right) U(t, k_x, k_z)$$

Boundary Conditions

- Pad and apply a taper at the Boundary
- Zero velocity layer at the top to produce a reflecting boundary condition.

Higher-order pseudospectral

- Taylor series

$$U(\delta t+t,x)=U(t,x)+\frac{\partial U}{\partial t}(t,x)\delta t+\frac{1}{2}\frac{\partial^2 U}{\partial t^2}(t,x)\delta t^2+\ldots$$

$$U(-\delta t+t,x)=U(t,x)-\frac{\partial U}{\partial t}(t,x)\delta t+\frac{1}{2}\frac{\partial^2 U}{\partial t^2}(t,x)\delta t^2+\ldots$$

- Adding up

$$U(\delta t+t,x)+U(\delta t-t,x)=2U(t,x)+\frac{\partial^2 U}{\partial t^2}(t,x)\delta t^2+\frac{2}{4!}\frac{\partial^4 U}{\partial t^4}(t,x)\delta t^4+\ldots$$

Higher-Order Pseudospectralå

$$\frac{\partial^4 U}{\partial t^4}(t,x) = \frac{\partial^2}{\partial t^2}\frac{\partial^2 U}{\partial t^2} = \frac{\partial^2}{\partial t^2}\left(c^2 \Delta U\right) = c^2 \Delta \frac{\partial^2}{\partial t^2} U$$

$$= c^2 \Delta\left(c^2 \Delta U\right)$$

$$= c^4 \Delta^2 U + c^2 \Delta(c^2)\Delta U + c^2 \nabla(c^2) \bullet \nabla(\Delta U)$$

$$\approx c^4 \Delta^2 U$$

- Insert time derivative approximation into wave equation

$$U(\delta t + t) = -U(-\delta t + t) + 2U(t) - (c\delta t)^2 \mathrm{FT}_{\vec{k}}^{-1}\left\{|\vec{k}|^2\, \mathrm{FT}_{\vec{x}}\{U(t)\}\right\} + \frac{2}{4!}\frac{\partial^4 U}{\partial t^4}(t)\delta t^4$$

$$U(\delta t + t) = -U(-\delta t + t) + 2U(t) - (c\delta t)^2 \mathrm{FT}_{\vec{k}}^{-1}\left\{|\vec{k}|^2\, \mathrm{FT}_{\vec{x}}\{U(t)\}\right\}$$

$$+\frac{2}{4!}(c\delta t)^4 \mathrm{FT}_{\vec{k}}^{-1}\left\{|\vec{k}|^4\, \mathrm{FT}_{\vec{x}}\{U(0,\vec{x})\}\right\}$$

Modified eq approach

Cohen(2002)

Exact solution of constant velocity wave equation

$$\frac{\partial \hat{U}}{\partial t^2}(t,\vec{k}) = c^2 \, |\vec{k}|^2 \, \hat{U}(t,\vec{k})\}$$

$$U(\Delta t + t,\vec{x}) = -U(-\Delta t + t,\vec{x})$$
$$+2\mathrm{FT}_{\vec{k}}^{-1}\left\{\cos(2\pi c\,|\vec{k}|\,\Delta t)\mathrm{FT}_{\vec{x}}\{U(t,\vec{x})\}\right\}$$

Constant velocity to variable velocity

$$U(\Delta t,\vec{x}) = -U(-\Delta t,\vec{x}) +$$
$$2\mathrm{FT}_{\vec{k}}^{-1}\left\{\cos(2\pi c(\vec{x})\,|\vec{k}|\,\Delta t)\mathrm{FT}_{\vec{x}}\{U(0,\vec{x})\}\right\}$$

Cosine Interpolation
(Etgen 2009)

$$\cos(v(x)\,|\,k\,|\,dt) \approx$$

$$\left(\frac{v_{\max}^2 - v^2(x)}{v_{\max}^2 - v_{\min}^2}\right)\cos(v_{\min}\,|\,k\,|\,dt) + \left(\frac{v^2(x) - v_{\min}^2}{v_{\max}^2 - v_{\min}^2}\right)\cos(v_{\max}\,|\,k\,|\,dt)$$

$$\approx 1 - \frac{\left(v\ dt\,|\,k\,|\right)^2}{2!} + \frac{\left(v\ dt\,|\,k\,|\right)^4}{4!} - \frac{\left(v^2 - v_{\max}^2\right)\left(v^2 - v_{\min}^2\right)}{4!}dt^4\,|\,k\,|^4 + \ldots$$

$$\cos(v(x)\,|\,k\,|\,dt) \approx \sum_{n=1}^{N} F_n(\vec{x})G_n(\vec{k})$$

Song, X., and Fomel, S., 2010
Fowler 2009
Yu Zhang 2008

Aliasing and Time-Stepping in 2D

The maximum wavenumber:
$$\vec{k} = \left(\pm \frac{\pi}{\Delta x}, \pm \frac{\pi}{\Delta x} \right)$$

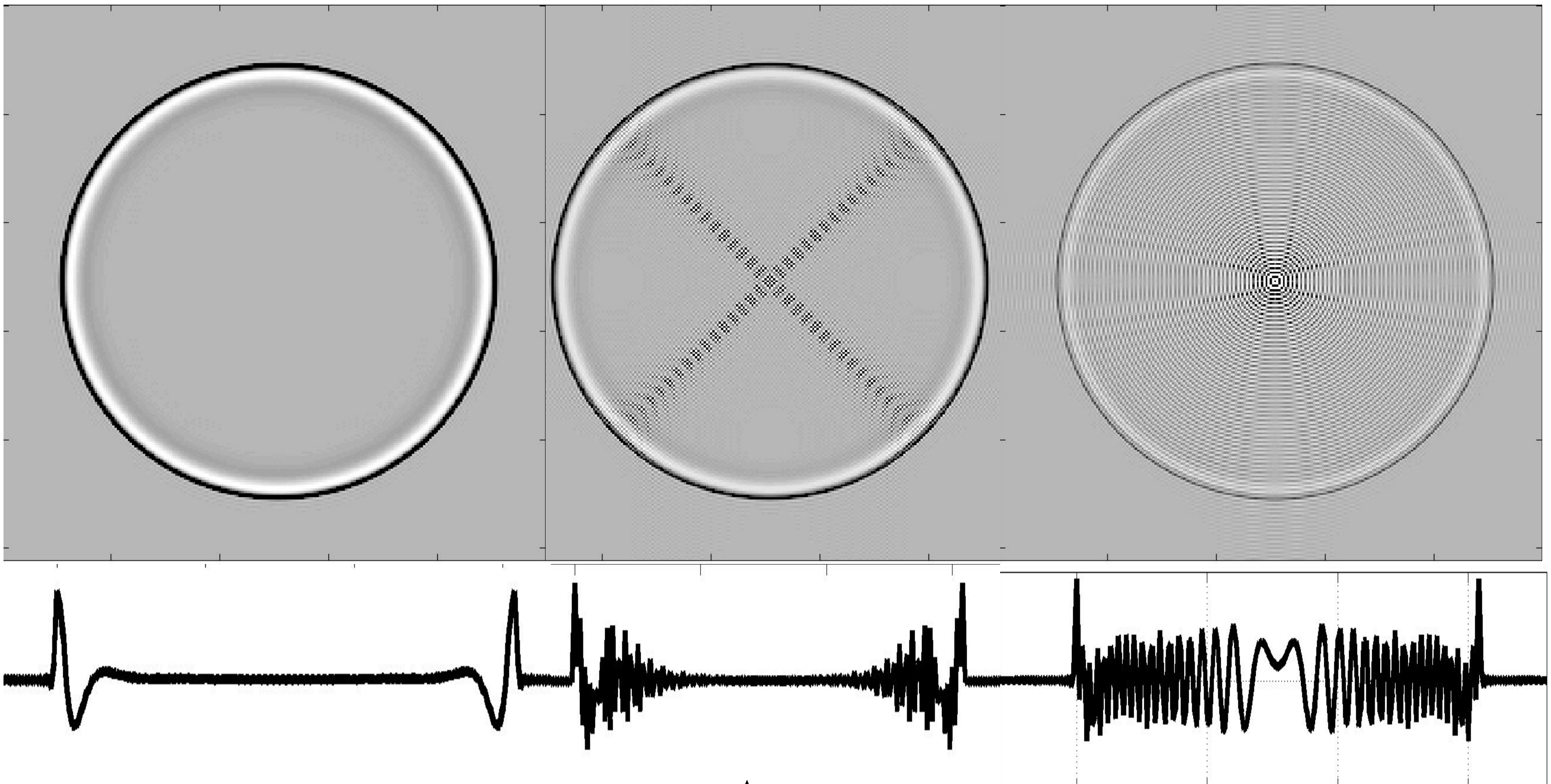
From the dispersion relation
$$f(\vec{k}) = c \, \| \vec{k} \|$$

Generates max frequencies:
$$f = \frac{\sqrt{2}\pi c}{\Delta x}$$

The time variable has a Nyquist number
$$f_{nyq} = \frac{\pi}{\Delta t}$$

Therefore,
$$\frac{c\Delta t}{\Delta x} < \frac{1}{\sqrt{2}}$$

Propagation Near Stability



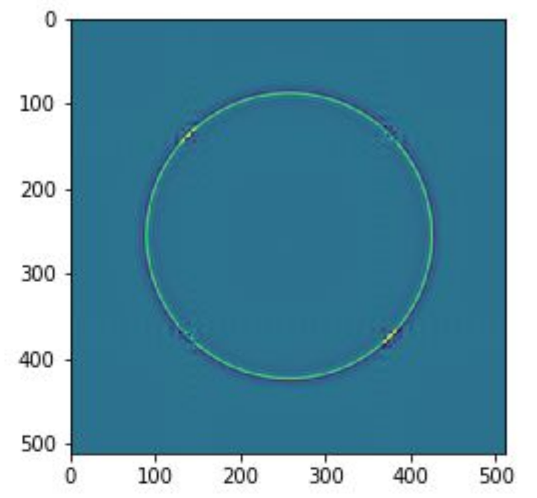
$$0.6 = \frac{c\Delta t}{\Delta x} < 0.71 \qquad 0.8 = \frac{c\Delta t}{\Delta x} > 0.71 \qquad 1.0 = \frac{c\Delta t}{\Delta x} > 0.71$$

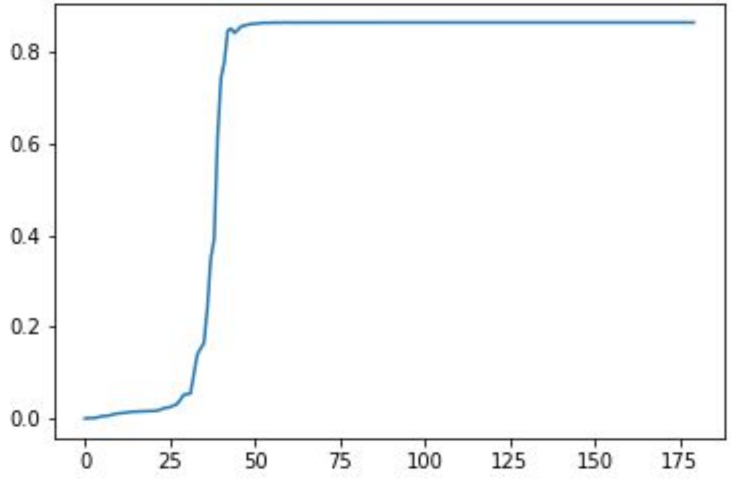
Pseudospectral Code

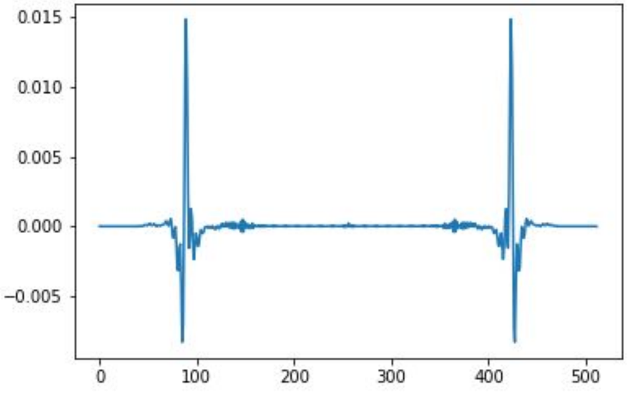
```
kx = np.fft.fftfreq(len(x), d=dx) * 2*np.pi
kz = np.fft.fftfreq(len(z), d=dx).T *2*np.pi
kxx = np.outer(np.ones(kz.shape), (kx**2))
kzz = np.outer(kz**2, np.ones(kx.shape))
powcos = np.zeros(nsteps)
v=2000
opk2 = -(kzz + kxx)
for istep in range(0,nstepstest):
   spec = np.fft.fft2(snapnow)
   tmp = 2 * snapnow - snapthen + dt**2 * v**2 * (np.fft.ifft2(spec * opk2))

   snapthen = snapnow
   snapnow = tmp
   if(istep < len(w_ormsby)):
       snapnow[nz//2,nx//2]=snapnow[nz//2,nx//2]+w_ormsby[istep]

   powcos[istep] = np.linalg.norm(snapnow)
```

Aliased cosine Extrapolation

- Courant number is 1.2
- The pseudospectral method stable at 0.5,
- higher order stable at 0.7







Rewrite wave equation as a system

(JB Chen, 2006)

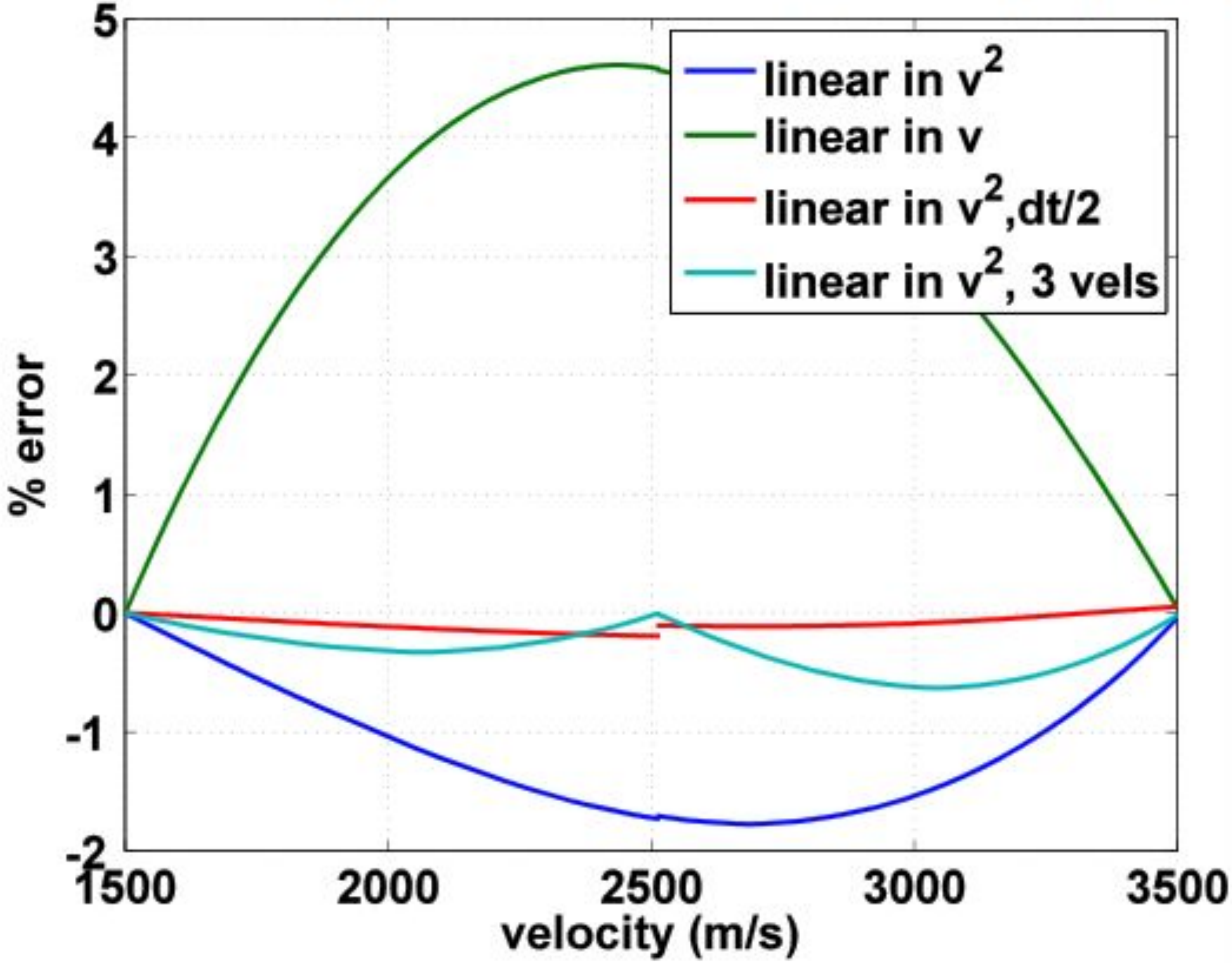
$$\partial_t \begin{bmatrix} \mathrm{U} \\ \mathrm{V} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ v^2\Delta & 0 \end{bmatrix} \begin{bmatrix} \mathrm{U} \\ \mathrm{V} \end{bmatrix},$$

Which for constant velocity has the solution

$$\begin{bmatrix} \mathrm{U}^{\mathrm{n+1}} \\ \mathrm{V}^{\mathrm{n+1}} \end{bmatrix} = FT_{\vec{k}}^{-1} \left\{ \begin{bmatrix} \cos(\omega\delta t) & \sin(\omega\delta t)/\omega \\ -\omega\sin(\omega\delta t) & \cos(\omega\delta t) \end{bmatrix} FT_{\vec{x}} \begin{bmatrix} \mathrm{U}^{\mathrm{n}} \\ \mathrm{V}^{\mathrm{n}} \end{bmatrix} \right\}$$

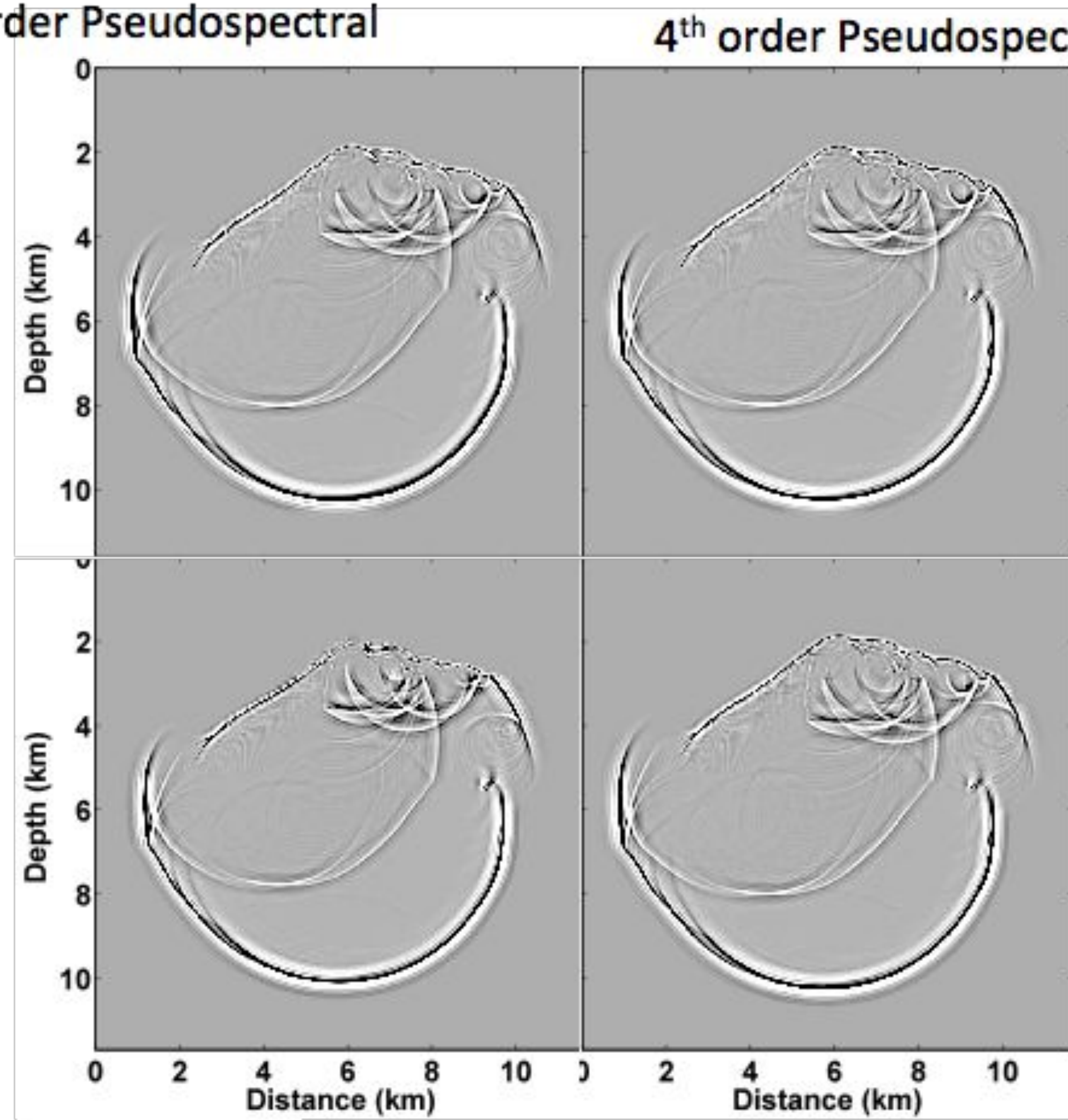
$$\omega = 2\pi v \,|\, \vec{k} \,|$$

Error in approximating cos(w(k,x)dt)



Second Order Pseudospectral

4th order Pseudospectral

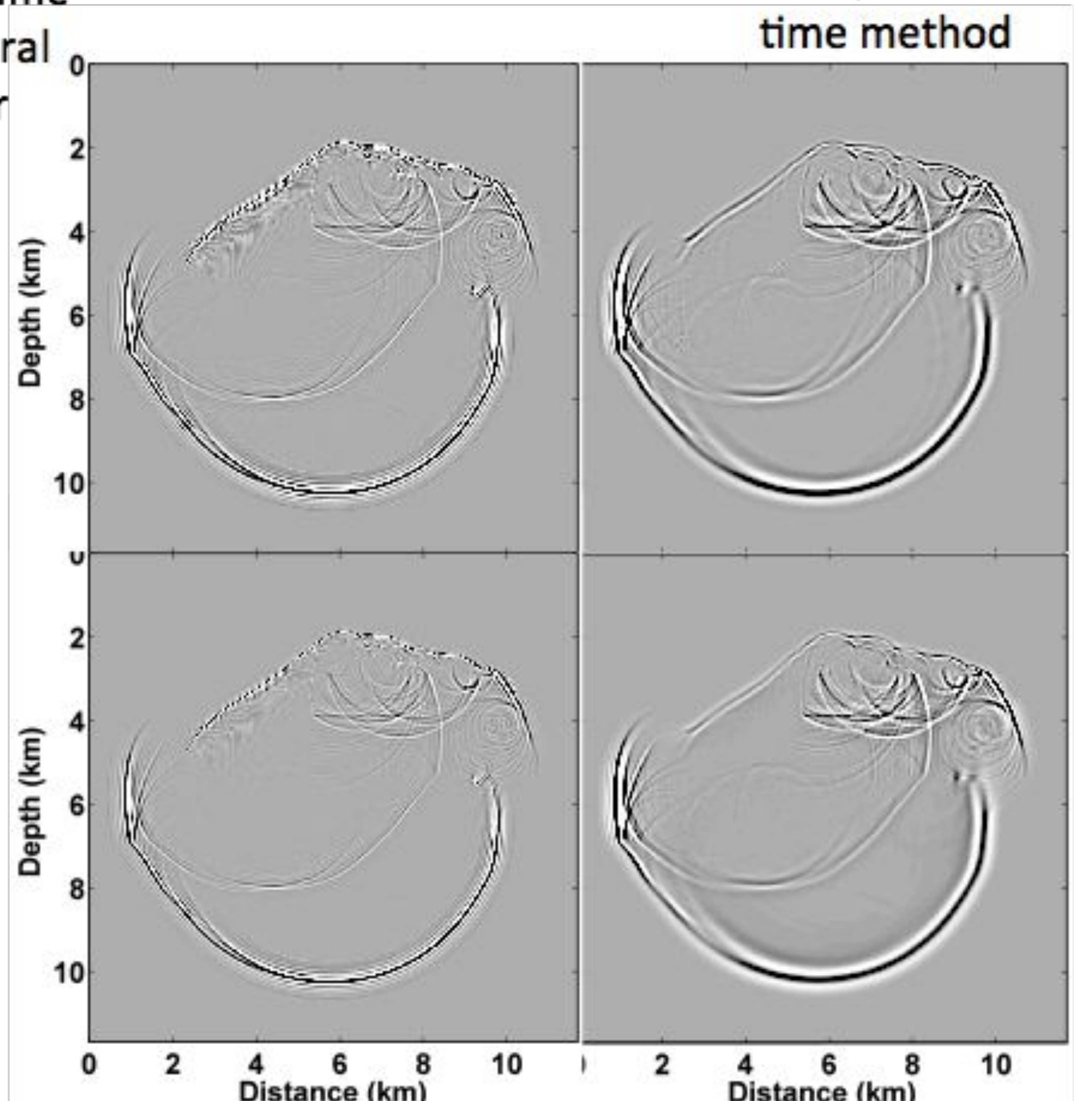


First-order Split step

Second-order split step

One way in time pseudospectral second order

Interpolation one-way in time method

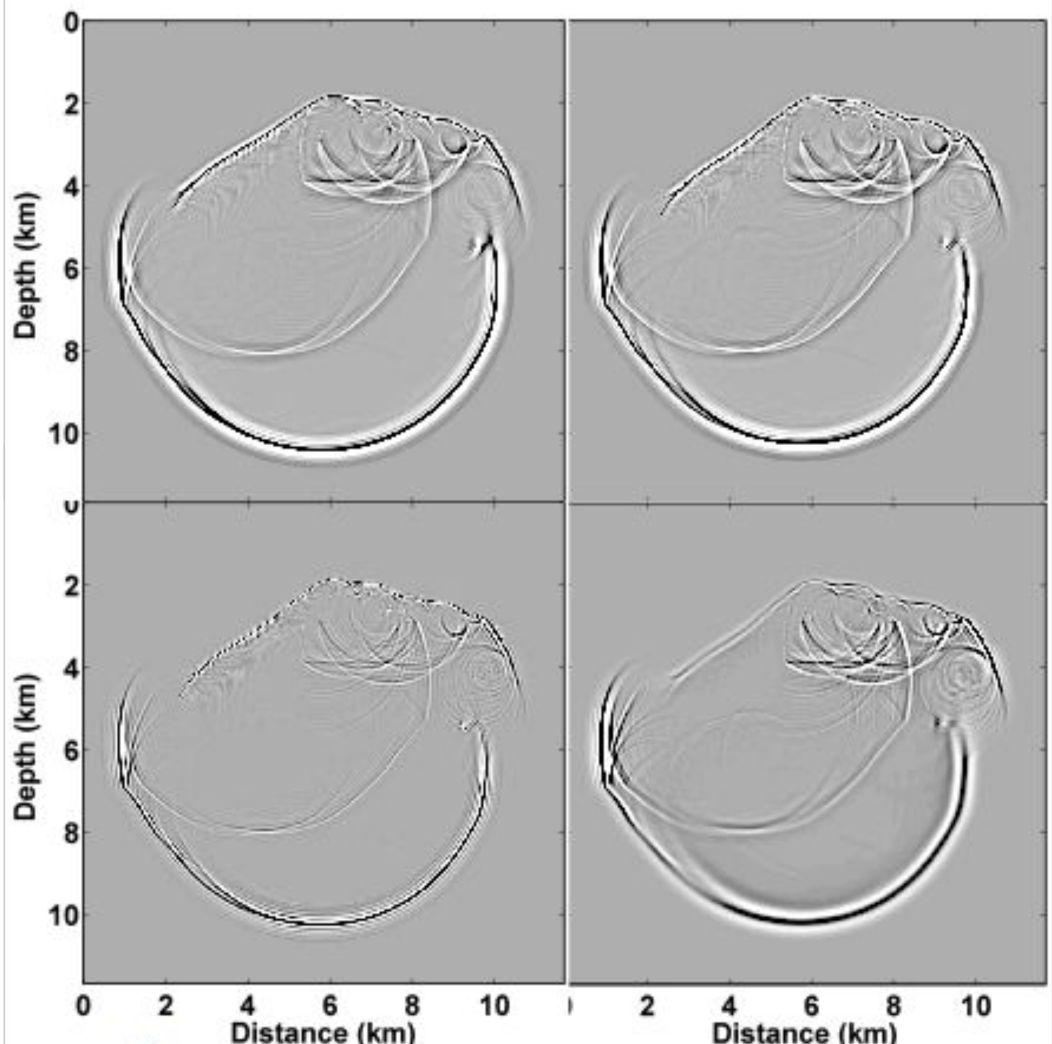


Evolutionary Pseudospectral second order

Evolutionary interpolation

Cosine Interpolation 2 vels

Cosine Interpolation 4 vels



Evolutionary pseudospectral second order

Evolutionary interpolation

	Time(s)	Timestep(ms)	# of FFTs per timestep
1(a) 2^{nd} Order pseudospectral two way	90	1.2	2
1(b) 4^{th} Order pseudospectral two way	113	1.5	3
1(c) Split-step first order	103	1.5	3
1(d) Split-step second order	130	1.5	4
2(a) One-way time pseudospectral	70	5	6
2(b) One-way time POU	38	9	4
2(c) Evolution equation pseudospectral	215	1.5	6
2(d) Evolution equation POU	88	9	8
3(b)Two way cosine , 2 velocities	104	1.9	3
3(c) Two way cosine , 4 velocities	188	1.9	5

Pseudo-acoustic

- Elastic wave equation is computationally expensive
- Elastic RTM imaging condition requires wavefield separation at each propagation step
- Alternatively, Propagate each wave mode (P,SV) separately
- TTI

Dispersion Relations For elastic

$$\det\left[c_{ijkl}n_j n_l - \rho V^2 \delta_{ik}\right] = 0$$

In TTI media

$$\omega^4 = \left[\left(v_{px}^2 + v_{sz}^2\right)\left(k_x^2 + k_y^2\right) + \left(v_{pz}^2 + v_{sz}^2\right)k_z^2\right]\omega^2$$
$$- v_{px}^2 v_{sz}^2 \left(k_x^2 + k_y^2\right)^2 - v_{pz}^2 v_{sz}^2 k_z^4$$
$$+ \left[v_{pz}^2\left(v_{pn}^2 - v_{px}^2\right) - v_{sz}^2\left(v_{pn}^2 + v_{pz}^2\right)\right]\left(k_x^2 + k_y^2\right)k_z^2$$

$$v_{pn} = v_{pz}\sqrt{1+2\delta}, v_{px} = v_{pz}\sqrt{1+2\varepsilon}$$

Fletcher, 2009

Dispersion Relations

- setting $v_{sz} = 0$

$$\omega^4 = \left[\left(v_{px}^2\right)\left(k_x^2 + k_y^2\right) + \left(v_{pz}^2\right)k_x^2\right]\omega^2$$
$$+\left[v_{pz}^2\left(v_{pn}^2 - v_{px}^2\right)\right]\left(k_x^2 + k_y^2\right)k_z^2$$

- Can create a couple system of PDEs to solve the pseudo-acoustic equation.
- There are still two solutions of the dispersion relation so still shear wave propagation.

Solving the square root

$$\omega_{v_p}^2 = v_{P_z}^2 (k^2 (1 - \frac{f}{2}) +$$

$$\varepsilon k_x^2 + \frac{f}{2}\sqrt{k^4 + \frac{4k_x^2}{f}\left(2\delta k_z^2 - \varepsilon(k_z^2 - k_x^2)\right) + \frac{4\varepsilon^2 k_x^4}{f^2}})$$

$$f = 1 - \frac{v_{S_0}^2}{v_{P_0}^2}$$

Tsvankin, 2001

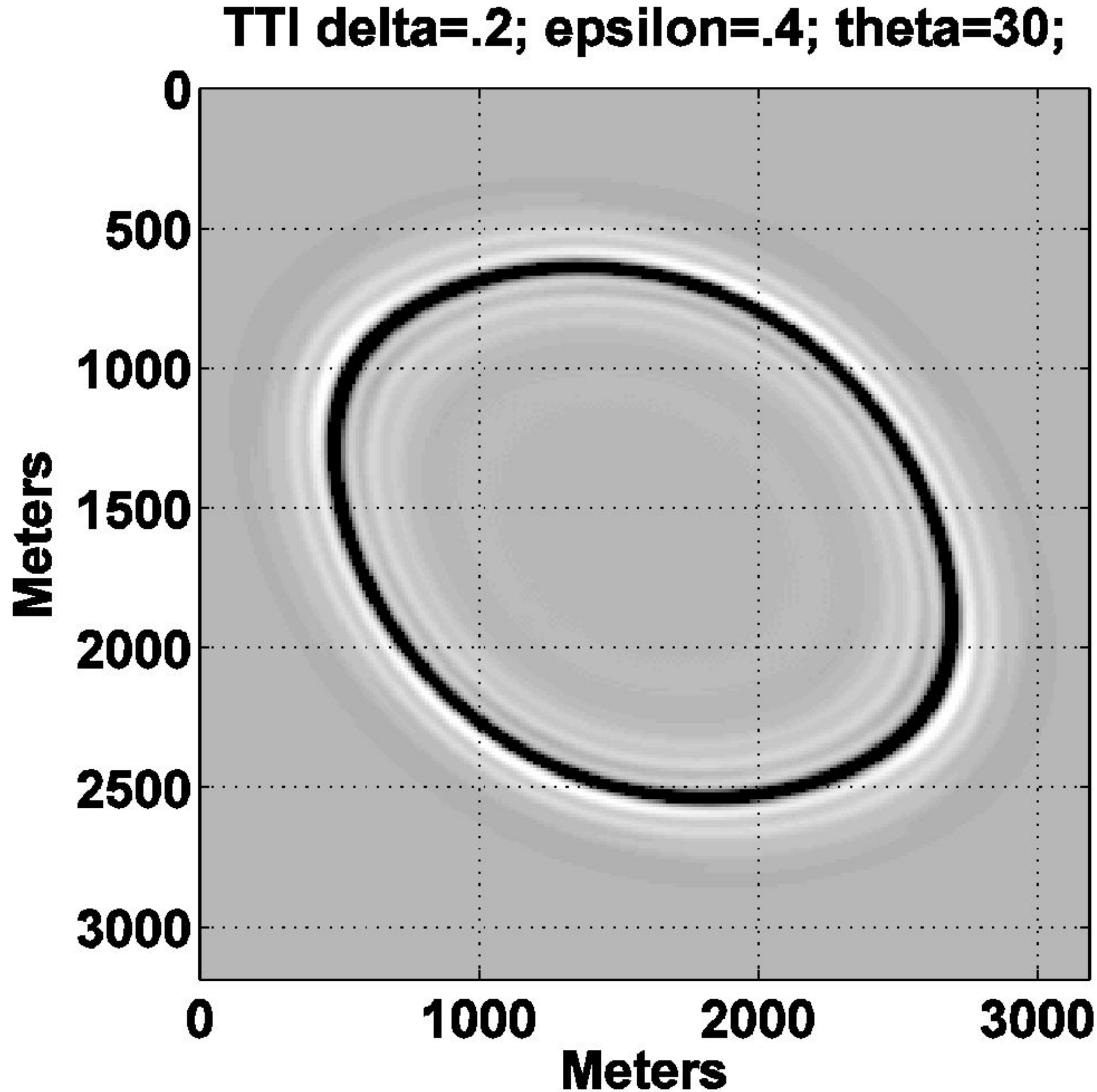
Weak anisotropy approximation

$$\omega_{v_p}^2 \cong v_x^2 k_x^2 + v_n^2 k_z^2 + \left(v_n^2 - v_x^2\right)\frac{k_z^2 k_z^2}{k_x^2 + k_z^2}$$

Pseudo-acoustic wave propagation for any dispersion relationship

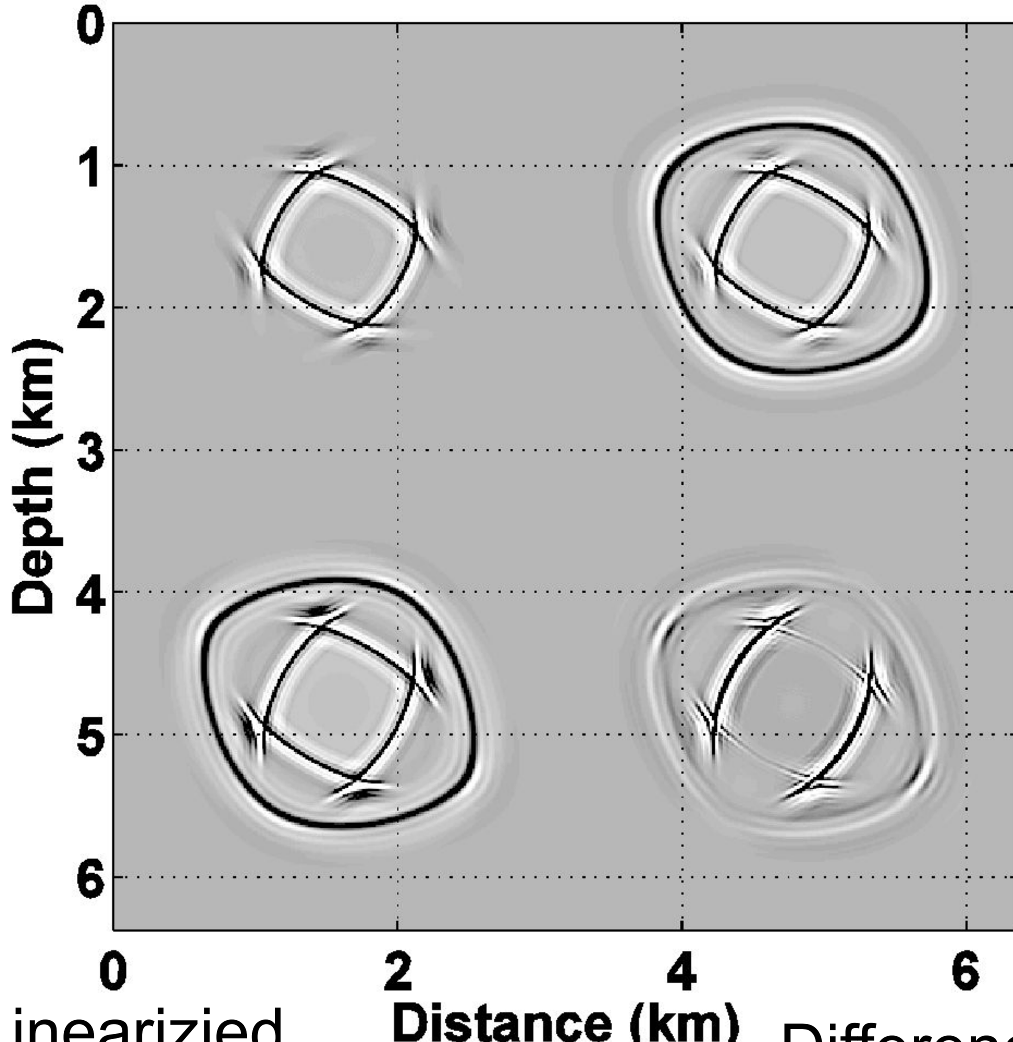
$$\frac{\partial^2 U}{\partial t^2} = -FT_{\vec{k}}^{-1}\left\{\omega^2(\vec{k},\vec{x})FT\left\{U(t,\vec{x})\right\}\right\}$$

$$U(\Delta t,\vec{x}) = U(-\Delta t,\vec{x}) +$$
$$2FT_{\vec{k}}^{-1}\left\{\cos\left(2\pi\omega(\vec{k},\vec{x})\Delta t\right)FT_{\vec{x}}\left\{U(0,\vec{x})\right\}\right\}$$


TTI delta=.2; epsilon=.4; theta=30;
0
500
1000
1500
2000
2500
3000
Meters
0
1000
2000
3000
Meters

S-wave Exact
dispersion relation

P&S-wave Exact
dispersion relation

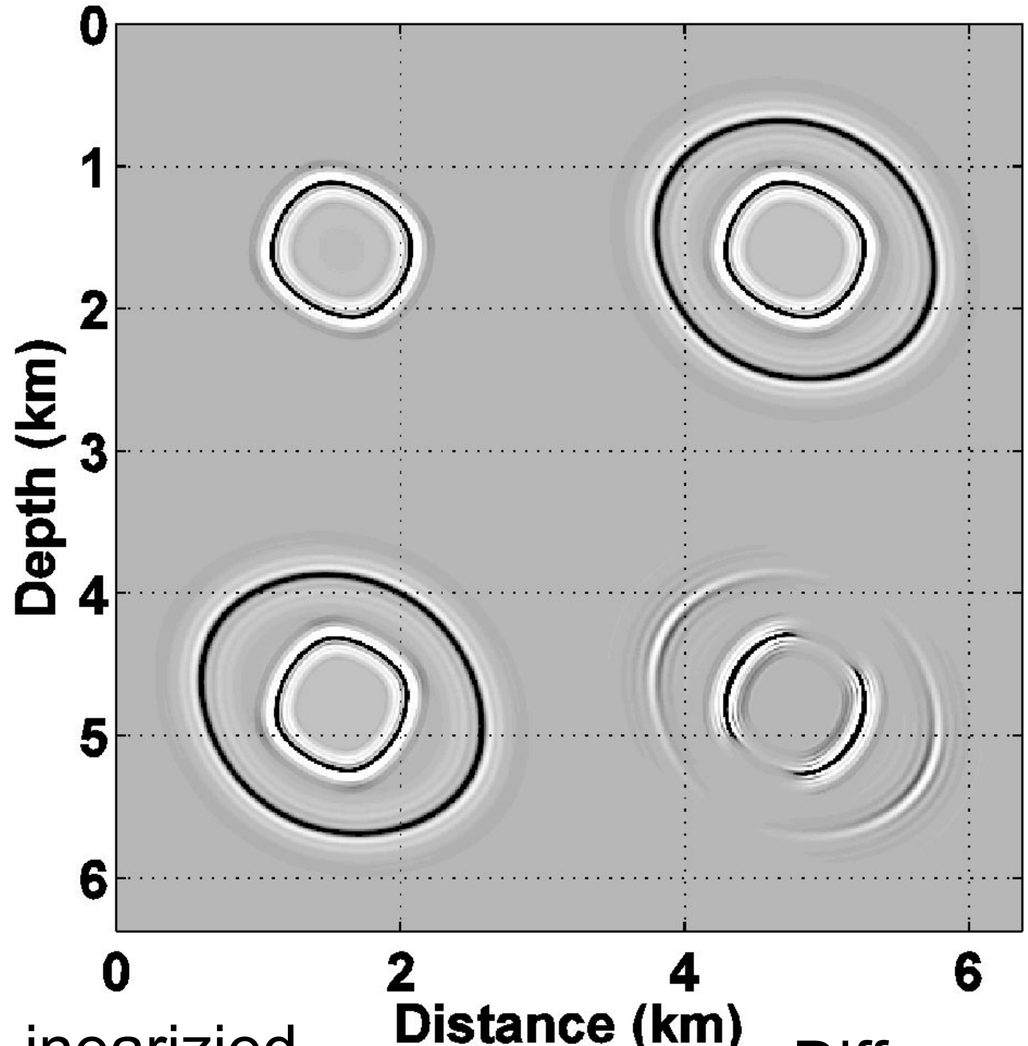


P&S-wave Linearizied
dispersion relation

Difference between
Linearizied & exact

S-wave Exact dispersion relation

P&S-wave Exact dispersion relation



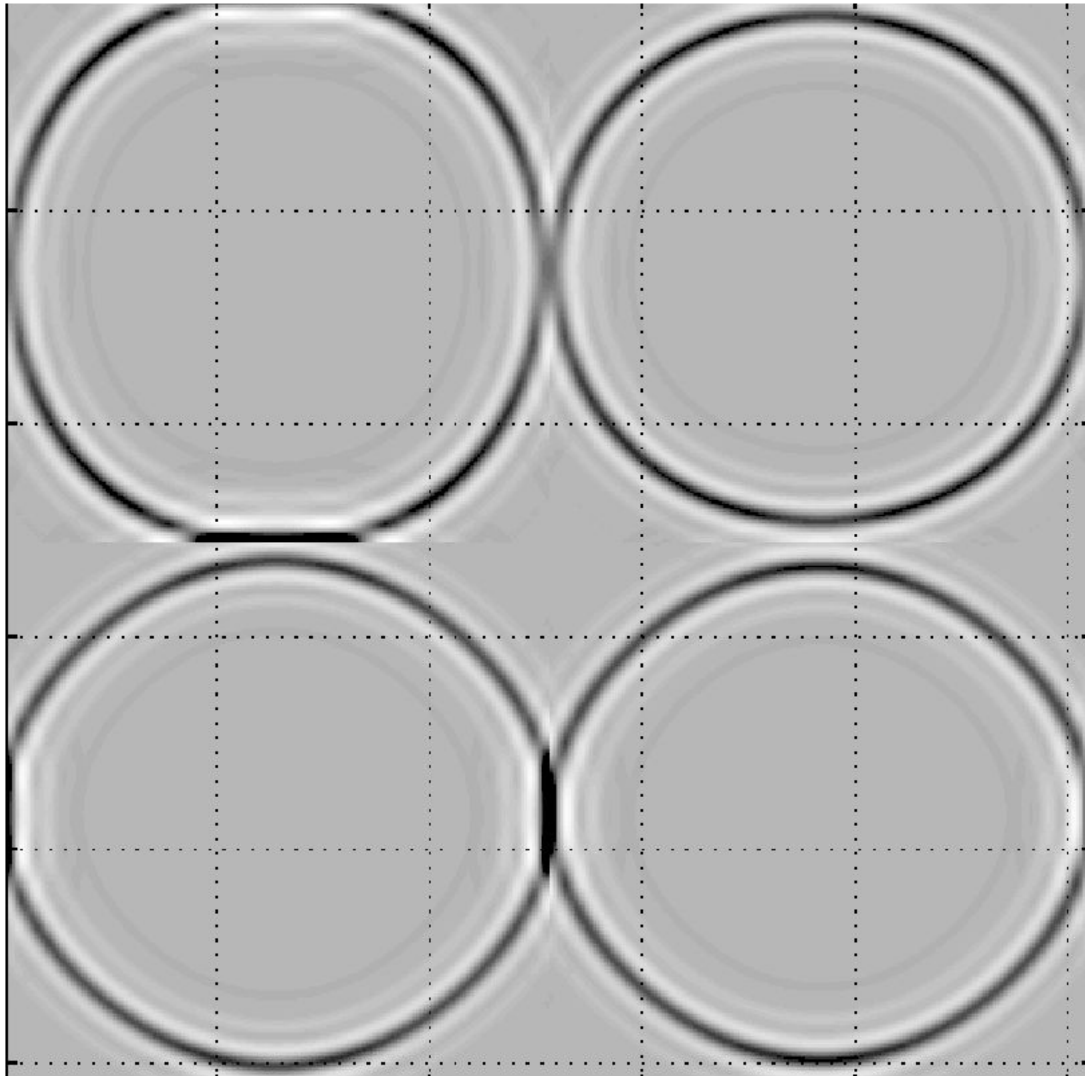
P&S-wave Linearizied dispersion relation

Difference between Linearizied & exact

P Wave 3D Orthorhombic

XZ plane

XZ plane



XY plane

XY plane-280m

Cost

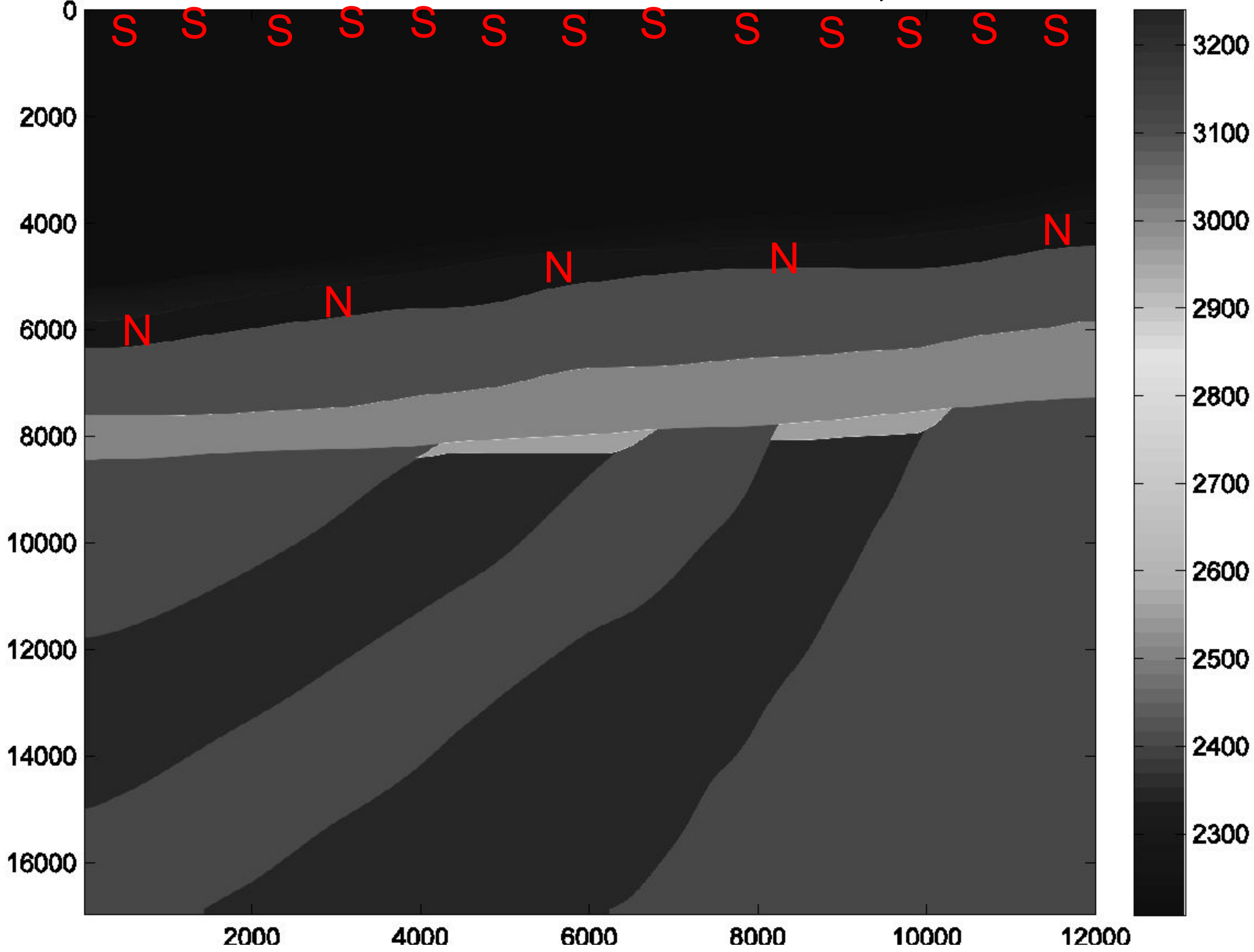
	PP	PS
Max vel	5500m/s	2500m/s
Min vel	1500m/s	500m/s
fmax	55Hz	55Hz
dx	12.5m	4m
dt		
Computation cost	1	3^4 (3D)

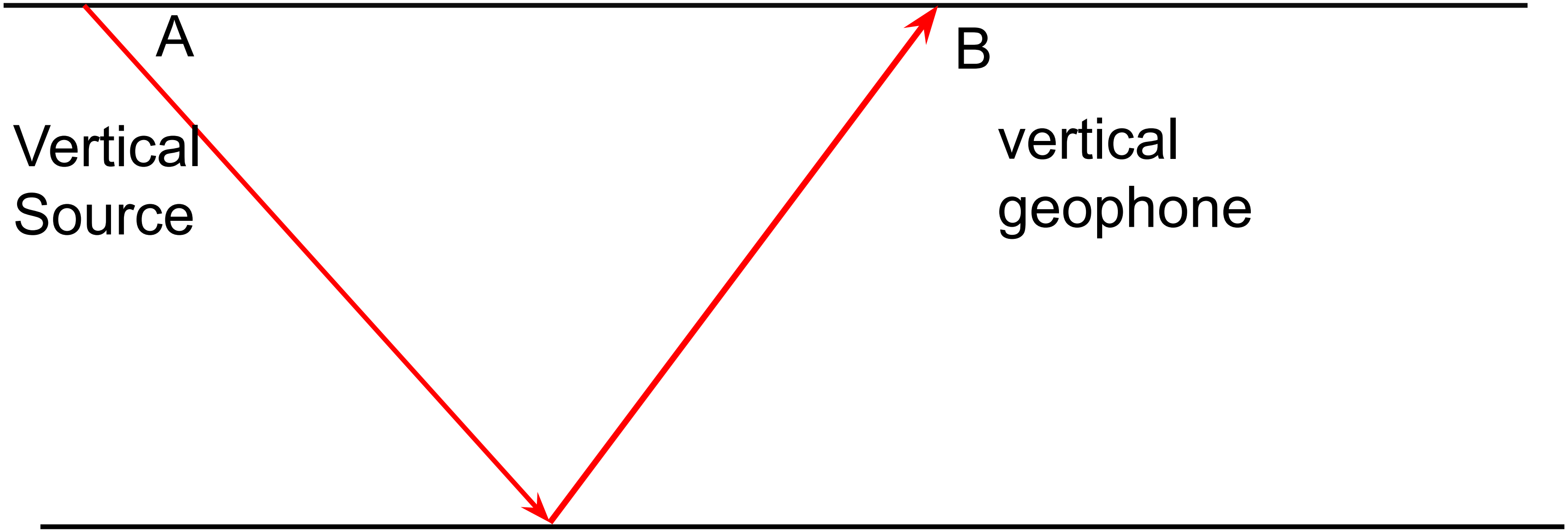
$$\lambda_{\min} = v_{\min} / f_{\max}$$

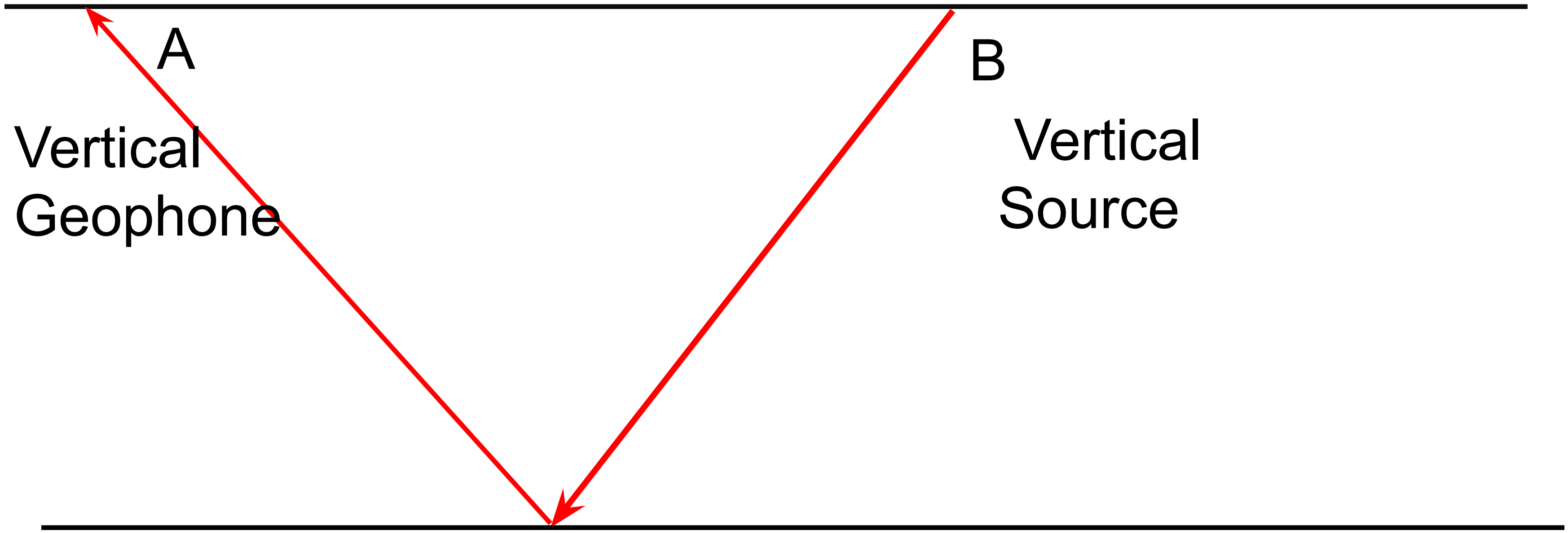
+ 2 samples/ wavelength

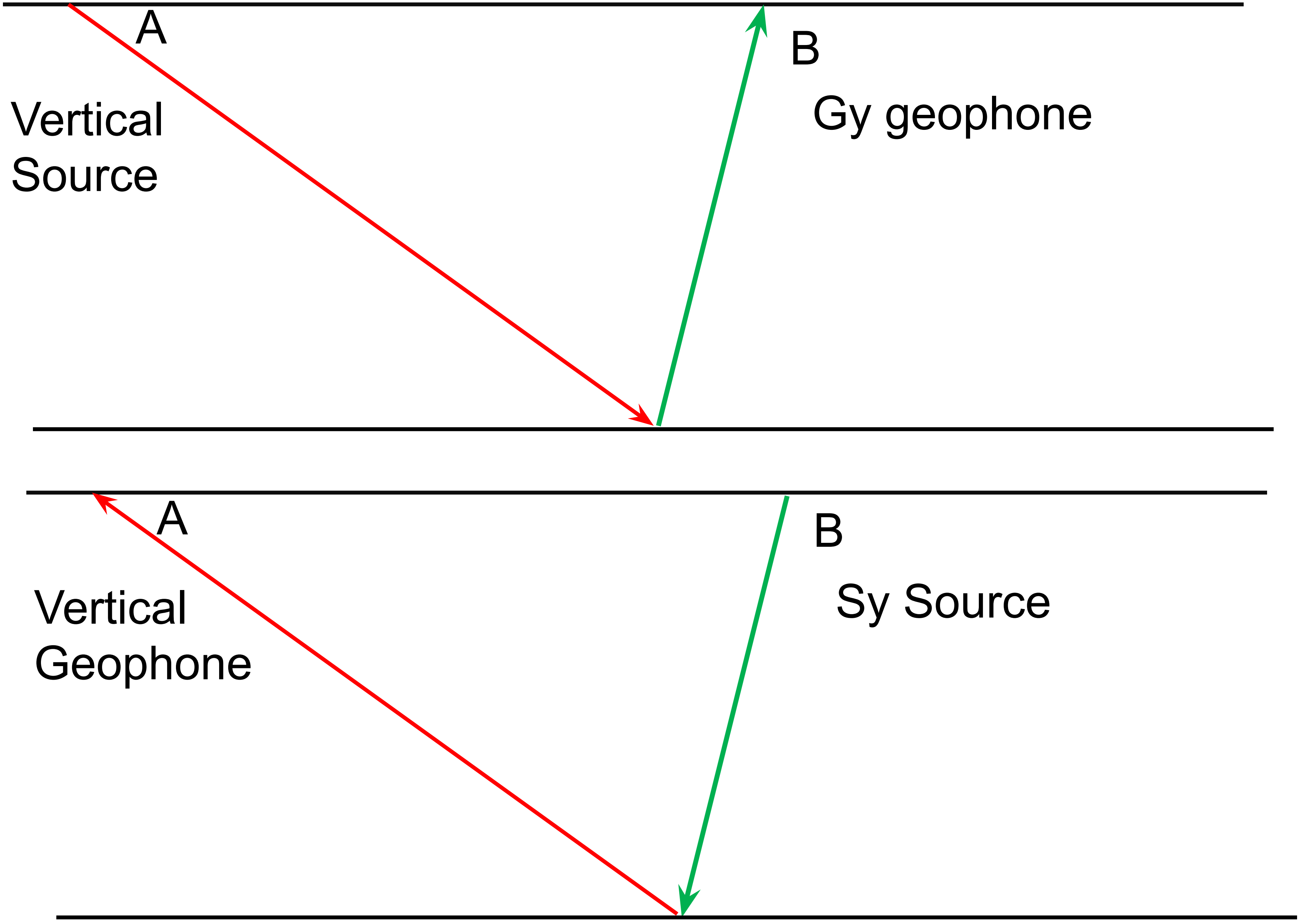
$$\frac{c\Delta t}{\Delta x} < \frac{1}{\sqrt{2}}$$

OBS Geometry, dx shot=12.5m dx node=100m,

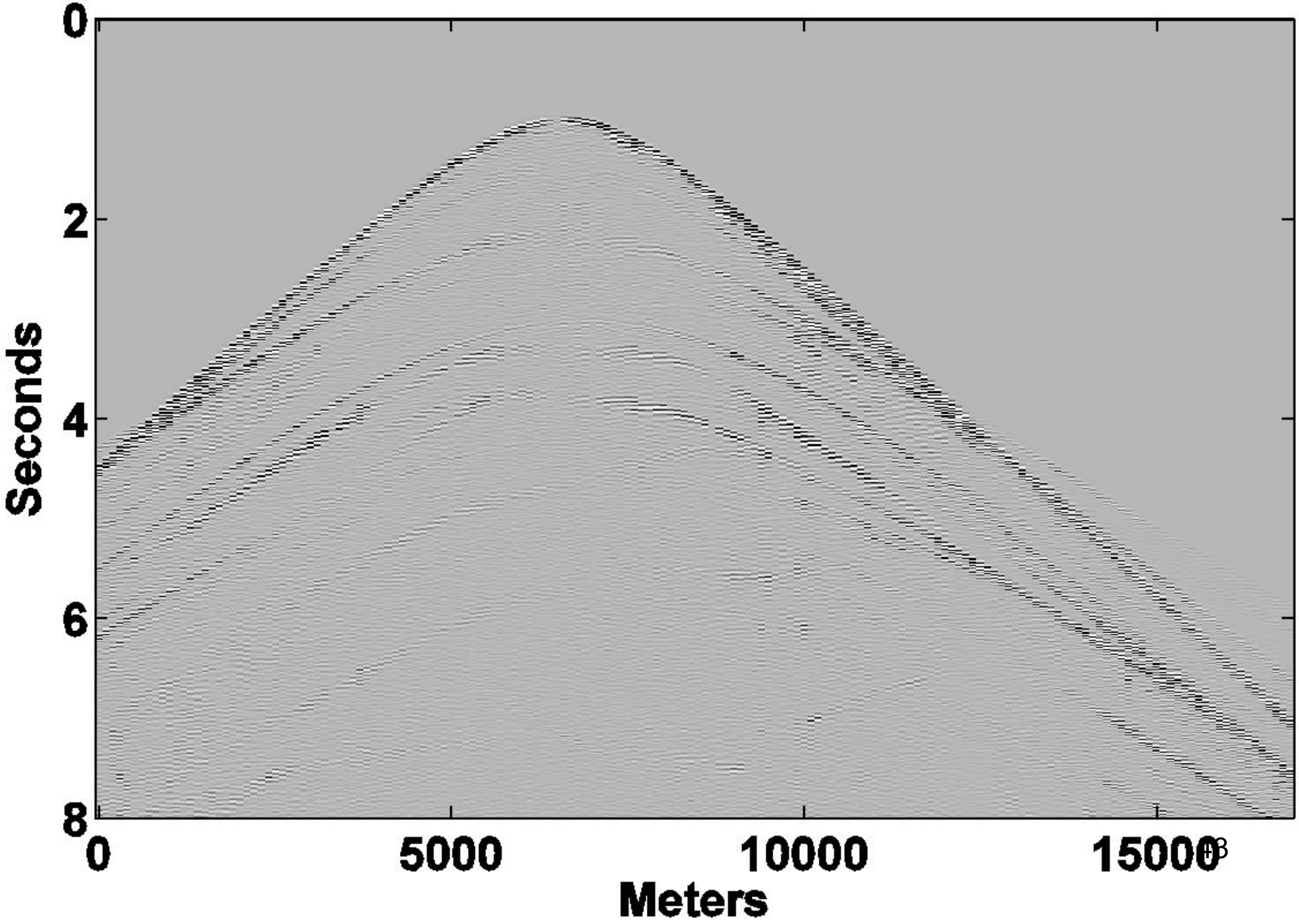



A
B
Vertical
Source
vertical
geophone

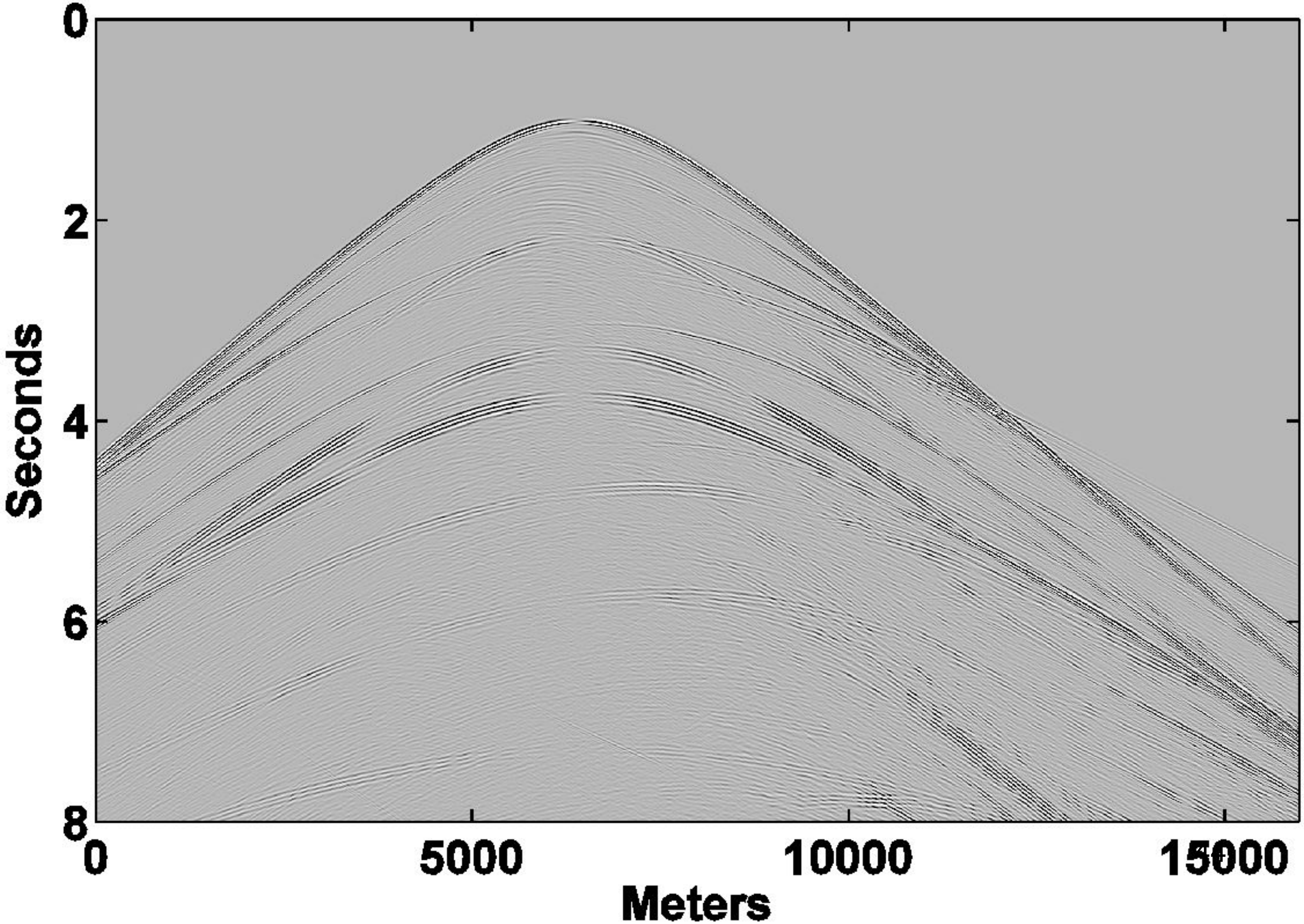

A
B
Vertical
Geophone
Vertical
Source


A
B
Vertical
Source
Gy geophone
A
B
Vertical
Geophone
Sy Source

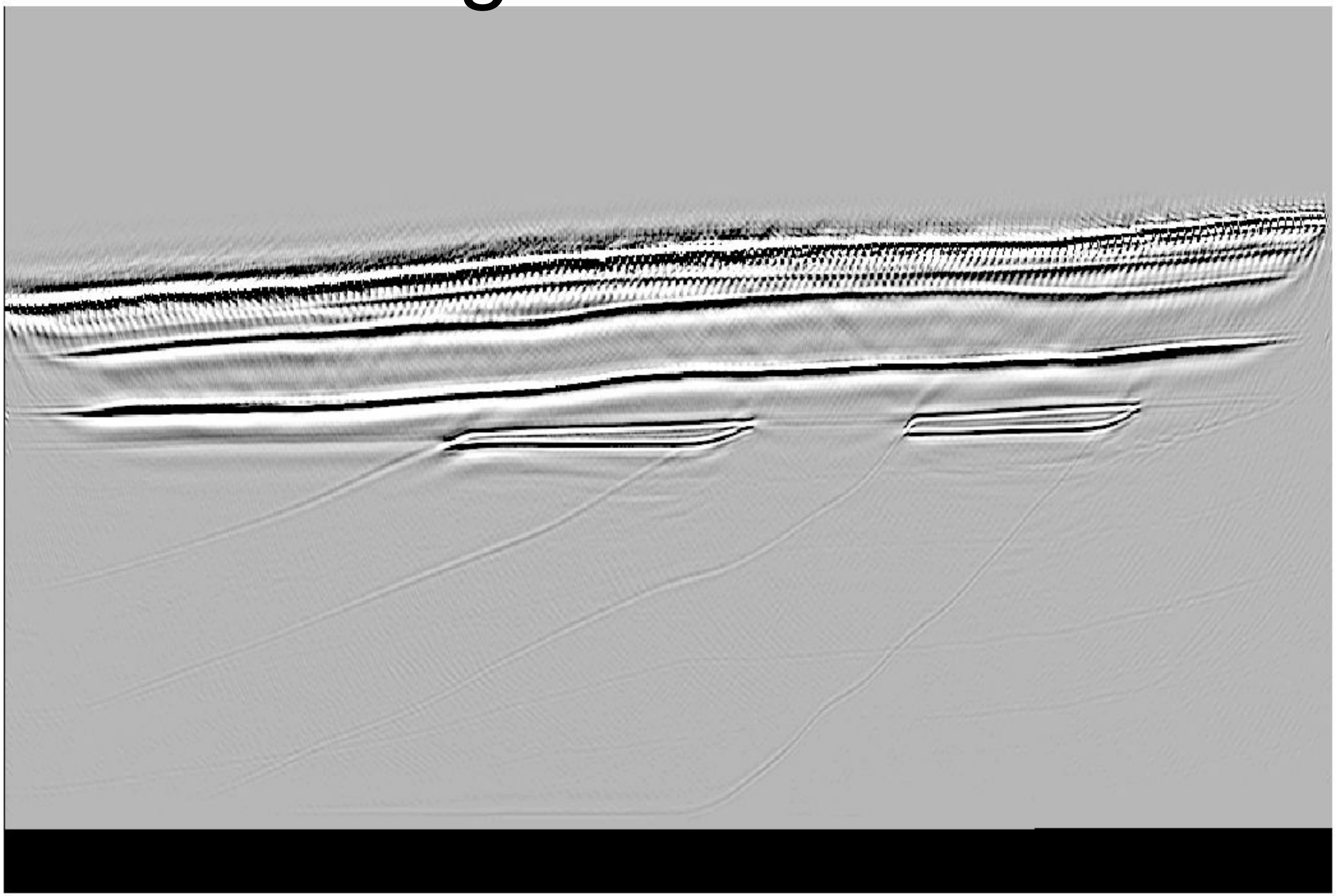
Common Shot Gather



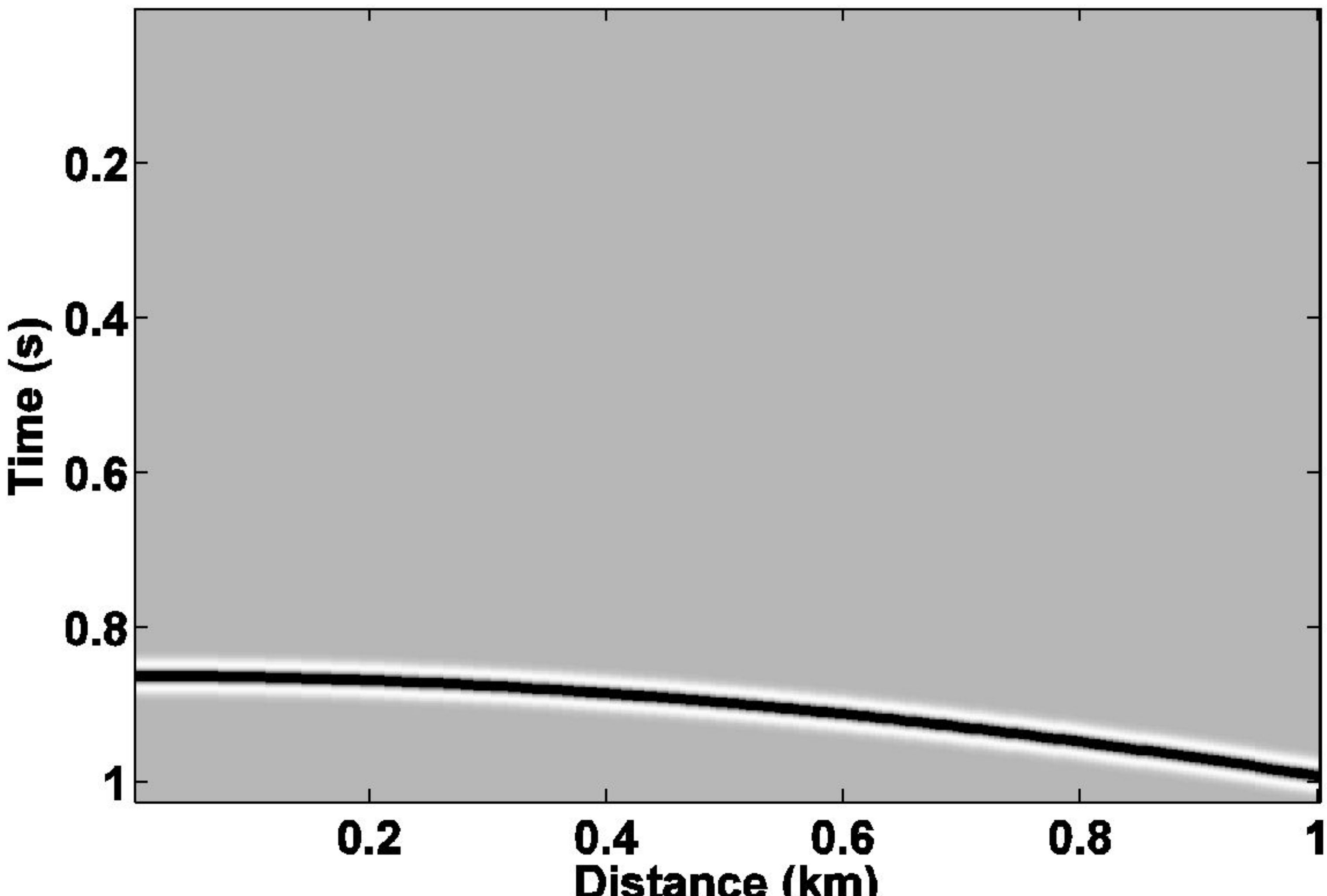
Common Node Gather



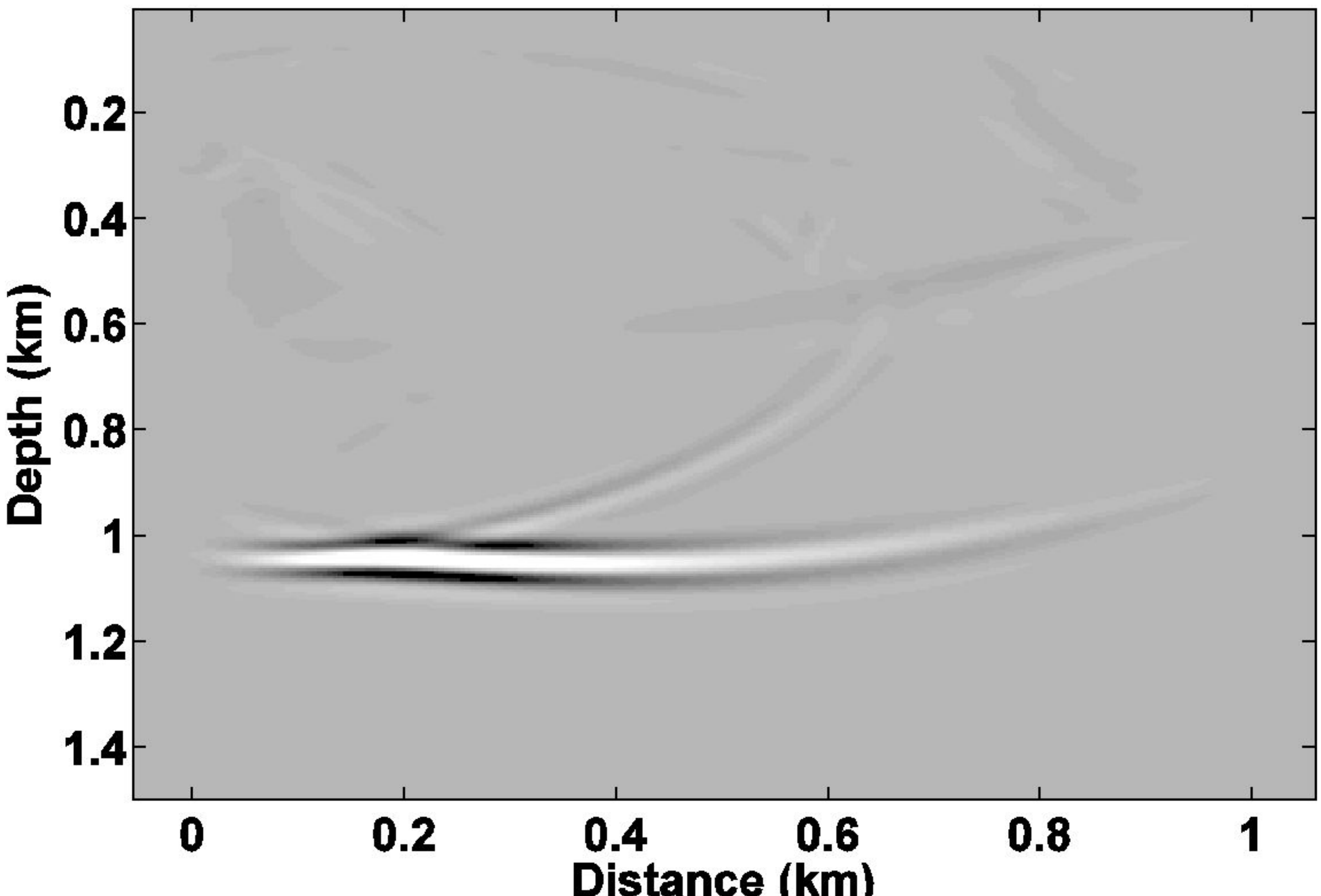
Migrated PP



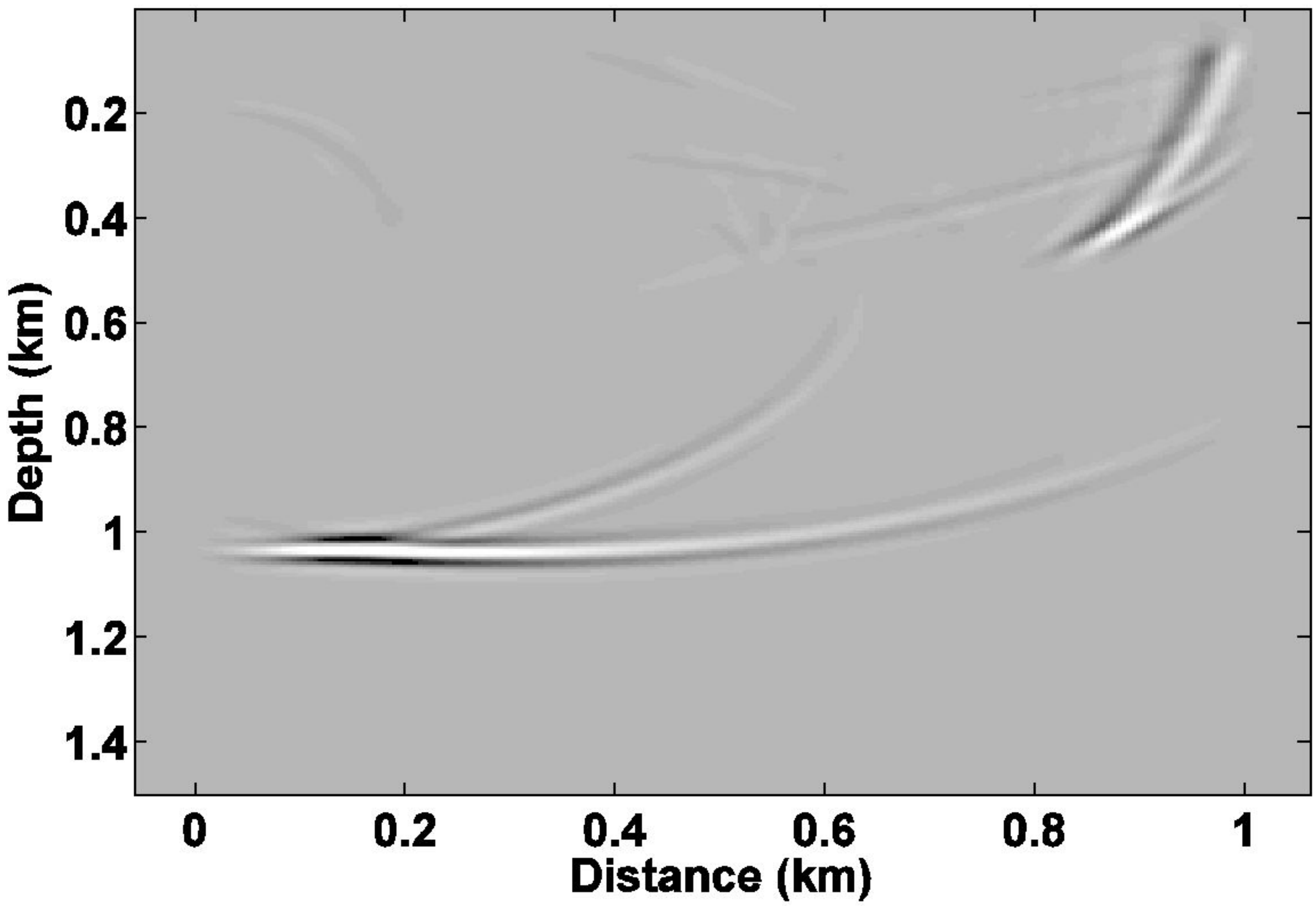
OBC PS Node gather



Migrated single shot PP



Migrated single shot PS



Conclusions

- Pseudospectral Methods are very similar to interpolation of Phase-shift methods.
- Evolutionary sol'n can take larger timesteps but have more complicated initial conditions and also requires more FFTs
- Evolutionary and one-way in time didn't give accurate results
- Phase-shift methods can more easily approximate anisotropic media

Acknowledgements

- CREWES sponsors
- POTSI, University of Calgary
- NSERC
- Coauthors: Gary Margrave, Michael Lamoureux