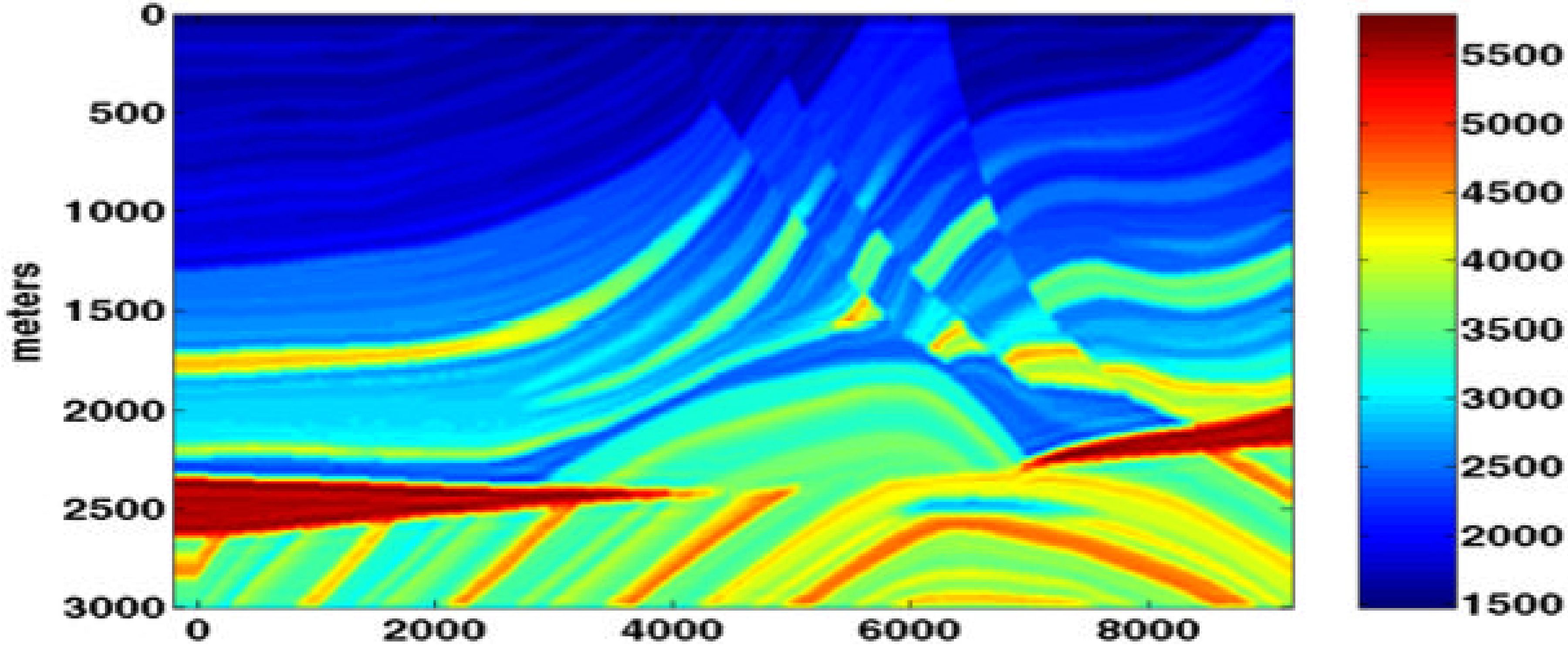
Two new explicit depth migration schemes that honour local velocity gradients

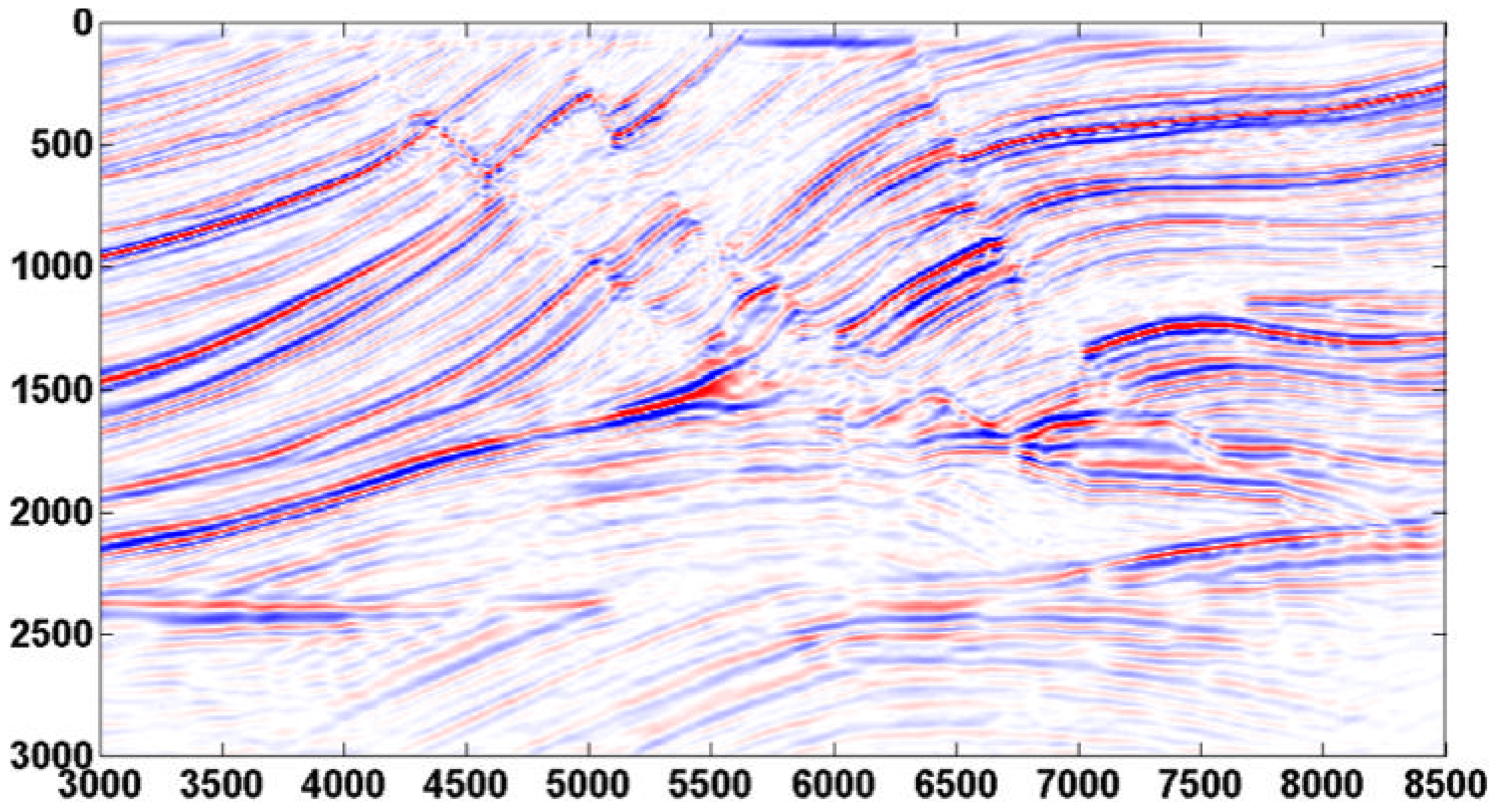
Chad Hogan and Gary Margrave

CREWES, University of Calgary

Seismic imaging



Seismic imaging



- A typical CREWES Marmousi seismic image

Seismic imaging

- Can we improve?

Seismic imaging

- Can we improve?
- Better images....

Seismic imaging

- Can we improve?
- Better images....
- Faster calculation....

CREWES

Seismic imaging

- Can we improve?
- Better images....
- Faster calculation....
- Better: horizontal velocity gradient (The Stolk operator in GPSPI migration).

Seismic imaging

- Can we improve?
- Better images....
- Faster calculation....
- Better: horizontal velocity gradient (The Stolk operator in GPSPI migration).
- Faster: vertical velocity gradient (Stabilizing explicit $\omega - x$ migration using local WKBJ operators).

GPSPI

$\Psi(x, z, \omega)$ is the (frequency) data that would be recorded by a geophone at point (x, z)

$$\Psi(x, z = \Delta z, \omega) = \mathbf{T}_\alpha \Psi(x, z = 0, \omega)$$

$\mathbf{T}_\alpha$ is a *wavefield extrapolation operator*.

GPSPI

$$\Psi(x, z = \Delta z, \omega) = \mathcal{F}^{-1}\left[\alpha\left(v(x), k_x, \omega\right)\mathcal{F}\left[\Psi(x, z = 0, \omega)\right]\right]$$

where

$$\alpha\left(v\left(x\right), k_x, \omega\right) = \begin{cases} e^{i\Delta z k_z(x)}, & |k_x| \leq \frac{\omega}{v(x_0)} \\ e^{-|\Delta z k_z(x)|}, & |k_x| > \frac{\omega}{v(x_0)} \end{cases}$$

$$k_z(x) = \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2}$$

GPSPI

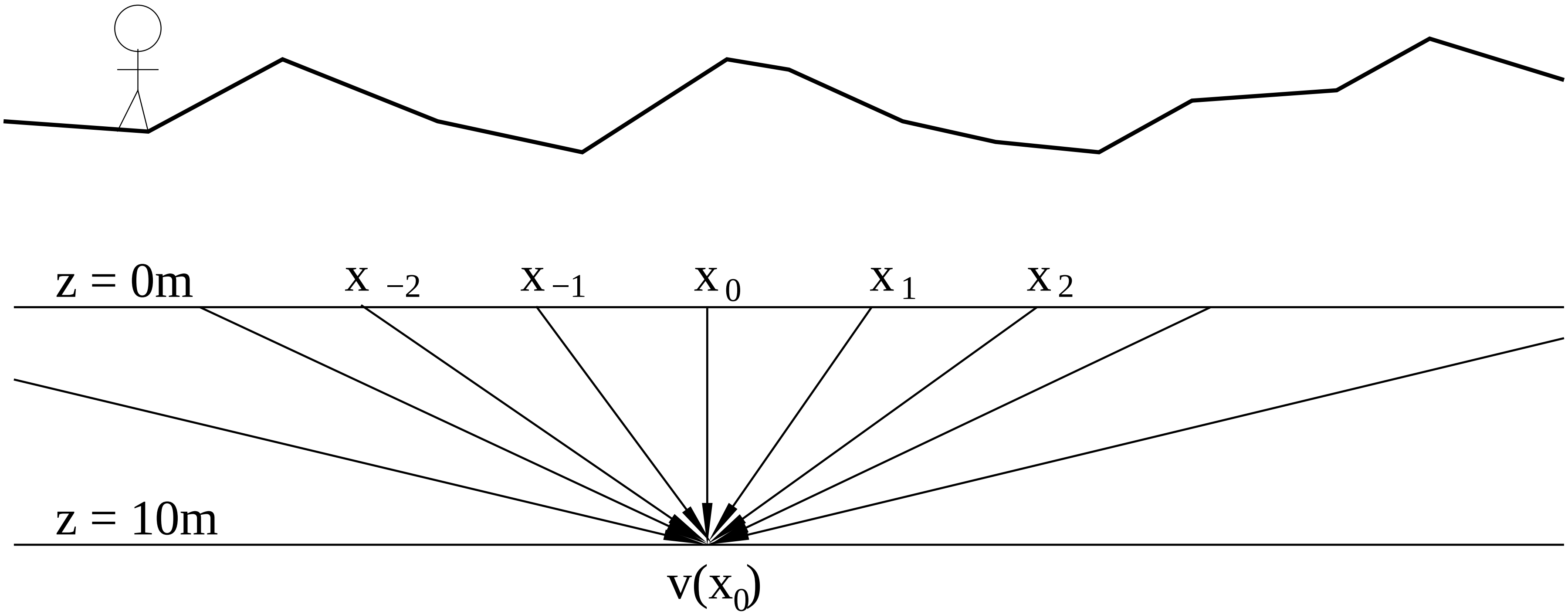
$$\Psi(x, z = \Delta z, \omega) = \mathcal{F}^{-1}\left[\alpha\left(v(x), k_x, \omega\right)\mathcal{F}\left[\Psi(x, z = 0, \omega)\right]\right]$$

where

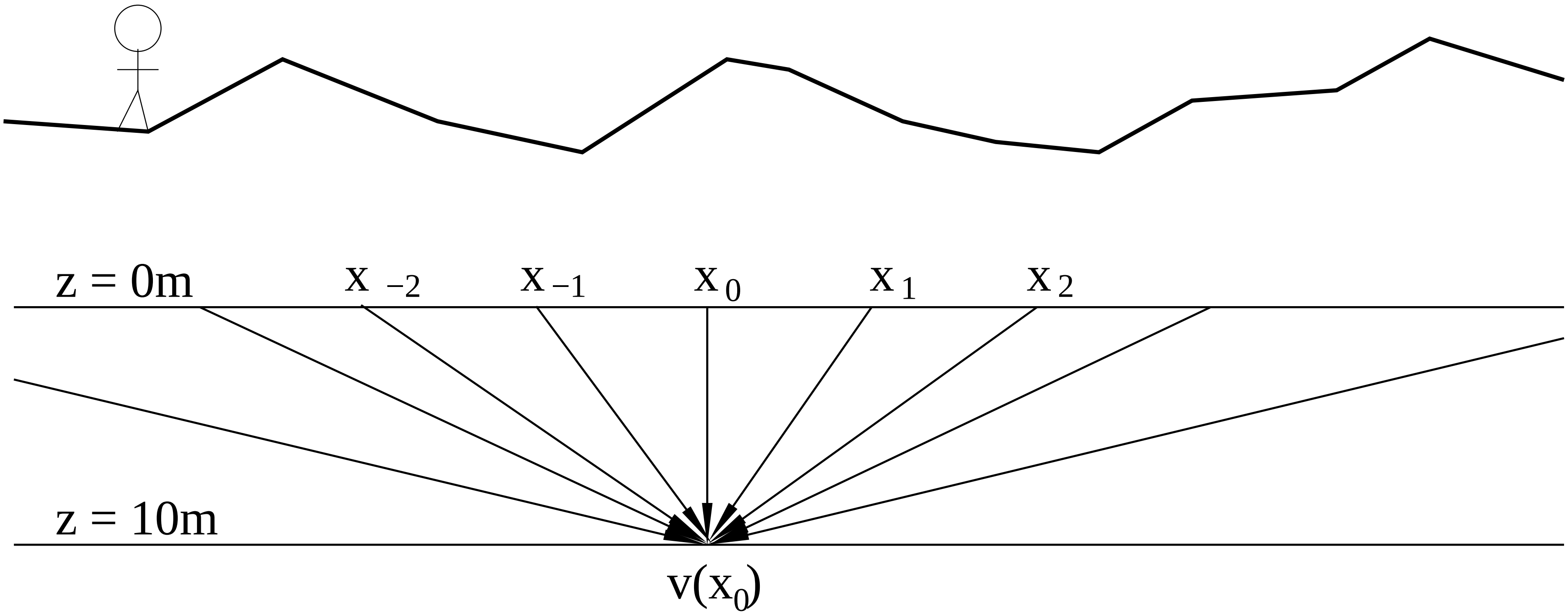
$$\alpha\left(v\left(x\right), k_x, \omega\right) = \begin{cases} e^{i\Delta z k_z(x)}, & |k_x| \leq \frac{\omega}{v(x_0)} \\ e^{-|\Delta z k_z(x)|}, & |k_x| > \frac{\omega}{v(x_0)} \end{cases}$$

$$k_z(x) = \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2}$$

Making better images



Making better images



- We use the local output velocity. What about a horizontal derivative?

The Stolk correction

$$k_z(x) = \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2}$$

The Stolk correction

$$k_z(x) = \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2}$$

$$+ \frac{ik_x\omega^2}{2v(x_0)} \left(\frac{\partial}{\partial x} s(x_0)\right) \left(\frac{\omega^2}{v(x_0)^2} - k_x^2\right)^{-3/2}$$

CREWES

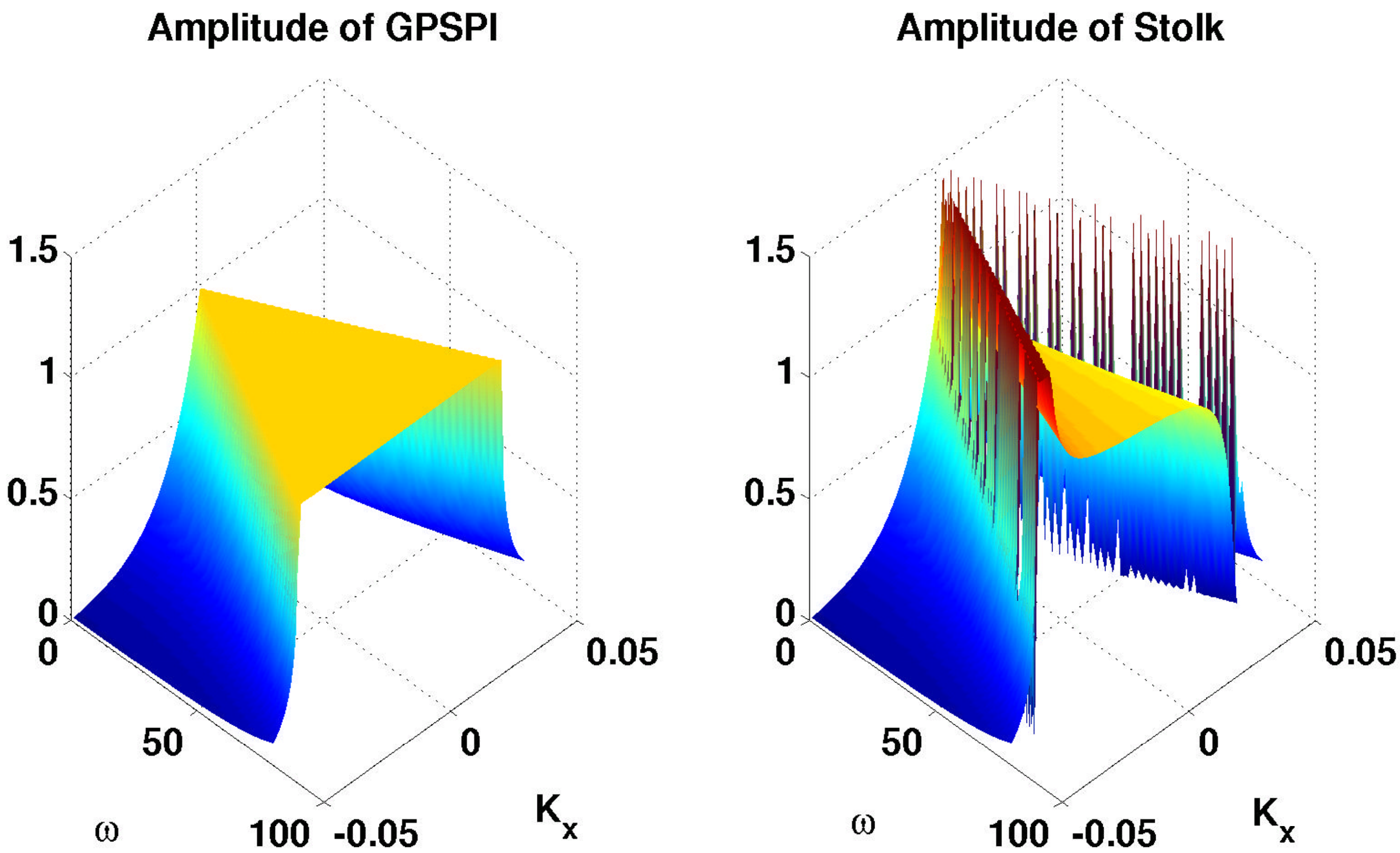
The Stolk correction

$$k_z(x) = \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2}$$

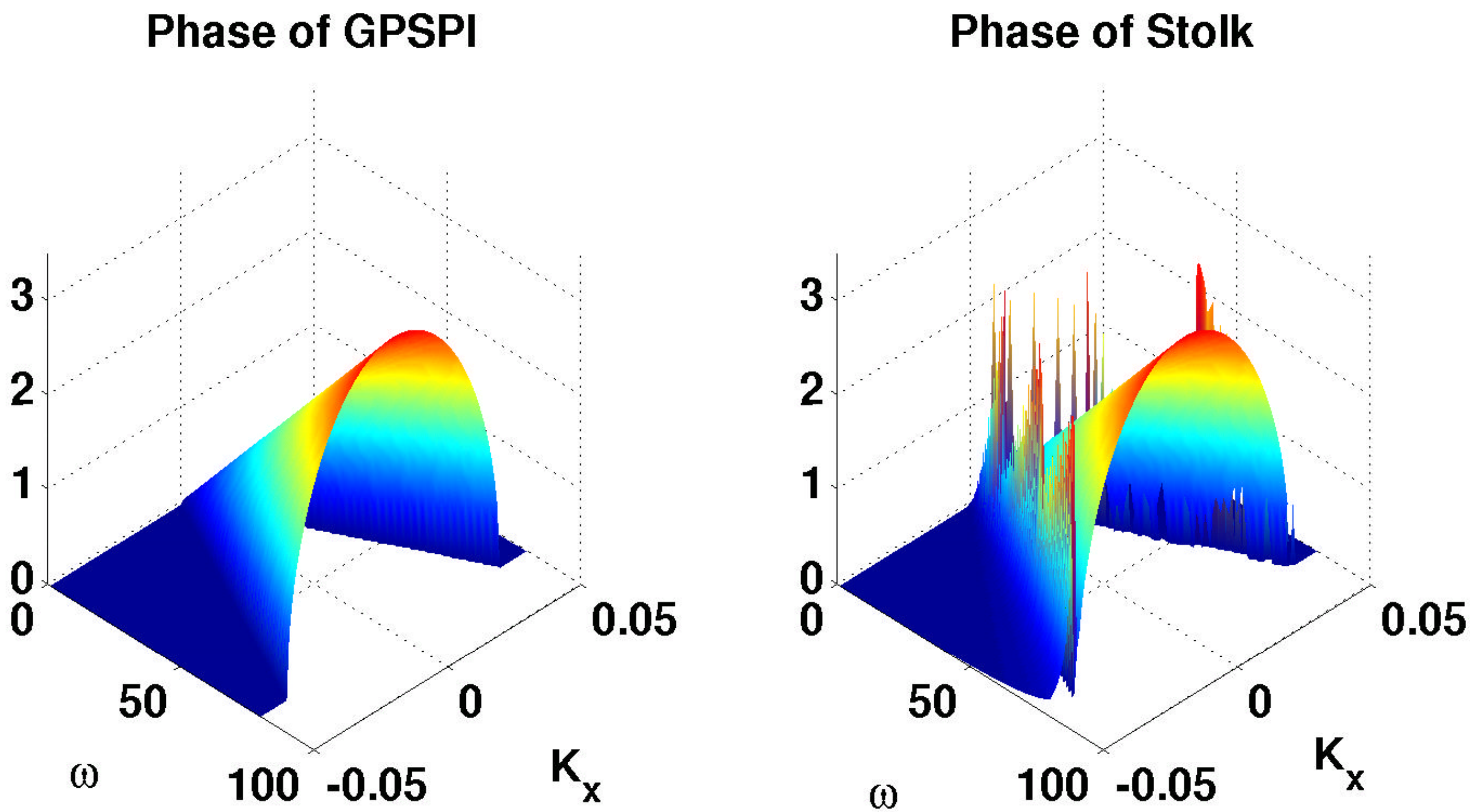
$$+ \frac{ik_x\omega^2}{2v(x_0)} \left(\frac{\partial}{\partial x} s(x_0)\right) \left(\frac{\omega^2}{v(x_0)^2} - k_x^2\right)^{-3/2}$$

CREWES

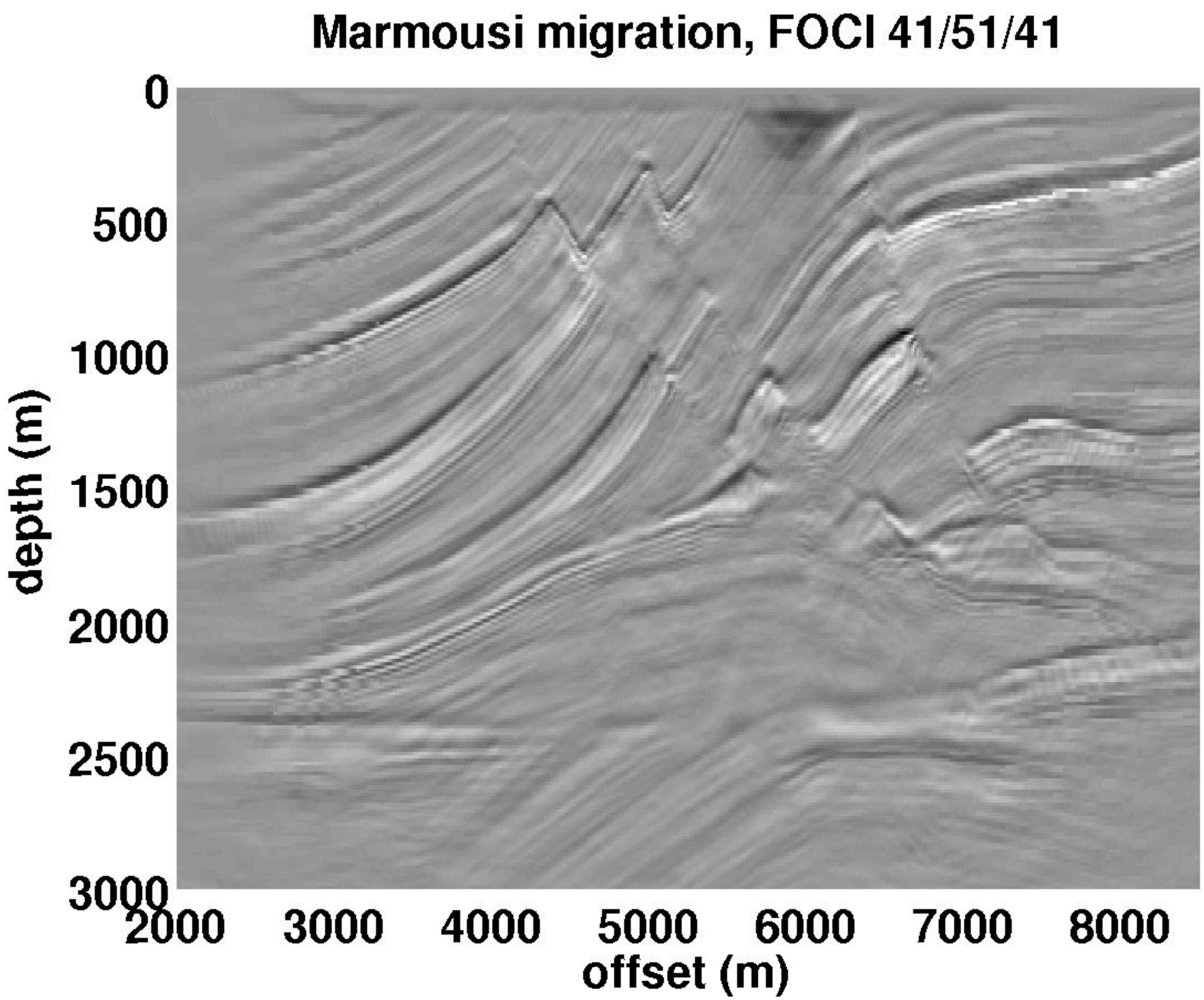
GPSPI and Stolk



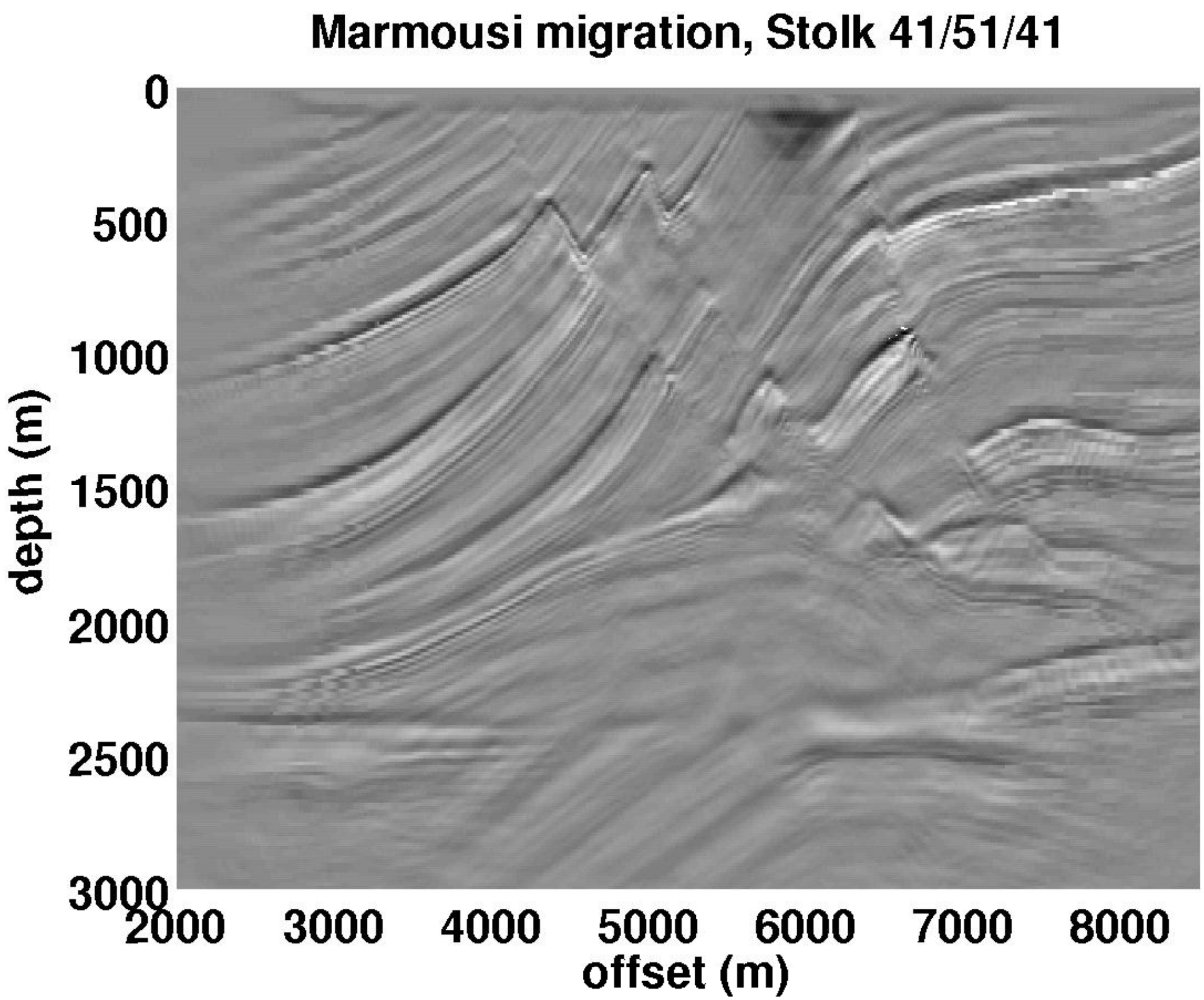
GPSPI and Stolk



GPSPI migration



Stolk migration



Stolk migration

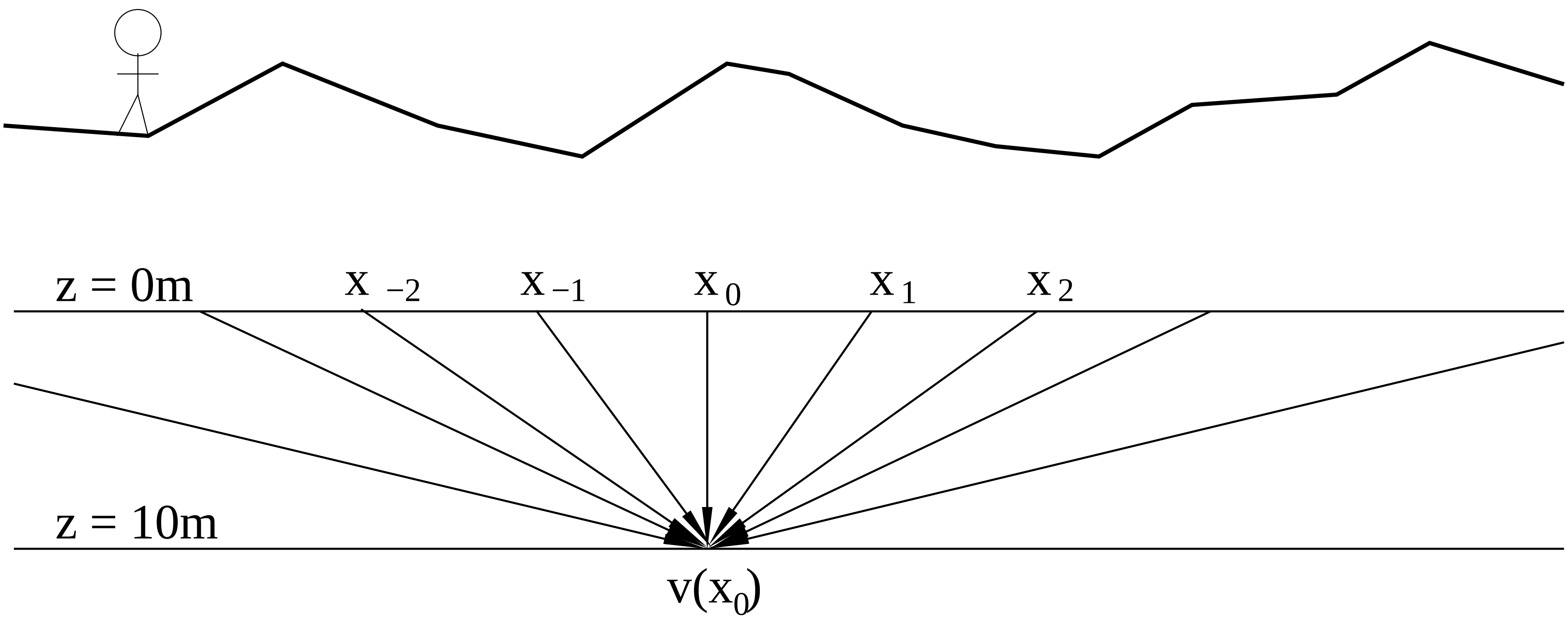
- Not much of an improvement (if any).
- Unfortunately, it takes at least twice as long to run.
- High-frequency correction maybe makes Marmousi less than the ideal candidate.

What about faster?

- FOCI is fast, but it requires stabilization. This stabilization is very difficult in 3D.

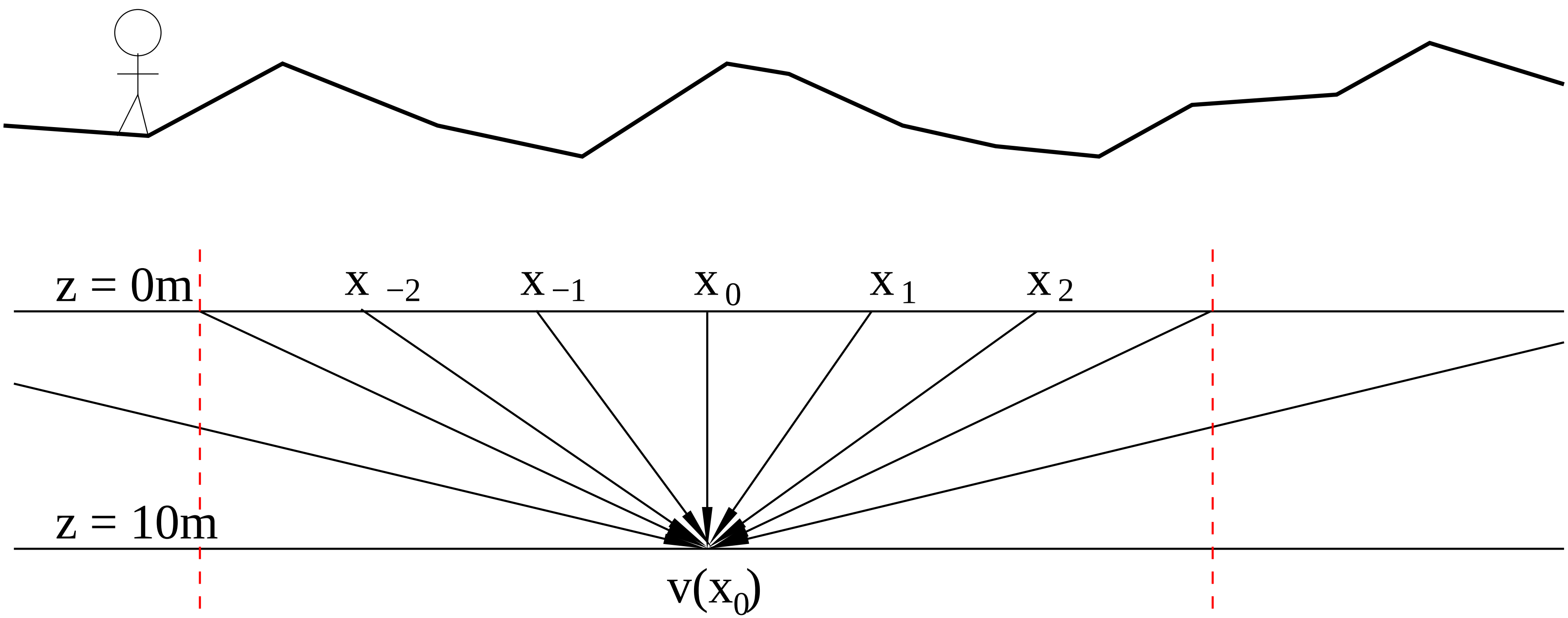
What about faster?

- FOCI is fast, but it requires stabilization. This stabilization is very difficult in 3D.



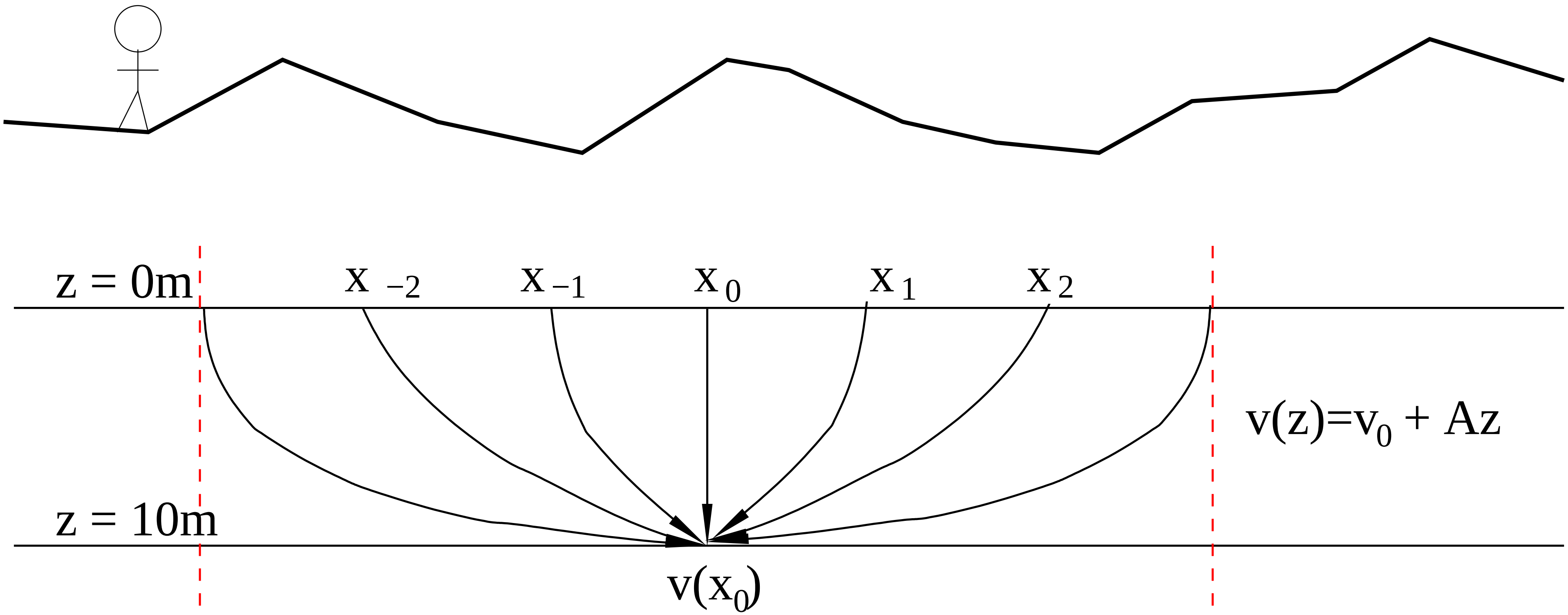
What about faster?

- FOCI is fast, but it requires stabilization. This stabilization is very difficult in 3D.



What about faster?

- FOCI is fast, but it requires stabilization. This stabilization is very difficult in 3D.



V(z)

$$\alpha = \exp\left(i\Delta z \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2} \right)$$

CREWES

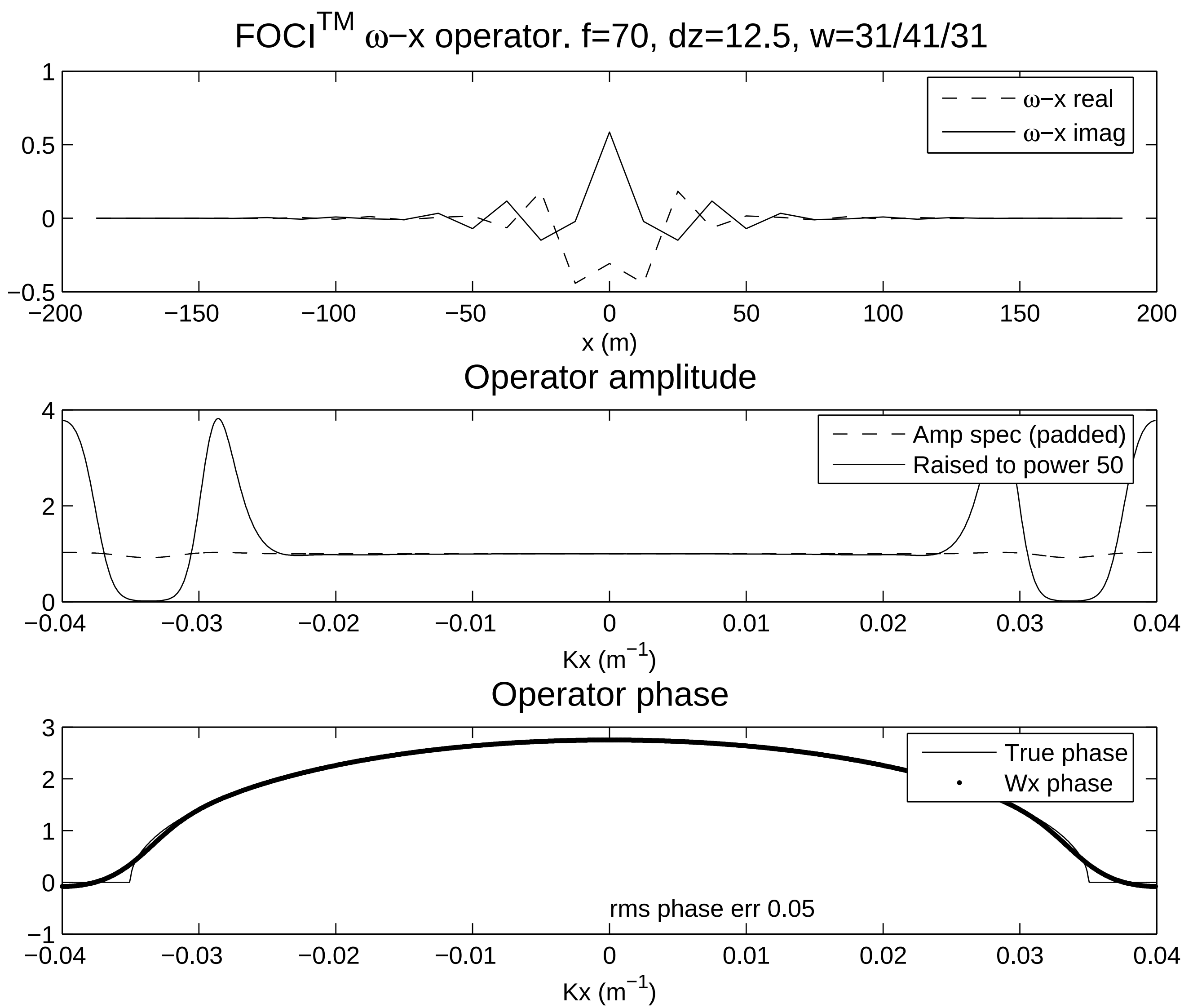
V(z)

$$\alpha = \exp\left(i\Delta z \sqrt{\frac{\omega^2}{v(x_0)^2} - k_x^2} \right)$$

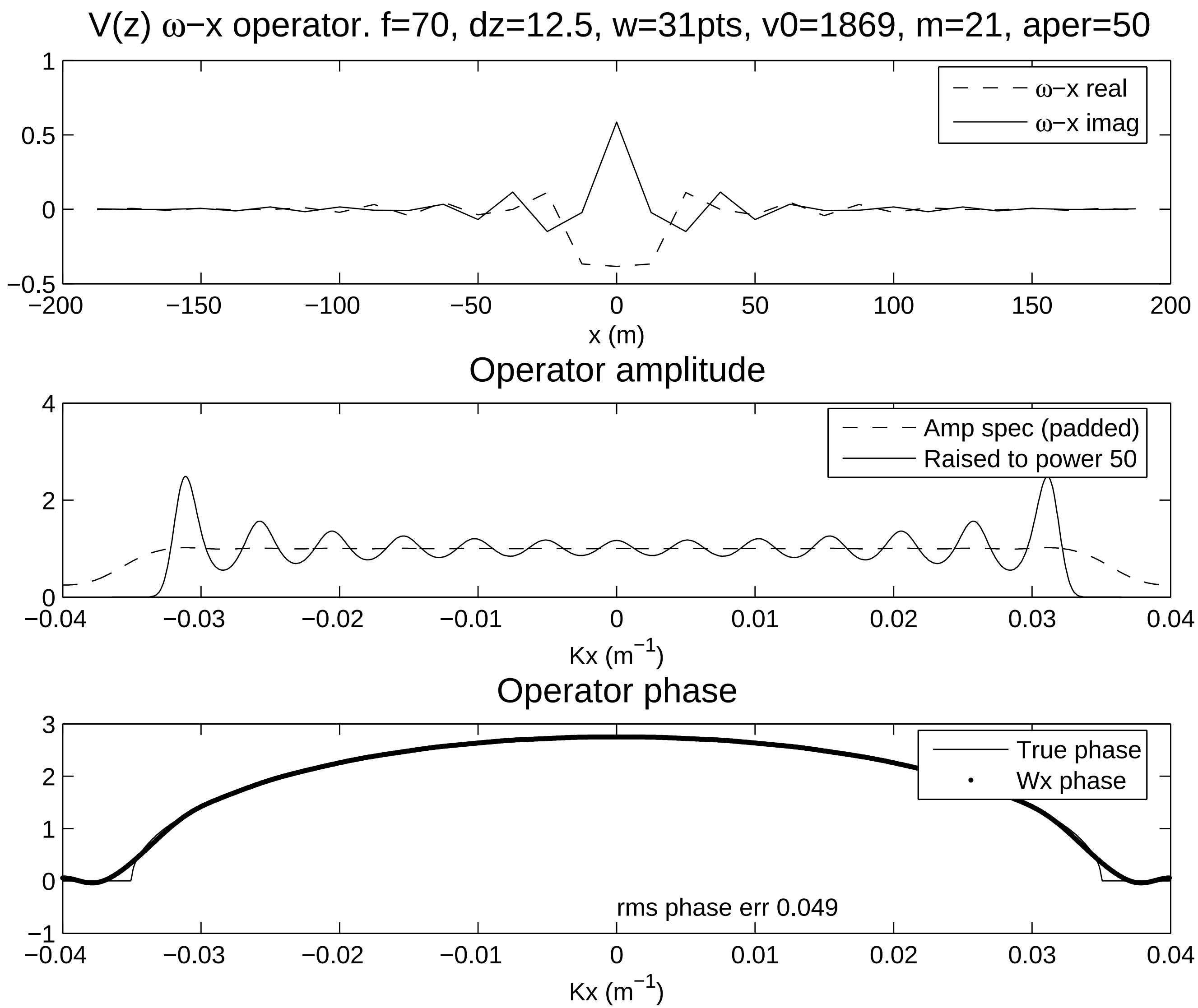
$$\alpha = \exp\left(i \int_0^{\Delta z} \sqrt{\frac{\omega^2}{v(z')^2} - k_x^2 dz'} \right)$$

$$v(z) = v_0 + Az$$

FOCI in $\omega - x$

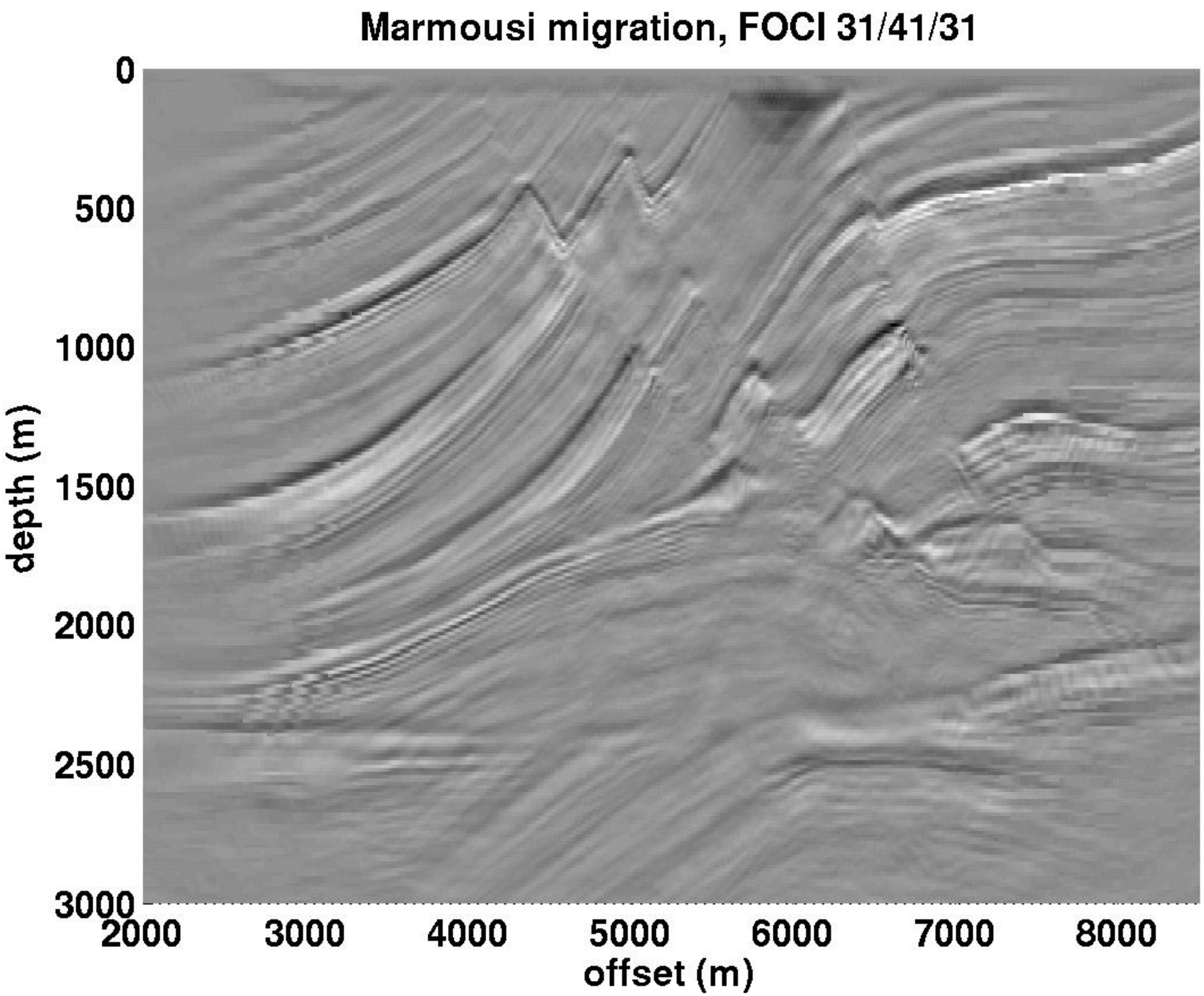


V(z) in $\omega - x$



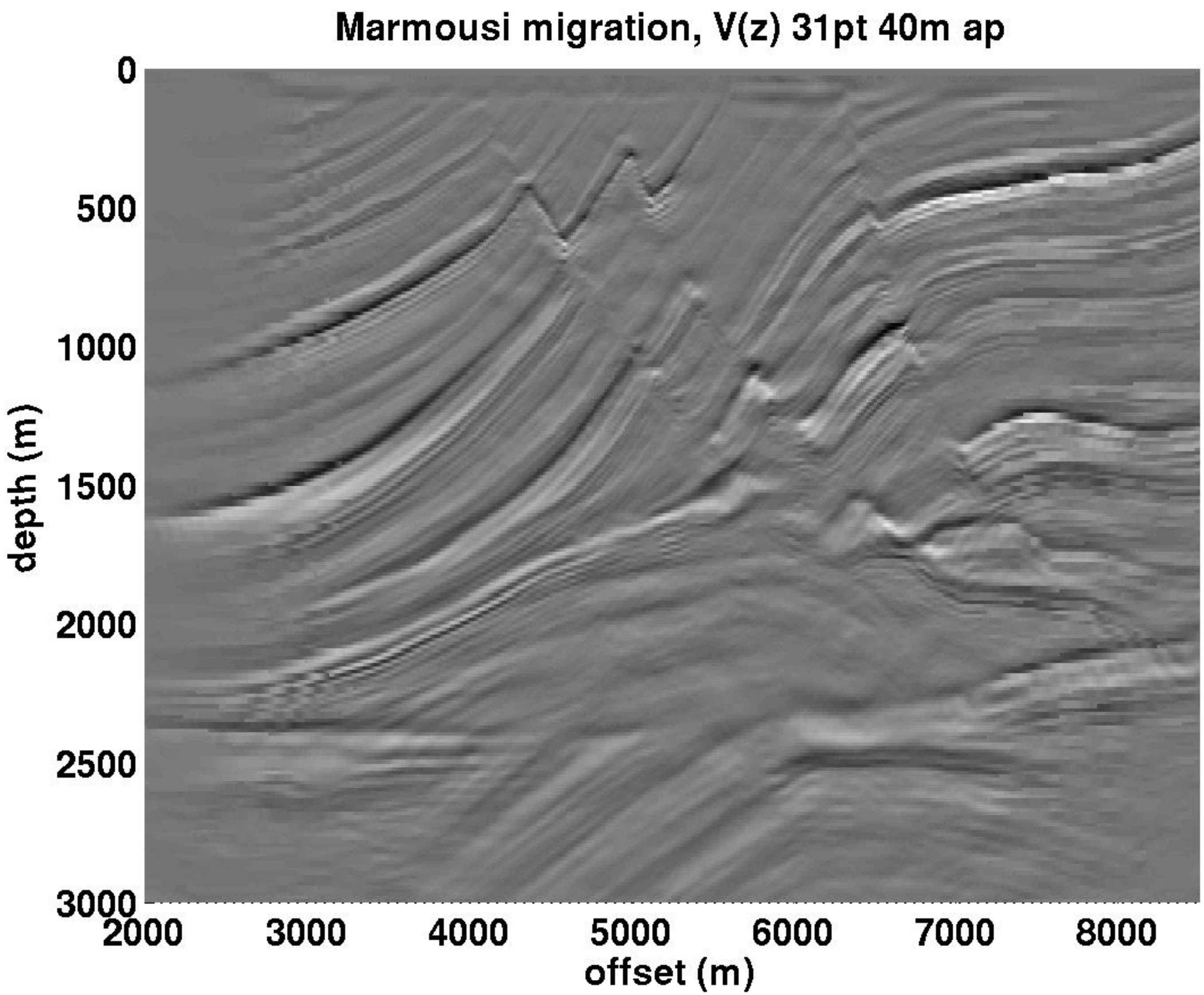
V(z) ω−x operator. f=70, dz=12.5, w=31pts, v0=1869, m=21, aper=50
ω−x real
ω−x imag
x (m)
Operator amplitude
Amp spec (padded)
Raised to power 50
Kx (m−1)
Operator phase
True phase
Wx phase
rms phase err 0.049
Kx (m−1)

31 point FOCI image



CREWES

31 point V(z) image, 40m aperture



Conclusions

- GPSPI makes nice images already

Conclusions

- GPSPI makes nice images already
- The Stolk correction only adds computational time

Conclusions

- GPSPI makes nice images already
- The Stolk correction only adds computational time
- ....but maybe Marmousi is a lousy test?

Conclusions

- GPSPI makes nice images already
- The Stolk correction only adds computational time
- ....but maybe Marmousi is a lousy test?
- V(z) does a nice job of truncating the operator naturally

Conclusions

- GPSPI makes nice images already
- The Stolk correction only adds computational time
- ....but maybe Marmousi is a lousy test?
- V(z) does a nice job of truncating the operator naturally
- ....this means we can try it in 3D

Acknowledgements

- Thanks to all CREWES sponsors
- Kevin Hall, Henry Bland, and Rolf Maier
- Hugh Geiger and Saleh Al-Saleh