A numerical tour of wave propagation

Pengliang Yang[*]

ABSTRACT

This tutorial is written for beginners as an introduction to basic wave propagation using finite difference method, from acoustic and elastic wave modeling, to reverse time migration and full waveform inversion. Most of the theoretical delineations summarized in this tutorial have been implemented in Madagascar with Matlab, C and CUDA programming, which will benefit readers' further study.

BASIC WAVE EQUATION

Define $\mathbf{x} = (x, y, z)$, time t, the s-th energy source function $\tilde{S}(\mathbf{x}, t; \mathbf{x}_s)$, pressure $p(\mathbf{x}, t; \mathbf{x}_s)$, particle velocity $\mathbf{v}(\mathbf{x}, t)$, material density $\rho(\mathbf{x})$, the bulk modulus $\kappa(\mathbf{x})$. Now we have

- Newton's law

$$\rho(\mathbf{x}) \frac{\partial \mathbf{v}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t} = \nabla p(\mathbf{x}, t; \mathbf{x}_s). \tag{1}$$

- Constitutive law

$$\frac{1}{\kappa(\mathbf{x})} \frac{\partial p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t} = \nabla \cdot \mathbf{v}(\mathbf{x}, t; \mathbf{x}_s) + \tilde{S}(\mathbf{x}, t; \mathbf{x}_s). \tag{2}$$

Acoustic wave equation

Acoustics is a special case of fluid dynamics (sound waves in gases and liquids) and linear elastodynamics. Note that elastodynamics is a more accurate representation of earth dynamics, but most industrial seismic processing based on acoustic model. Recent interest in quasiacoustic anisotropic approximations to elastic P-waves.

Assume $\tilde{S}(\mathbf{x}, t; \mathbf{x}_s)$ is differentiable constitutive law w.r.t. time t. Substituting Eq. (2) into the differentiation of Eq. (1) gives

$$\frac{1}{\kappa(\mathbf{x})} \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = \nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla p(\mathbf{x}, t; \mathbf{x}_s) \right) + \frac{\partial \tilde{S}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t}. \tag{3}$$

[*]**e-mail:** ypl.2100@gmail.com

We introduce $v(\mathbf{x}) = \sqrt{\kappa(\mathbf{x})/\rho(\mathbf{x})}$ (compressional p-wave velocity):

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = \rho(\mathbf{x})\nabla \cdot \left(\frac{1}{\rho(\mathbf{x})}\nabla p(\mathbf{x}, t; \mathbf{x}_s)\right) + \rho(\mathbf{x})\frac{\partial \tilde{S}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t} \quad (4)$$

Under constant density condition, we obtain the 2nd-order equation

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = \nabla^2 p(\mathbf{x}, t; \mathbf{x}_s) + f_s(\mathbf{x}, t; \mathbf{x}_s) \quad (5)$$

where $\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial z^2}$, $f_s(\mathbf{x}, t; \mathbf{x}_s) = \rho(\mathbf{x})\frac{\partial \tilde{S}(\mathbf{x},t;\mathbf{x}_s)}{\partial t}$. In 2D case, it is

$$\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = v^2(\mathbf{x})\left(\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial z^2} + \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial x^2}\right) + f_s(\mathbf{x}, t; \mathbf{x}_s). \quad (6)$$

A shot of acoustic wavefield obtained at t=0.35s with 4-th order finite difference scheme and the sponge absorbing boundary condition is shown in Figure 1, where the source is put at the center of the model. For 3D, it becomes

$$\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = v^2(\mathbf{x})\left(\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial z^2} + \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial x^2} + \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial y^2}\right) + f_s(\mathbf{x}, t; \mathbf{x}_s) \quad (7)$$

Similarly, we put the source at the center of a 3D volume (size=100x100x100), performed the modeling for 300 steps in time and recorded the corresponding wavefield at kt=250, see Figure 2.

The above spatial operator is spatially homogeneous. This isotropic formula is simple and easy to understand, and becomes the basis for many complicated generalizations in which the anisotropy may come in. In 2D case, the elliptically-anisotropic wave equation reads

$$\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} = v_1^2(\mathbf{x})\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial z^2} + v_2^2(\mathbf{x})\frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial x^2} + f_s(\mathbf{x}, t; \mathbf{x}_s) \quad (8)$$

Here, I use the Hess VTI model shown in Figure 3a and Figure 3b. We perform 1000 steps of modeling with time interval $\Delta t = 0.001s$, and capture the wavefield at $t = 0.9s$, as shown in Figure 4.

Elastic wave equation

In elastic wave equation, the modulus $\kappa(\mathbf{x})$ corresponds to two Lame parameters: $\lambda + 2\mu = \rho v_p^2$ and $u = \rho v_s^2$, in which v_p and v_s denote the P- and S-wave velocity. The
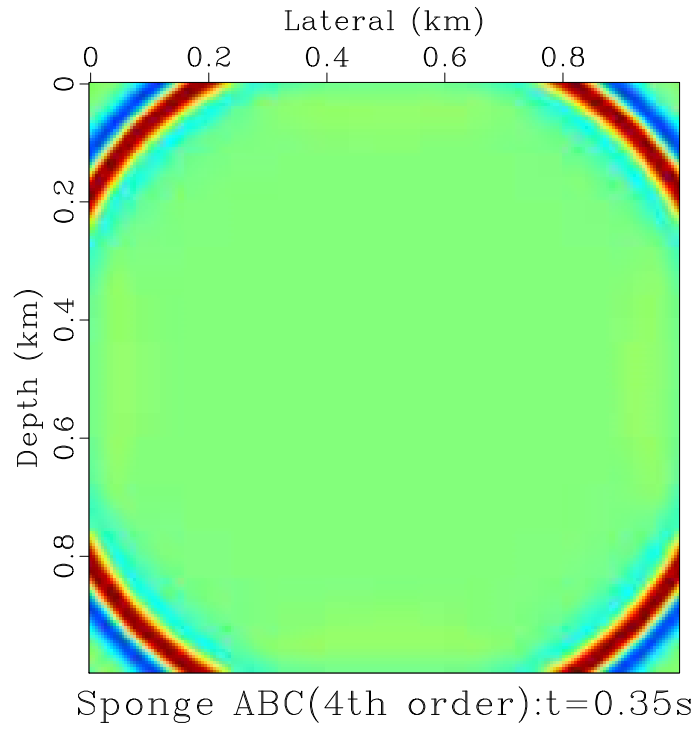
Figure 1: A snap of acoustic wavefield obtained at t=0.35s with 4-th order finite difference scheme and the sponge absorbing boundary condition.
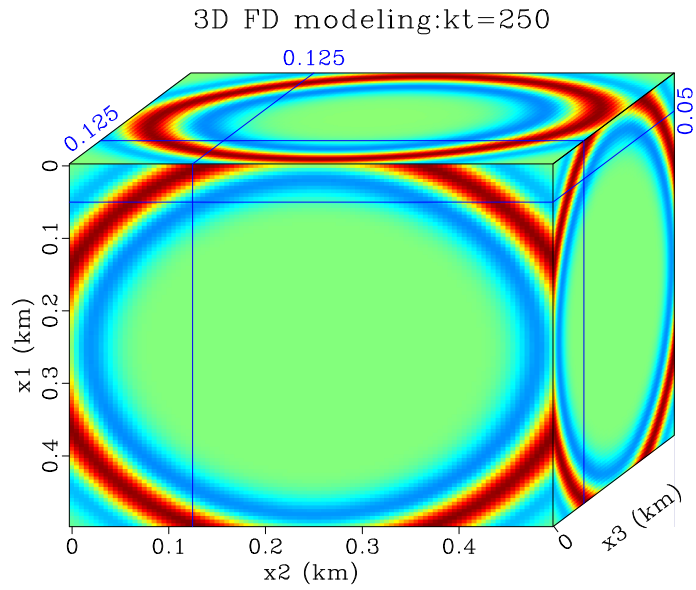

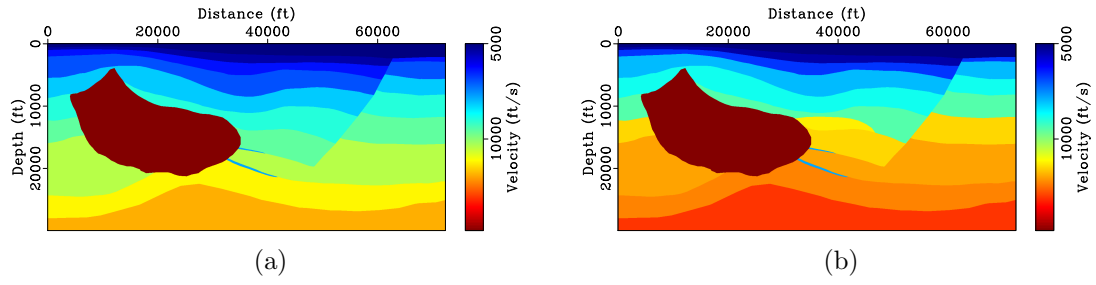Figure 2: A wavefield snap recorded at kt=250, 300 steps modeled.
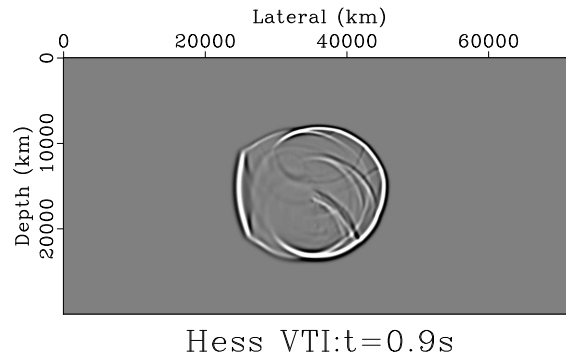
Figure 3: Two velocity components of Hess VTI model

Figure 4: Wavefield at $kt = 0.9s$, 1000 steps of modeling with time interval $\Delta t = 0.001s$ performed.

elastic wave equation can be written as

$$
\begin{cases}
\dfrac{\partial v_x}{\partial t} = \dfrac{1}{\rho}\left(\dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{xz}}{\partial x}\right) \\[2mm]
\dfrac{\partial v_z}{\partial t} = \dfrac{1}{\rho}\left(\dfrac{\partial \tau_{zx}}{\partial z} + \dfrac{\partial \tau_{zz}}{\partial z}\right) \\[2mm]
\dfrac{\partial \tau_{xx}}{\partial t} = (\lambda + 2\mu)\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_z}{\partial z} \\[2mm]
\dfrac{\partial \tau_{zz}}{\partial t} = \lambda\dfrac{\partial v_x}{\partial x} + (\lambda + 2\mu)\dfrac{\partial v_z}{\partial z} \\[2mm]
\dfrac{\partial \tau_{xz}}{\partial t} = \mu\dfrac{\partial v_x}{\partial x} + \mu\dfrac{\partial v_z}{\partial z}
\end{cases}
\tag{9}
$$

where τ_{ij} (sometimes σ_{ij}) is stress, v_i is particle velocity, $i, j = x, z$. We display the 2 components of elastic wave propagation at $kt = 270$, $nt = 300$ modeled with $\Delta t = 0.001$ in Figure 5, in which the grid size is 200x200, the spatial interval is $\Delta x = \Delta z = 5m$, and the velocities are chosen to be $Vp = 2km/s$, $Vs = Vp/\sqrt{2}$.
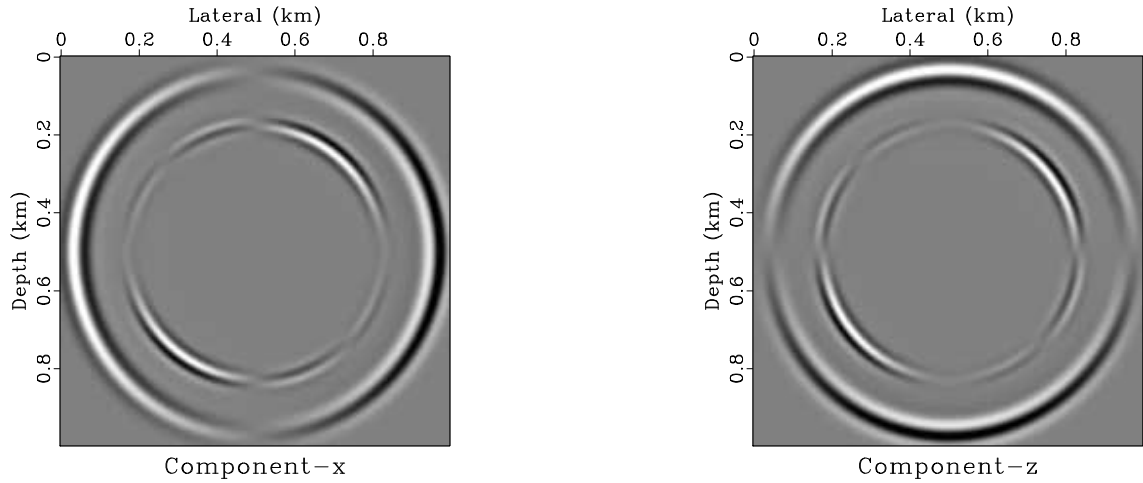

Figure 5: Two components of elastic wave propagation at $kt = 270$, $nt = 300$ modeled with $\Delta t = 0.001$. Grid size=200x200,$\Delta x = \Delta z = 5m$, $Vp = 2km/s$, $Vs = Vp/\sqrt{2}$

FORWARD MODELING

Taylor and Páde expansion

The Taylor expansion of a function $f(x + h)$ at x is written as

$$
f(x + h) = f(x) + \frac{\partial f(x)}{\partial x}h + \frac{1}{2!}\frac{\partial^2 f(x)}{\partial x^2}h^2 + \frac{1}{3!}\frac{\partial^3 f(x)}{\partial x^3}h^3 + \ldots.
\tag{10}
$$

A popular example is

$$
(1 + x)^\alpha = 1 + \alpha x + \frac{\alpha(\alpha - 1)}{2!}x^2 + \ldots + \frac{\alpha(\alpha - 1)\cdots(\alpha - n + 1)}{n!}x^n + \ldots.
\tag{11}
$$

Here we mainly consider the following expansion formula:

$$(1-x)^{\frac{1}{2}} = 1 - \frac{1}{2}x - \frac{1}{8}x^2 - \frac{1}{16}x^3 - \frac{5}{128}x^4 - \dots, |x| < 1. \tag{12}$$

The Páde expansion of Eq. (12) follows from expansion in continuous fractions:

$$(1-x)^{\frac{1}{2}} = 1 - \frac{x/2}{1 - \frac{x/4}{1 - \frac{x/4}{\dots}}} \tag{13}$$

I provide an informal derivation:

$$y = (1-x)^{\frac{1}{2}} \Rightarrow x = 1 - y^2 = (1-y)(1+y) \Rightarrow 1 - y = \frac{x}{1+y}$$

$$y = 1 - \frac{x}{1+y} = 1 - \frac{x}{1 + (1 - \frac{x}{1+y})} = 1 - \frac{x/2}{1 - \frac{x/2}{1+y}}$$

$$= 1 - \frac{x/2}{1 - \frac{x/2}{2 - \frac{x}{1+y}}} = 1 - \frac{x/2}{1 - \frac{x/4}{1 - \frac{x/2}{1+y}}} = \dots$$

The 1st-order Pade expansion is:

$$(1-x)^{\frac{1}{2}} = 1 - \frac{x}{2} \tag{14}$$

The 2nd-order Pade expansion is:

$$(1-x)^{\frac{1}{2}} = 1 - \frac{x/2}{1 - \frac{x}{4}}. \tag{15}$$

And the 3rd-order one is:

$$(1-x)^{\frac{1}{2}} = 1 - \frac{x/2}{1 - \frac{x/4}{1 - \frac{x}{4}}}. \tag{16}$$

Approximate the wave equation

The innovative work was done by John Claerbout, and is well-known as $15°$ wave equation to separate the up-going and down-going waves (Claerbout, 1971, 1986).

Eliminating the source term, the Fourier transform of the scalar wave equation (Eq. (5)) can be specified as:

$$\frac{\omega^2}{v^2} = k_x^2 + k_z^2. \tag{17}$$

The down-going wave equation in Fourier domain is

$$k_z = \sqrt{\frac{\omega^2}{v^2} - k_x^2} = \frac{\omega}{v}\sqrt{1 - \frac{v^2 k_x^2}{\omega^2}}. \tag{18}$$

Using the different order Pade expansions, we have:

$$
\begin{cases}
1\text{st}-\text{order}: k_z = \dfrac{\omega}{v}\left(1 - \dfrac{v^2 k_x^2}{2\omega^2}\right) \\[3mm]
2\text{nd}-\text{order}: k_z = \dfrac{\omega}{v}\left(1 - \dfrac{\frac{2v^2 k_x^2}{\omega^2}}{4 - \frac{v^2 k_x^2}{\omega^2}}\right) \\[5mm]
3\text{rd}-\text{order}: k_z = \dfrac{\omega}{v}\left(1 - \dfrac{\frac{v^2 k_x^2}{2\omega^2} - \frac{v^4 k_x^4}{8\omega^4}}{1 - \frac{v^2 k_x^2}{2\omega^2}}\right) \\[5mm]
4\text{th}-\text{order}: k_z = \dfrac{\omega}{v}\left(1 - \dfrac{\frac{v^2 k_x^2}{2\omega^2} - \frac{v^4 k_x^4}{4\omega^4}}{1 - \frac{3v^2 k_x^2}{4\omega^2} + \frac{v^4 k_x^4}{16\omega^4}}\right)
\end{cases}
\tag{19}
$$

The corresponding time domain equations are:

$$
\begin{cases}
1\text{st}-\text{order}: \dfrac{\partial^2 p}{\partial t\,\partial z} + \dfrac{v}{2}\dfrac{\partial^2 p}{\partial x^2} - \dfrac{1}{v}\dfrac{\partial^2 p}{\partial t^2} = 0,\ (\text{the well}-\text{known }15^\circ\text{ wave equation}) \\[3mm]
2\text{nd}-\text{order}: \dfrac{\partial^3 p}{\partial t^2 \partial z} - \dfrac{v^2}{4}\dfrac{\partial^3 p}{\partial x^2 \partial z} - \dfrac{1}{v}\dfrac{\partial^3 p}{\partial t^3} + \dfrac{3v}{4}\dfrac{\partial^3 p}{\partial x^2 \partial t} = 0,\ (45^\circ\text{ wave equation}) \\[3mm]
3\text{rd}-\text{order}: \dfrac{\partial^4 p}{\partial t^3 \partial z} - \dfrac{v^2}{2}\dfrac{\partial^4 p}{\partial x^2 \partial t \partial z} - \dfrac{1}{v}\dfrac{\partial^4 p}{\partial t^4} + v\dfrac{\partial^4 p}{\partial x^2 \partial t^2} - \dfrac{v^3}{8}\dfrac{\partial^4 p}{\partial x^4} = 0 \\[3mm]
4\text{th}-\text{order}: \dfrac{\partial^5 p}{\partial t^4 \partial z} - \dfrac{3v^2}{4}\dfrac{\partial^5 p}{\partial x^2 \partial t^2 \partial z} - \dfrac{v^4}{16}\dfrac{\partial^5 p}{\partial x^4 \partial z} + \dfrac{1}{v}\dfrac{\partial^5 p}{\partial t^5} + \dfrac{5v}{4}\dfrac{\partial^5 p}{\partial x^2 \partial t^3} + \dfrac{5v^3}{16}\dfrac{\partial^5 p}{\partial t \partial x^4} = 0
\end{cases}
\tag{20}
$$

Absorbing boundary condition (ABC)

Clayton-Enquist boundary condition

To simulate the wave propagation in the infinite space, the absorbing boundary condition (ABC), namely the proximal approximation (PA) boundary condition, was proposed in Clayton and Engquist (1977) and Engquist and Majda (1977). The basic idea is to use the wave equation with opposite direction at the boundary. Take the bottom boundary as an example. Here, allowing for the incident wave is down-going, we use the up-going wave equation at the bottom boundary. Using a model including 3 layers (Figure 6), Figure 7 displays 10 shots data volume obtained the Clayton-Enquist boundary condition.

Sponge ABC

The sponge ABC was proposed by Cerjan et al. (1985). The principle is very simple: attenuating the refections exponentially in the extended artificial boundary (Figure 8) area by multiplying a factor less $d(u)$ than 1. Commonly, we use the factor

$$
d(u) = \exp(-[0.015 * (nb - i)]^2),\, u = x, z\,(i\Delta x \text{ or } i\Delta z)
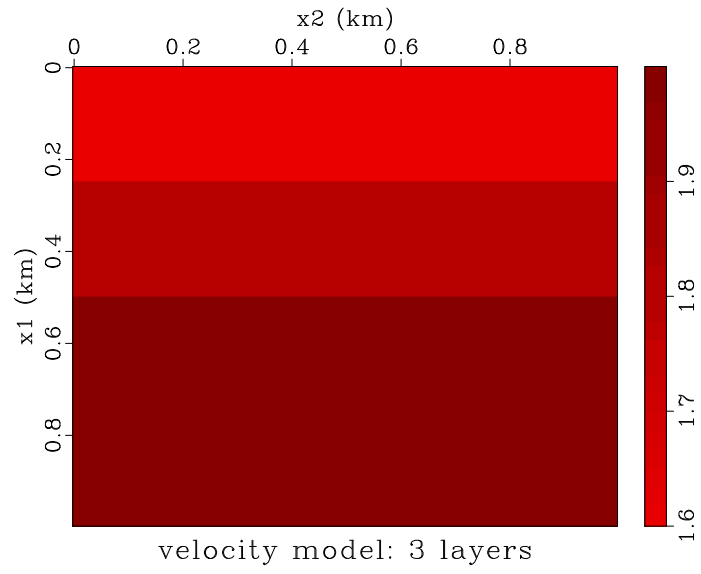\tag{21}
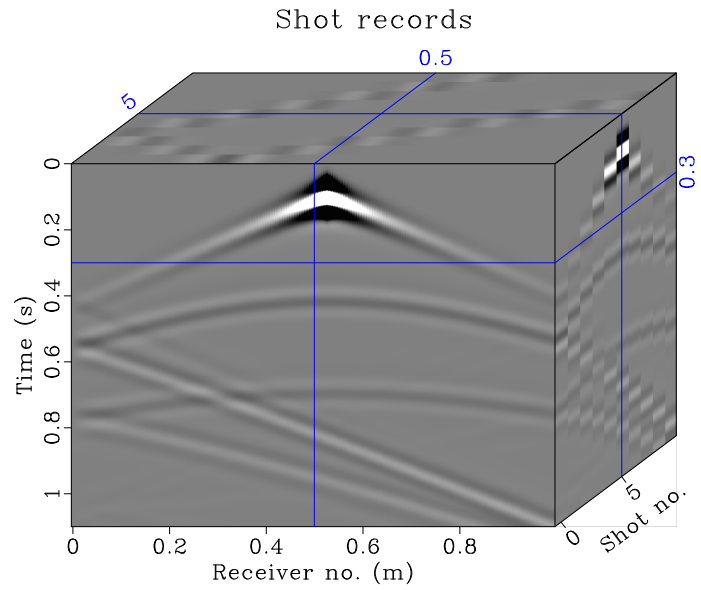$$

Figure 6: Velocity model: 3 layers

Figure 7: 10 shots data volume obtained using the Clayton-Enquist boundary condition.

where nb is the thickness of the artificial boundary on each side of the model. Usually, we choose it to be $nb = 20 \sim 30$. The sponge ABC can be easily applied to a wide range of wave propagation problems, including some governing wave equations for complicated medium.
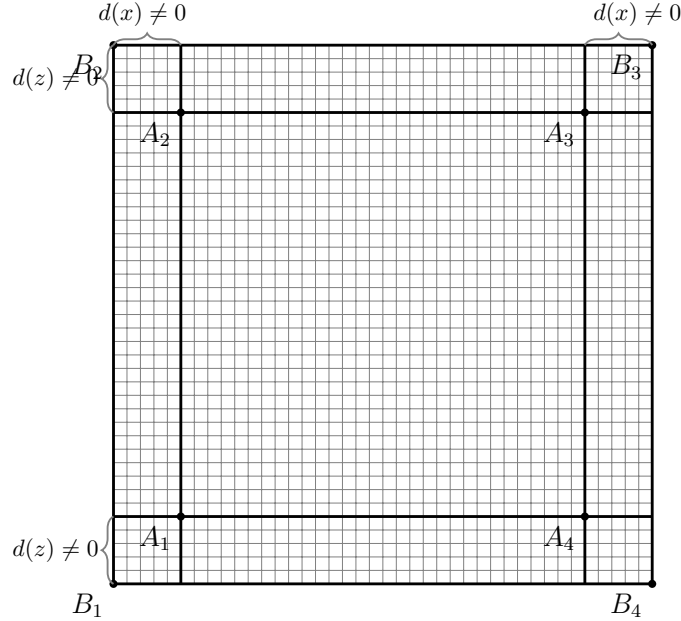

Figure 8: A schematic diagram of extended artificial boundary area. $A_1A_2A_3A_4$ is the original model zone, which is extended to be $B_1B_2B_3B_4$ with artificial boundary. In the extended bounary area, the attenuation coeffcient $d(u) \neq 0$; In the model zone $A_1A_2A_3A_4$, $d(u) = 0$, $u = x, z$.

Perfectly Matched Layer (PML)

The PML ABC was proposed in electromagnetics computation (Berenger, 1994). In seismic wave propagation community, two versions of PML boundary condition have been developed: the split PML (SPML) and nonsplit PML (NPML).

Split PML (SPML) for acoustics

It is possible for us to split the wave field into two components: x-component p_x and z-component p_z (Carcione et al., 2002). Then the acoustic wave equation becomes

$$
\begin{cases}
p = p_x + p_z \\
\dfrac{\partial p_x}{\partial t} = v^2 \dfrac{\partial v_x}{\partial x} \\
\dfrac{\partial p_z}{\partial t} = v^2 \dfrac{\partial v_z}{\partial z} \\
\dfrac{\partial v_x}{\partial t} = \dfrac{\partial p}{\partial x} \\
\dfrac{\partial v_z}{\partial t} = \dfrac{\partial p}{\partial z}
\end{cases}
\tag{22}
$$

To absorb the boundary reflection, by adding the decaying coefficients $d(u)$ the SPML governing equation can be specified as (Collino and Tsogka, 2001)

$$
\begin{cases}
p = p_x + p_z \\
\dfrac{\partial p_x}{\partial t} + d(x)p_x = v^2 \dfrac{\partial v_x}{\partial x} \\
\dfrac{\partial p_z}{\partial t} + d(z)p_z = v^2 \dfrac{\partial v_z}{\partial z} \\
\dfrac{\partial v_x}{\partial t} + d(x)v_x = \dfrac{\partial p}{\partial x} \\
\dfrac{\partial v_z}{\partial t} + d(z)v_z = \dfrac{\partial p}{\partial z}
\end{cases}
\tag{23}
$$

where $d(x)$ and $d(z)$ are the ABC coefficients designed to attenuate the reflection in the boundary zone, see Figure 8. There exists many forms of ABC coefficients function. In the absorbing layers, we use the following model for the damping parameter $d(x)$ (Collino and Tsogka, 2001):

$$
d(u) = d_0 \left(\frac{u}{L}\right)^2, d_0 = -\frac{3v}{2L} \ln(R), u = x, z
\tag{24}
$$

where L indicates the PML thinkness; x represents the distance between current position (in PML) and PML inner boundary. R is always chosen as $10^{-3} \sim 10^{-6}$. It is important to note that the same idea can be applied to elastic wave equation (Collino and Tsogka, 2001). The split version of wave equation is very suitable for the construction of seismic Poynting vector. A straightforward application is the angle gather extration using Poynting vector, see Section .

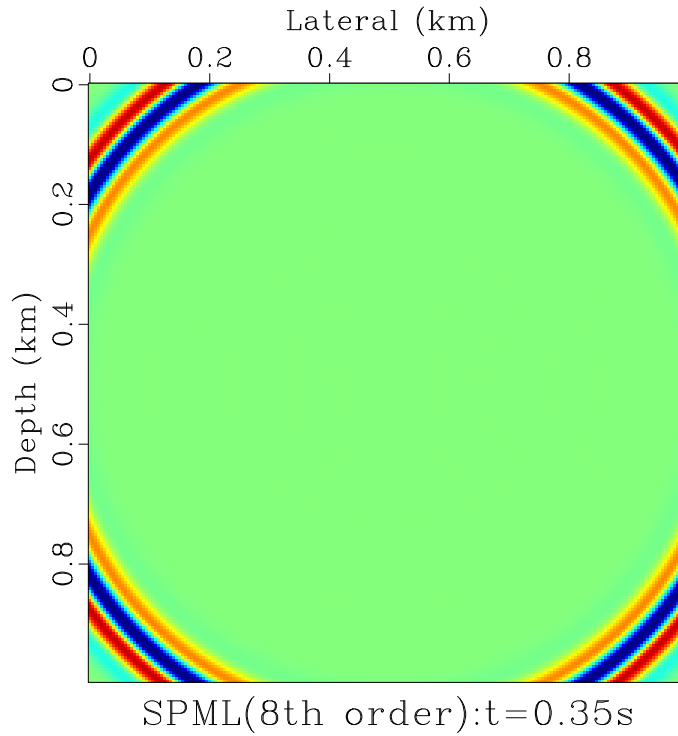A numerical example of SPML using 8th order staggered finite difference scheme is given in Figure 9.

Figure 9: Wavefield snap of SPML with 8th order finite difference

Nonsplit Convolutional-PML (CPML) for acoustics

Another approach to improve the behavior of the discrete PML at grazing incidence consists in modifying the complex coordinate transform used classically in the PML to introduce a frequency-dependent term that implements a Butterworth-type filter in the layer. This approach has been developed for Maxwells equations named convolutional PML (CPML) (Roden and Gedney, 2000) or complex frequency shifted-PML (CFS-PML). The key idea is that for waves whose incidence is close to normal, the presence of such a filter changes almost nothing because absorption is already almost perfect. But for waves with grazing incidence, which for geometrical reasons do not penetrate very deep in the PML, but travel there a longer way in the direction parallel to the layer, adding such a filter will strongly attenuate them and will prevent them from leaving the PML with significant energy .

Define $Ax = \frac{\partial p}{\partial x}$, $Az = \frac{\partial p}{\partial z}$. Then the acoustic wave equation reads

$$\frac{\partial^2 p}{\partial t^2} = v^2 \left(\frac{\partial Ax}{\partial x} + \frac{\partial Az}{\partial z} \right).$$

To combine the absorbing effects into the acoustic equation, we merely need to com-

bine two convolution terms into the above equations:

$$\begin{cases} \dfrac{\partial^2 p}{\partial t^2} = v^2\left(Px + Pz\right) \\[2mm] Px = \dfrac{\partial Ax}{\partial x} + \Psi_x \\[2mm] Pz = \dfrac{\partial Az}{\partial z} + \Psi_z \\[2mm] Ax = \dfrac{\partial p}{\partial x} + \Phi_x \\[2mm] Az = \dfrac{\partial p}{\partial z} + \Phi_z \end{cases} \tag{25}$$

where Ψ_x, Ψ_z are the convolution terms of Ax and Az; Φ_x, Φ_z are the convolution terms of Px and Pz. These convolution terms can be computed via the following relation:

$$\begin{cases} \Psi_x^n = b_x \Psi_x^{n-1} + (b_x - 1)\partial_x^{n-1/2} Ax \\[2mm] \Psi_z^n = b_z \Psi_z^{n-1} + (b_z - 1)\partial_z^{n-1/2} Az \\[2mm] \Phi_x^n = b_x \Phi_x^{n-1} + (b_x - 1)\partial_x^{n-1/2} P \\[2mm] \Phi_z^n = b_z \Phi_z^{n-1} + (b_z - 1)\partial_z^{n-1/2} P \end{cases} \tag{26}$$

where $b_x = e^{-d(x)\Delta t}$ and $b_z = e^{-d(z)\Delta t}$. More details about the derivation of C-PML, the interested readers are referred to Collino and Tsogka (2001) and Komatitsch and Martin (2007).

Nonsplit PML (NPML) for elastics

The nonsplit PML for elastic implementation is

$$\begin{cases} \rho\dfrac{\partial v_x}{\partial t} = \left(\dfrac{\partial \tau_{xx}}{\partial x} + \dfrac{\partial \tau_{xz}}{\partial z}\right) - \Omega_{xx} - \Omega_{xz} \\[2mm] \rho\dfrac{\partial v_z}{\partial t} = \left(\dfrac{\partial \tau_{xz}}{\partial x} + \dfrac{\partial \tau_{zz}}{\partial z}\right) - \Omega_{zx} - \Omega_{zz} \\[2mm] \dfrac{\partial \tau_{xx}}{\partial t} = (\lambda + 2\mu)\dfrac{\partial v_x}{\partial x} + \lambda\dfrac{\partial v_z}{\partial z} - (\lambda + 2\mu)\Psi_{xx} - \lambda\Psi_{zz} \\[2mm] \dfrac{\partial \tau_{zz}}{\partial t} = \lambda\dfrac{\partial v_x}{\partial x} + (\lambda + 2\mu)\dfrac{\partial v_z}{\partial z} - \lambda\Psi_{xx} - (\lambda + 2\mu)\Psi_{zz} \\[2mm] \dfrac{\partial \tau_{xz}}{\partial t} = \mu\dfrac{\partial v_x}{\partial x} + \mu\dfrac{\partial v_z}{\partial z} - \mu\Psi_{zx} - \mu\Psi_{xz} \end{cases} \tag{27}$$

where the auxiliary variables are governed via the following relation

$$
\begin{cases}
\dfrac{\partial \Omega_{xx}}{\partial t} + d(x)\Omega_{xx} = d(x)\dfrac{\partial \tau_{xx}}{\partial x}, \dfrac{\partial \Omega_{xz}}{\partial t} + d(z)\Omega_{xz} = d(z)\dfrac{\partial \tau_{xz}}{\partial z} \\[2mm]
\dfrac{\partial \Omega_{zx}}{\partial t} + d(x)\Omega_{zx} = d(x)\dfrac{\partial \tau_{xz}}{\partial x}, \dfrac{\partial \Omega_{zz}}{\partial t} + d(z)\Omega_{zz} = d(z)\dfrac{\partial \tau_{zz}}{\partial z} \\[2mm]
\dfrac{\partial \Psi_{xx}}{\partial t} + d(x)\Psi_{xx} = d(x)\dfrac{\partial v_x}{\partial x}, \dfrac{\partial \Psi_{xz}}{\partial t} + d(z)\Psi_{xz} = d(z)\dfrac{\partial v_x}{\partial z} \\[2mm]
\dfrac{\partial \Psi_{zx}}{\partial t} + d(x)\Psi_{zx} = d(x)\dfrac{\partial v_z}{\partial x}, \dfrac{\partial \Psi_{zz}}{\partial t} + d(z)\Psi_{zz} = d(z)\dfrac{\partial v_z}{\partial z}
\end{cases}
\tag{28}
$$

Discretization

The Taylor series expansion of a function $f(x)$ can be written as

$$
\begin{cases}
f(x+h) = f(x) + \dfrac{\partial f(x)}{\partial x}h + \dfrac{1}{2!}\dfrac{\partial^2 f(x)}{\partial x^2}h^2 + \dfrac{1}{3!}\dfrac{\partial^3 f(x)}{\partial x^3}h^3 + \ldots \\[3mm]
f(x-h) = f(x) - \dfrac{\partial f(x)}{\partial x}h + \dfrac{1}{2!}\dfrac{\partial^2 f(x)}{\partial x^2}h^2 - \dfrac{1}{3!}\dfrac{\partial^3 f(x)}{\partial x^3}h^3 + \ldots
\end{cases}
\tag{29}
$$

It leads to

$$
\begin{cases}
\dfrac{f(x+h) + f(x-h)}{2} = f(x) + \dfrac{1}{2!}\dfrac{\partial^2 f(x)}{\partial x^2}h^2 + \dfrac{1}{4!}\dfrac{\partial^4 f(x)}{\partial x^4}h^4 + \ldots \\[3mm]
\dfrac{f(x+h) - f(x-h)}{2} = \dfrac{\partial f(x)}{\partial x}h + \dfrac{1}{3!}\dfrac{\partial^3 f(x)}{\partial x^3}h^3 + \dfrac{1}{5!}\dfrac{\partial^5 f(x)}{\partial x^5}h^5 + \ldots
\end{cases}
\tag{30}
$$

Let $h = \Delta x/2$. This implies

$$
\begin{cases}
\dfrac{\partial f(x)}{\partial x} = \dfrac{f(x + \Delta x/2) - f(x - \Delta x/2)}{\Delta x} + O(\Delta x^2) \\[3mm]
f(x) = \dfrac{f(x + \Delta x/2) + f(x - \Delta x/2)}{2} + O(\Delta x^2)
\end{cases}
\tag{31}
$$

Higher-order approximation of staggered-grid finite difference

To approximate the 1st-order derivatives as accurate as possible, we express it in the following

$$
\begin{aligned}
\frac{\partial f}{\partial x} =& a_1 \frac{f(x+\Delta x/2) - f(x-\Delta x/2)}{\Delta x} + \\
& a_2 \frac{f(x+3\Delta x/2) - f(x-3\Delta x/2)}{3\Delta x} + \\
& a_3 \frac{f(x+5\Delta x/2) - f(x-5\Delta x/2)}{5\Delta x} + \cdots \\
=& c_1 \frac{f(x+\Delta x/2) - f(x-\Delta x/2)}{\Delta x} + \\
& c_2 \frac{f(x+3\Delta x/2) - f(x-3\Delta x/2)}{\Delta x} + \\
& c_3 \frac{f(x+5\Delta x/2) - f(x-5\Delta x/2)}{\Delta x} + \cdots
\end{aligned}
\tag{32}
$$

where $c_i = a_i/(2i-1)$. Substituting the $f(x+h)$ and $f(x-h)$ with (29) for $h = \Delta x/2, 3\Delta x/2, \ldots$ results in

$$
\begin{aligned}
\frac{\partial f}{\partial x} =& c_1 \left(\Delta x \frac{\partial f}{\partial x} + \frac{1}{3}(\frac{\Delta x}{2})^2 \frac{\partial^3 f}{\partial x^3} + \cdots \right) / \Delta x \\
& + c_2 \left(3\Delta x \frac{\partial f}{\partial x} + \frac{1}{3}(\frac{3\Delta x}{2})^2 \frac{\partial^3 f}{\partial x^3} + \cdots \right) / \Delta x \\
& + c_3 \left(5\Delta x \frac{\partial f}{\partial x} + \frac{1}{3}(\frac{5\Delta x}{2})^2 \frac{\partial^3 f}{\partial x^3} + \cdots \right) / \Delta x + \ldots \\
=& (c_1 + 3c_2 + 5c_3 + 7c_4 + \cdots) \frac{\partial f}{\partial x} \\
& + \frac{\Delta x^2}{3 \cdot 2^2}(c_1 + 3^3 c_2 + 5^3 c_3 + 7^3 c_4 + \cdots) \frac{\partial^3 f}{\partial x^3} \\
& + \frac{\Delta x^4}{3 \cdot 2^4}(c_1 + 3^5 c_2 + 5^5 c_3 + 7^5 c_4 + \cdots) \frac{\partial^5 f}{\partial x^5} + \cdots \\
=& (a_1 + a_2 + a_3 + a_4 + \cdots) \frac{\partial f}{\partial x} \\
& + \frac{\Delta x^2}{3 \cdot 2^2}(a_1 + 3^2 a_2 + 5^2 a_3 + 7^2 a_4 + \cdots) \frac{\partial^3 f}{\partial x^3} \\
& + \frac{\Delta x^4}{3 \cdot 2^4}(a_1 + 3^4 a_2 + 5^4 a_3 + 7^4 a_4 + \cdots) \frac{\partial^5 f}{\partial x^5} + \cdots
\end{aligned}
\tag{33}
$$

Thus, taking first N terms means

$$
\begin{cases}
a_1 + a_2 + a_3 + \cdots + a_N = 1 \\
a_1 + 3^2 a_2 + 5^2 a_3 + \cdots + (2N-1)^2 a_N = 0 \\
a_1 + 3^4 a_2 + 5^4 a_3 + \cdots + (2N-1)^4 a_N = 0 \\
\qquad\qquad \cdots \\
a_1 + 3^{2N-2} a_2 + 5^{2N-2} a_3 + \cdots) + (2N-1)^{2N-2} a_N = 0
\end{cases}
\tag{34}
$$

In matrix form,

$$
\underbrace{\begin{bmatrix} 1 & 1 & \ldots & 1 \\ 1^2 & 3^2 & \ldots & (2N-1)^2 \\ \vdots & & \ddots & \vdots \\ 1^{2N-2} & 3^{2N-2} & \ldots & (2N-1)^{2N-2} \end{bmatrix}}_{\mathbf{V}} \underbrace{\begin{bmatrix} a_1 \\ a_2 \\ \vdots \\ a_N \end{bmatrix}}_{\mathbf{a}} = \underbrace{\begin{bmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{bmatrix}}_{\mathbf{b}}
\tag{35}
$$

The above matrix equation is Vandermonde-like system: $\mathbf{Va} = \mathbf{b}$, $\mathbf{a} = (a_1, a_2, \ldots, a_N)^T$. The Vandermonde matrix

$$
\mathbf{V} = \begin{bmatrix} 1 & 1 & \ldots & 1 \\ x_1 & x_2 & \ldots & x_N \\ \vdots & & \ddots & \vdots \\ x_1^{N-1} & x_2^{N-1} & \ldots & x_N^{N-1} \end{bmatrix}
\tag{36}
$$

in which $x_i = (2i - 1)^2$, has analytic solutions. $\mathbf{Va} = \mathbf{b}$ can be solved using the specific algorithms, see Bjorck (1996). And we obtain

$$
\frac{\partial f}{\partial x} = \frac{1}{\Delta x} \sum_{i=1}^{N} c_i (f(x + i\Delta x/2) - f(x - i\Delta x/2) + O(\Delta x^{2N})
\tag{37}
$$

The MATLAB code for solving the $2N$-order finite difference coefficients is provided in the following.

```matlab
function c=staggered_fdcoeff(NJ)
% Computing 2*N-order staggered-grid FD coefficients (NJ=2N)
% Example:
%       format long
%       NJ=10;
%       c=staggered_fdcoeff(NJ)

N=NJ/2;
x=zeros(N,1);
b=zeros(N,1);    b(1)=1;
c=b;
for k=1:N
     x(k)=(2*k-1)^2;
end

for k=1:N-1
     for i=N:-1:k+1
          b(i)=b(i)-x(k)*b(i-1);
     end
end
```

```
21
22  for  k=N−1:−1:1
23       for  i=k+1:N
24            b(i)=b(i)/(x(i)−x(i−k));
25       end
26       for  i=k:N−1
27            b(i)=b(i)−b(i+1);
28       end
29  end
30  for  k=1:N
31       c(k)=b(k)/(2*k−1);
32  end
```

In general, the stability of staggered-grid difference requires that

$$\Delta t \max(v)\sqrt{\frac{1}{\Delta x^2} + \frac{1}{\Delta z^2}} \le \frac{1}{\sum_{i=1}^{N}|c_i|}. \tag{38}$$

Define $C = \frac{1}{\sum_{i=1}^{N}|c_i|}$. Then, we have

$$\begin{cases} N = 1, & C = 1 \\ N = 2, & C = 0.8571 \\ N = 3, & C = 0.8054 \\ N = 4, & C = 0.7774 \\ N = 5, & C = 0.7595 \end{cases}$$

In the 2nd-order case, numerical dispersion is limited when

$$\max(\Delta x, \Delta z) < \frac{\min(v)}{10 f_{\max}}. \tag{39}$$

The 4th-order dispersion relation is:

$$\max(\Delta x, \Delta z) < \frac{\min(v)}{5 f_{\max}}. \tag{40}$$

Discretization of SPML

Take

$$\frac{\partial p_x}{\partial t} + d(x)p_x = v^2 \frac{\partial v_x}{\partial x}$$

for an example. Using the 2nd-order approximation in Eq. (31), we expand it at the time $(k + \frac{1}{2})\Delta t$ and the point $[ix\Delta x, iz\Delta z]$

$$\frac{p_x^{k+1}[ix, iz] - p_x^{k}[ix, iz]}{\Delta t} + d[ix]\frac{p_x^{k+1}[ix, iz] + p_x^{k}[ix, iz]}{2} = v^2[ix, iz]\frac{v_x^{k+\frac{1}{2}}[ix + \frac{1}{2}, iz] - v_x^{k+\frac{1}{2}}[ix - \frac{1}{2}, iz]}{\Delta x} \tag{41}$$

That is to say,

$$p_x^{k+1}[ix,iz] = \frac{1-0.5\Delta td[ix]}{1+0.5\Delta td[ix]}p_x^k[ix,iz] + \frac{1}{1+0.5\Delta td[ix]}\frac{\Delta tv^2[ix,iz]}{\Delta x}(v_x^{k+1\frac{1}{2}}[ix+\frac{1}{2},iz]-v_x^{k+\frac{1}{2}}[ix-\frac{1}{2},iz])$$
(42)

At time $k\Delta t$ and $[ix+\frac{1}{2},iz]$, we expand

$$\frac{\partial v_x}{\partial t} + d(x)v_x = \frac{\partial p}{\partial x}$$

as

$$\frac{v_x^{k+\frac{1}{2}}[ix+\frac{1}{2},iz]-v_x^{k-\frac{1}{2}}[ix+\frac{1}{2},iz]}{\Delta t}+d[ix]\frac{v_x^{k+\frac{1}{2}}[ix+\frac{1}{2},iz]+v_x^{k-\frac{1}{2}}[ix+\frac{1}{2},iz]}{2} = \frac{p^k[ix+1,iz]-p^k[ix,iz]}{\Delta x}$$
(43)

Thus, we have

$$v_x^{k+\frac{1}{2}}[ix+\frac{1}{2},iz] = \frac{1-0.5\Delta td[ix]}{1+0.5\Delta td[ix]}v_x^{k-\frac{1}{2}}[ix+\frac{1}{2},iz] + \frac{1}{1+0.5\Delta td[ix]}\frac{\Delta t}{\Delta x}(p^k[ix+1,iz]-p^k[ix,iz])$$
(44)

In summary,

$$
\begin{cases}
v_x^{k+\frac{1}{2}}[ix+\frac{1}{2},iz] = \dfrac{1-0.5\Delta td[ix]}{1+0.5\Delta td[ix]}v_x^{k-\frac{1}{2}}[ix+\frac{1}{2},iz]+ \\[2mm]
\dfrac{1}{1+0.5\Delta td[ix]}\dfrac{\Delta t}{\Delta x}(p^k[ix+1,iz]-p^k[ix,iz]) \\[2mm]
v_z^{k+\frac{1}{2}}[ix,iz+\frac{1}{2}] = \dfrac{1-0.5\Delta td[iz]}{1+0.5\Delta td[iz]}v_z^{k-\frac{1}{2}}[ix,iz+\frac{1}{2}]+ \\[2mm]
\dfrac{1}{1+0.5\Delta td[iz]}\dfrac{\Delta t}{\Delta z}(p^k[ix,iz+1]-p^k[ix,iz]) \\[2mm]
p_x^{k+1}[ix,iz] = \dfrac{1-0.5\Delta td[ix]}{1+0.5\Delta td[ix]}p_x^k[ix,iz]+ \\[2mm]
\dfrac{1}{1+0.5\Delta td[ix]}\dfrac{\Delta tv^2[ix,iz]}{\Delta x}(v_x^{k+\frac{1}{2}}[ix+\frac{1}{2},iz]-v_x^{k+\frac{1}{2}}[ix-\frac{1}{2},iz]) \\[2mm]
p_z^{k+1}[ix,iz] = \dfrac{1-0.5\Delta td[iz]}{1+0.5\Delta td[iz]}p_z^k[ix,iz]+ \\[2mm]
\dfrac{1}{1+0.5\Delta td[iz]}\dfrac{\Delta tv^2[ix,iz]}{\Delta z}(v_z^{k+\frac{1}{2}}[ix,iz+\frac{1}{2}]-v_z^{k+\frac{1}{2}}[ix,iz-\frac{1}{2}]) \\[2mm]
p^{k+1}[ix,iz] = p_x^{k+1}[ix,iz]+p_z^{k+1}[ix,iz]
\end{cases}
$$
(45)

If we define:

$$b' = \frac{1-0.5\Delta td}{1+0.5\Delta td}, b = \exp(-\Delta td)$$
(46)

we can easily find that b' is a good approximation of b up to 2nd order, allowing for the 2nd order Pade expansion:

$$\exp(z) \approx \frac{1+0.5z}{1-0.5z}$$
(47)

Then, we have

$$1 - b \approx 1 - b' = \frac{\Delta t d}{1 + 0.5 \Delta t d} \tag{48}$$

Discretization of NPML

Note that all sub-equations can be formulated in the following form:

$$\frac{\partial f}{\partial t} + df = \gamma. \tag{49}$$

The analytic solution of this equation is

$$f = -\frac{1}{d} e^{-dt} + \frac{1}{d} \gamma \tag{50}$$

In discrete form,

$$f(k\Delta t) = -\frac{1}{d} e^{-dk\Delta t} + \frac{1}{d} \gamma,$$
$$f((k+1)\Delta t) = -\frac{1}{d} e^{-dt} e^{-dk\Delta t} + \frac{1}{d} \gamma. \tag{51}$$

Thus,

$$f((k+1)\Delta t) = e^{-d\Delta t} f(k\Delta t) + \frac{1}{d}(1 - e^{-d\Delta t})\gamma \tag{52}$$

For $\frac{\partial \Omega_{xx}}{\partial t} + d(x)\Omega_{xx} = d(x)\frac{\partial \tau_{xx}}{\partial x}$, $\gamma = d(x)\frac{\partial \tau_{xx}}{\partial x}$, the update rule becomes

$$\Omega_{xx}^{k+1} = e^{-d(x)\Delta t}\Omega_{xx}^{k} + (1 - e^{-d(x)\Delta t})\frac{\partial \tau_{xx}^{k+1/2}}{\partial x} = b_x \Omega_{xx}^{k} + (1 - b_x)\frac{\partial \tau_{xx}^{k+1/2}}{\partial x} \tag{53}$$

where $b_x = e^{-d(x)\Delta t}$ and $b_z = e^{-d(z)\Delta t}$. Ω_{xx}, Ω_{xz}, Ω_{zx}, Ω_{zz}, Ψ_{xx}, Ψ_{xz}, Ψ_{zx} and Ψ_{zz} can be obtained in the same way:

$$\begin{cases} \Omega_{xx}^{k+1} = b_x \Omega_{xx}^{k} + (1 - b_x)\dfrac{\partial \tau_{xx}^{k+1/2}}{\partial x}, \Omega_{xz}^{k+1} = b_z \Omega_{xz}^{k} + (1 - b_z)\dfrac{\partial \tau_{xz}^{k+1/2}}{\partial z} \\[2mm] \Omega_{zx}^{k+1} = b_x \Omega_{zx}^{k} + (1 - b_z)\dfrac{\partial \tau_{xz}^{k+1/2}}{\partial x}, \Omega_{zz}^{k+1} = b_z \Omega_{zz}^{k} + (1 - b_z)\dfrac{\partial \tau_{zz}^{k+1/2}}{\partial z} \\[2mm] \Psi_{xx}^{k+1} = b_x \Psi_{xx}^{k} + (1 - b_x)\dfrac{\partial v_x^{k+1/2}}{\partial x}, \Psi_{xz}^{k+1} = b_z \Psi_{xz}^{k} + (1 - b_z)\dfrac{\partial v_x^{k+1/2}}{\partial z} \\[2mm] \Psi_{zx}^{k+1} = b_x \Psi_{zx}^{k} + (1 - b_x)\dfrac{\partial v_z^{k+1/2}}{\partial x}, \Psi_{zz}^{k+1} = b_z \Psi_{zz}^{k} + (1 - b_z)\dfrac{\partial v_z^{k+1/2}}{\partial z} \end{cases} \tag{54}$$

As can be seen from Eq. (27), we only need to subtract the reflection part Ω and Ψ after global updating (Eq. (9)). We summarize this precedure as follows:

Step 1: Perform the computation of Eq. (9) in whole area;

Step 2: In PML zone, subtract decaying parts according to Eq. (27).

REVERSE TIME MIGRATION (RTM)

Brief overview

One-way equation based imaging techniques are inadequate to obtain accurate images in complex media due to propagation direction changes in the background model (Biondi, 2006). These approaches are extremely limited when handling the problems of turning waves in the model containing sharp wave-speed contrasts and steeply dipping reflectors. As an advanced imaging technology without dip and extreme lateral velocity limitation, reverse time migration (RTM) was proposed early (Baysal et al., 1983; McMechan, 1983), but not practical in terms of stringent computation and memory requirement. However, it gained increasingly attention in recent years due to the tremendous advances in computer capability. Until recently, 3D prestack RTM is now feasible to obtain high fidelity images (Yoon et al., 2003; Guitton et al., 2006).

RTM implementation

RTM can be carried out as follows: (1) forward-extrapolating the source wavefield,(2) backward-extrapolating the receiver wavefield, both explicitly in time, and (3) apply an imaging condition.

Imaging condition

The cross-correlation imaging condition can be expressed as

$$I(\mathbf{x}) = \sum_{s=1}^{ns} \int_0^{t_{\max}} \mathrm{d}t \sum_{g=1}^{ng} p_s(\mathbf{x}, t; \mathbf{x}_s) p_g(\mathbf{x}, t; \mathbf{x}_g) \tag{55}$$

where $I(\mathbf{x})$ is the migration image value at point $\mathbf{x}$; and $p_s(\mathbf{x}, t)$ and $p_g(\mathbf{x}, t)$ are the forward and reverse-time wavefields at point $\mathbf{x}$. With illumination compensation, the cross-correlation imaging condition is given by

$$I(\mathbf{x}) = \sum_{s=1}^{ns} \frac{\int_0^{t_{\max}} \mathrm{d}t \sum_{g=1}^{ng} p_s(\mathbf{x}, t; \mathbf{x}_s) p_g(\mathbf{x}, t; \mathbf{x}_g)}{\int_0^{t_{\max}} \mathrm{d}t p_s(\mathbf{x}, t; \mathbf{x}_s) p_s(\mathbf{x}, t; \mathbf{x}_s) + \sigma^2} \tag{56}$$

in which σ^2 is chosen small to avoid being divided by zeros.

There exists a better way to carry out the illumination compensation, as suggested by Guitton et al. (2007)

$$I(\mathbf{x}) = \sum_{s=1}^{ns} \frac{\int_0^{t_{\max}} \mathrm{d}t \sum_{g=1}^{ng} p_s(\mathbf{x}, t; \mathbf{x}_s) p_g(\mathbf{x}, t; \mathbf{x}_g)}{\langle \int_0^{t_{\max}} \mathrm{d}t p_s(\mathbf{x}, t; \mathbf{x}_s) p_s(\mathbf{x}, t; \mathbf{x}_s) \rangle_{x,y,z}} \tag{57}$$

where $\langle\rangle_{x,y,z}$ stands for smoothing in the image space in the x, y, and z directions.

Yoon et al. (2003) define the seismic Poynting vector as

$$\mathbf{S} = \mathbf{v}p = \nabla p \frac{\mathrm{d}p}{\mathrm{d}t} p = (v_x p, v_z p). \tag{58}$$

Here, we denote S_s and S_r as the source wavefield and receiver wavefield Poynting vector. As mentioned before, boundary saving with split PML is a good scheme for the computation of Poynting vector, because p and (v_x, v_z) are available when backward reconstructing the source wavefield. The angle between the incident wave and the reflected wave can then be obtained:

$$\gamma = \arccos \frac{\mathbf{S}_s \cdot \mathbf{S}_r}{|\mathbf{S}_s||\mathbf{S}_r|} \tag{59}$$

The incident angle (or reflective angle) is half of γ, namely,

$$\theta = \frac{\gamma}{2} = \frac{1}{2} \arccos \frac{\mathbf{S}_s \cdot \mathbf{S}_r}{|\mathbf{S}_s||\mathbf{S}_r|} \tag{60}$$

Using Poynting vector to confine the spurious artefacts, Yoon and Marfurt (2006) propose a hard thresholding scheme to weight the imaging condition:

$$I(\mathbf{x}) = \sum_{s=1}^{ns} \frac{\int_0^{t_{\max}} \mathrm{d}t \sum_{g=1}^{ng} p_s(\mathbf{x}, t; \mathbf{x}_s) p_g(\mathbf{x}, t; \mathbf{x}_g) W(\theta)}{\int_0^{t_{\max}} \mathrm{d}t p_s(\mathbf{x}, t; \mathbf{x}_s) p_s(\mathbf{x}, t; \mathbf{x}_s) + \sigma^2} \tag{61}$$

where

$$W(\theta) = \begin{cases} 1 & \theta < \theta_{\max} \\ 0 & otherwise \end{cases} \tag{62}$$

Costa et al. (2009) modified the weight as

$$W(\theta) = \cos^3(\frac{\theta}{2}). \tag{63}$$

These approaches are better for eliminating the backward scattering waves in image.

Computation strategies and boundary saving

There are several possible ways to do RTM computation. The simplest one may be just storing the forward modeled wavefields on the disk, and reading them for imaging condition in the backward propagation steps. This approach requires frequent disk I/O and has been replaced by wavefield reconstruction method. The so-called wavefield reconstruction method is a way to recover the wavefield via backward reconstructing or forward remodeling, using the saved wavefield shots and boundaries. It is of special value for GPU computing because saving the data in device variables eliminates data transfer between CPU and GPU. By saving the last two wavefield

snaps and the boundaries, one can reconstruct the wavefield of every time step, in time-reversal order. The checkpointing technique becomes very useful to further reduce the storage (Symes, 2007; Dussaud et al., 2008). It is also possible to avert the issue of boundary saving by applying the random boundary condition, which may bring some noises in the migrated image (Clapp, 2009; Clapp et al., 2010; Liu et al., 2013b,a).

Yang et al. (2014) proposed an effective boundary for regular and staggered-grid finite differences. In the case of regular grid finite difference of order $2N$, we need to save N points on each inner side in the model zone to reconstruct the wavefield. For staggered grid finite difference of order $2N$, we need to save $2N - 1$ points on each inner in the model for perfect reconstruction. The concept of effective boundary saving does not depends on C or GPU implementation. However, it is of special value for GPU implemenation, because it eliminates the CPU-GPU data transfer for boundary saving. An example of effective boundary saving for regular grid finite difference is given in Figure 10. The imaging examples using effective boundary saving with staggered-grid finite difference can be found in the next section.

Numerical examples

I show my GPU-based RTM result for two benchmark models: Marmousi model (Versteeg, 1994) and Sigsbee model (DiMarco et al., 2001). Here, I use CPML boundary condition to obtain high quality imaging result.

The Marmousi model is shown in Figure 11. The spatial sampling interval is $\Delta x = \Delta z = 4m$. 51 shots are deployed. In each shot, 301 receivers are placed in the split shooting mode. The parameters we use are listed as follows: $nt = 13000$, $\Delta t = 0.3$ ms. Due to the limited resource on our computer, we store 65% boundaries using page-locked memory. Figure 12a and 12b give the resulting RTM image after Laplacian filtering. As shown in the figure, RTM with the effective boundary saving scheme produces excellent image: the normalized cross-correlation imaging condition greatly improves the deeper parts of the image due to the illumination compensation. The events in the central part of the model, the limits of the faults and the thin layers are much better defined.

The Sigsbee model is shown in Figure 13. The spatial interval is $\Delta x = \Delta z = 25m$. 55 shots are evenly distributed on the surface of the model. We still perform $nt = 13000$ time steps for each shot (301 receivers). Due to the larger model size, 75% boundaries have to be stored with the aid of pinned memory. Our RTM results are shown in Figure 14a and 14b. Again, the resulting image obtained by normalized cross-correlation imaging condition exhibits better resolution for the edges of the salt body and the diffraction points. Some events in the image using normalized cross-correlation imaging condition are more visible, while they have a much lower amplitude or are even completely lost in the image of cross-correlation imaging condition.
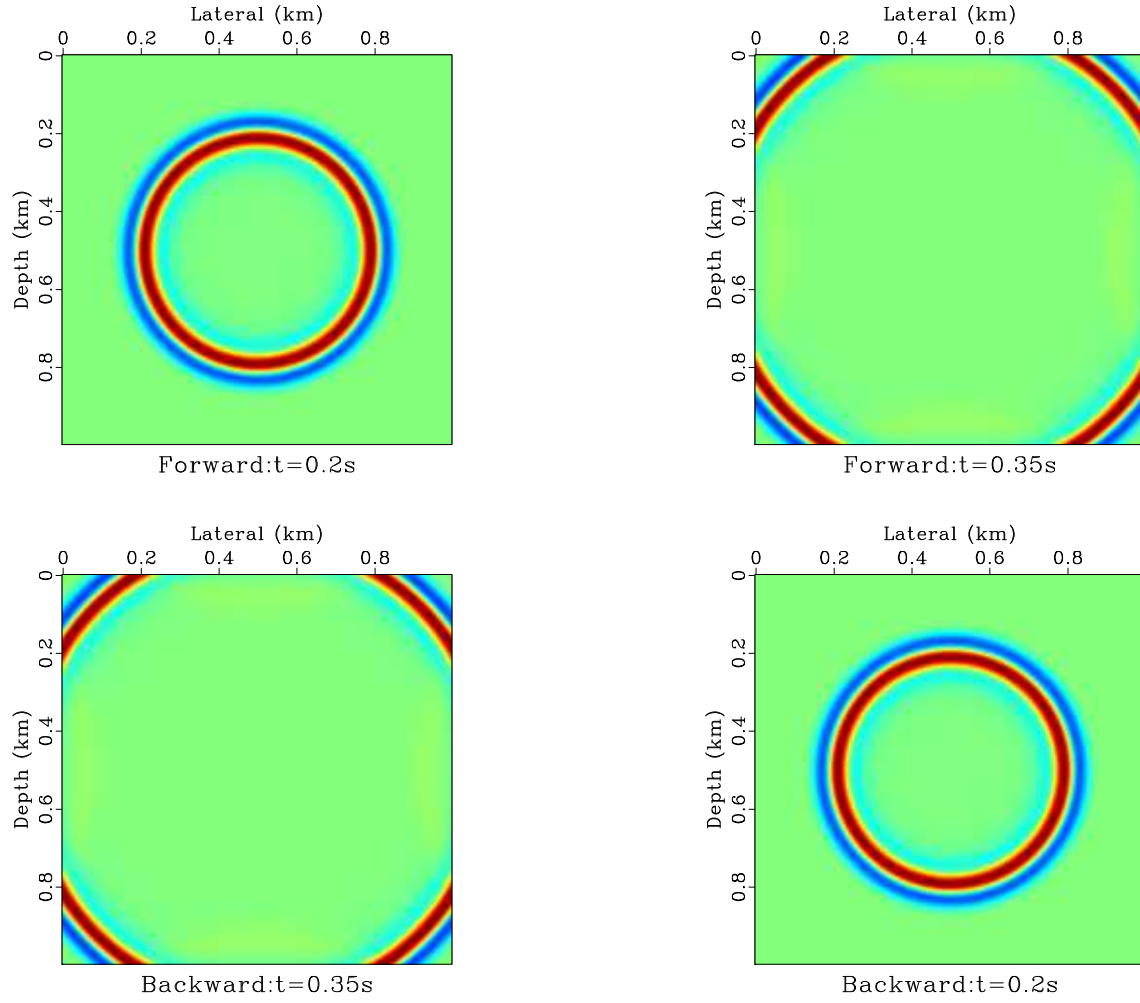
Figure 10: The forward modeled wavefield can be exactly reconstructed using effective boundary saving.
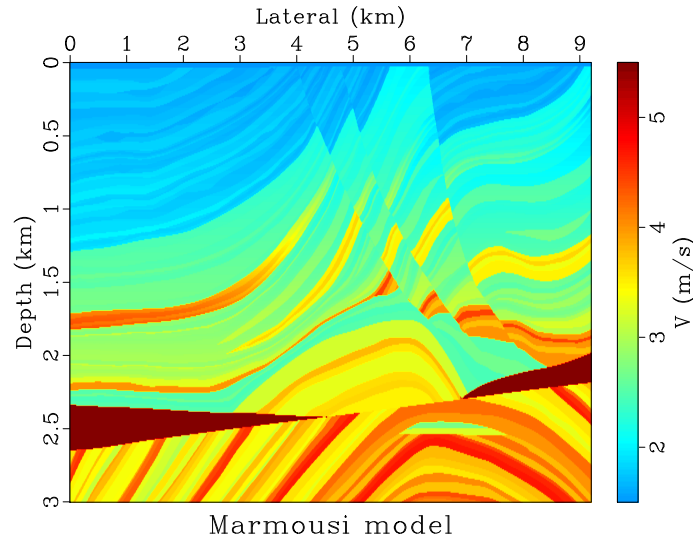
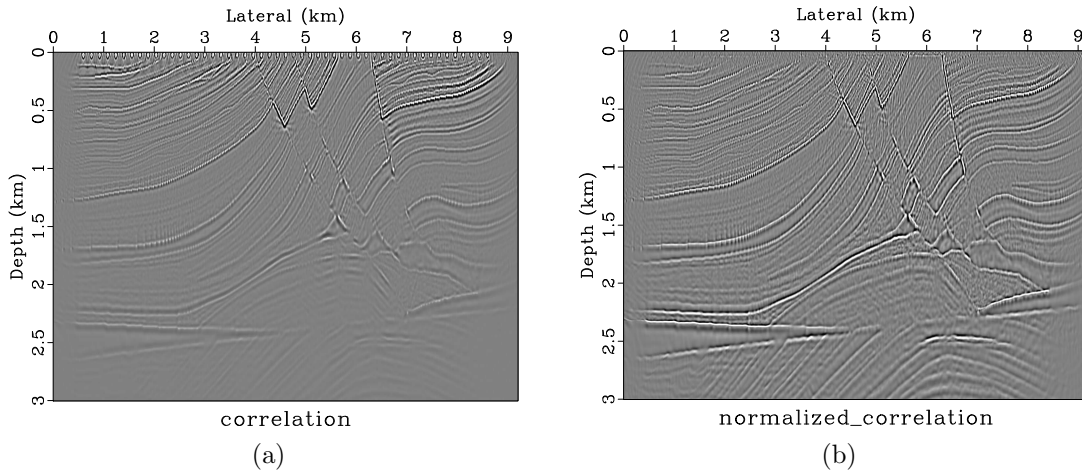Figure 11: The Marmousi velocity model.

Figure 12: RTM result of Marmousi model using effective boundary saving scheme (staggered grid finite difference). (a) Result of cross-correlation imaging condition. (b) Result of normalized cross-correlation imaging condition.
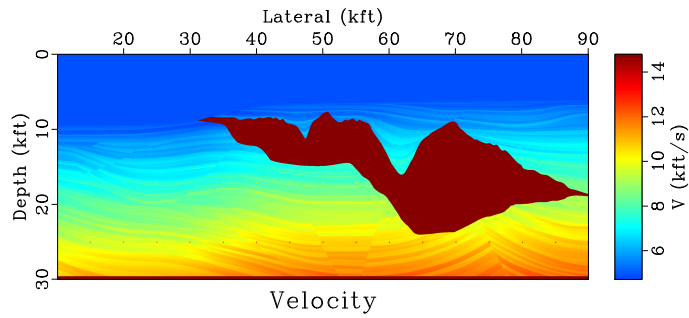

Figure 13: The Sigsbee velocity model.
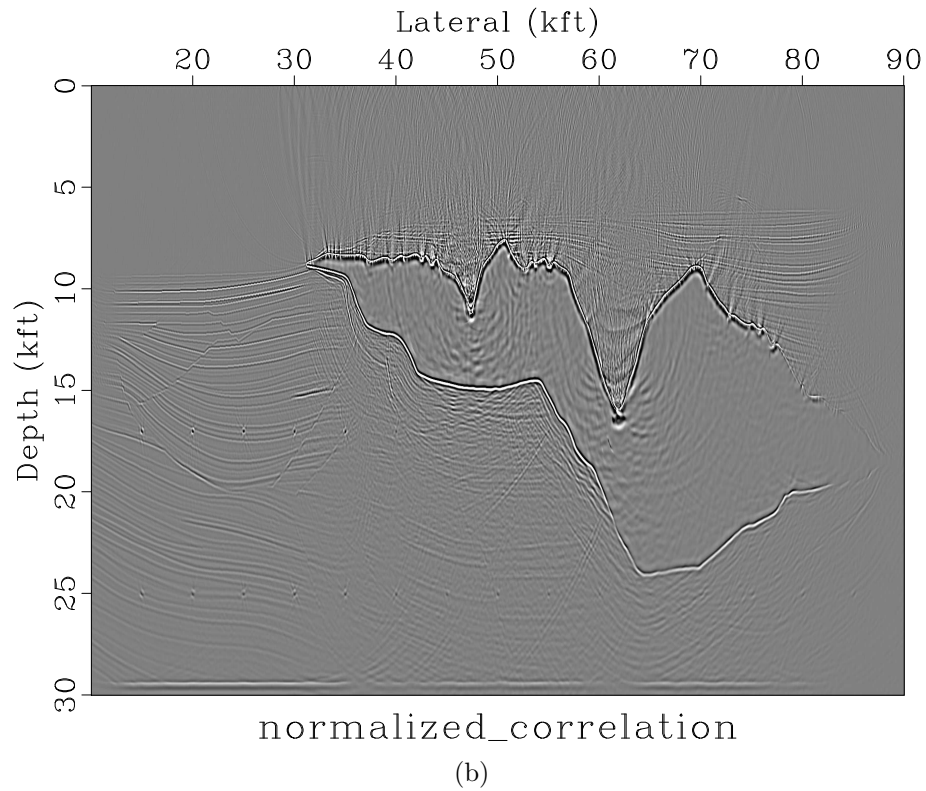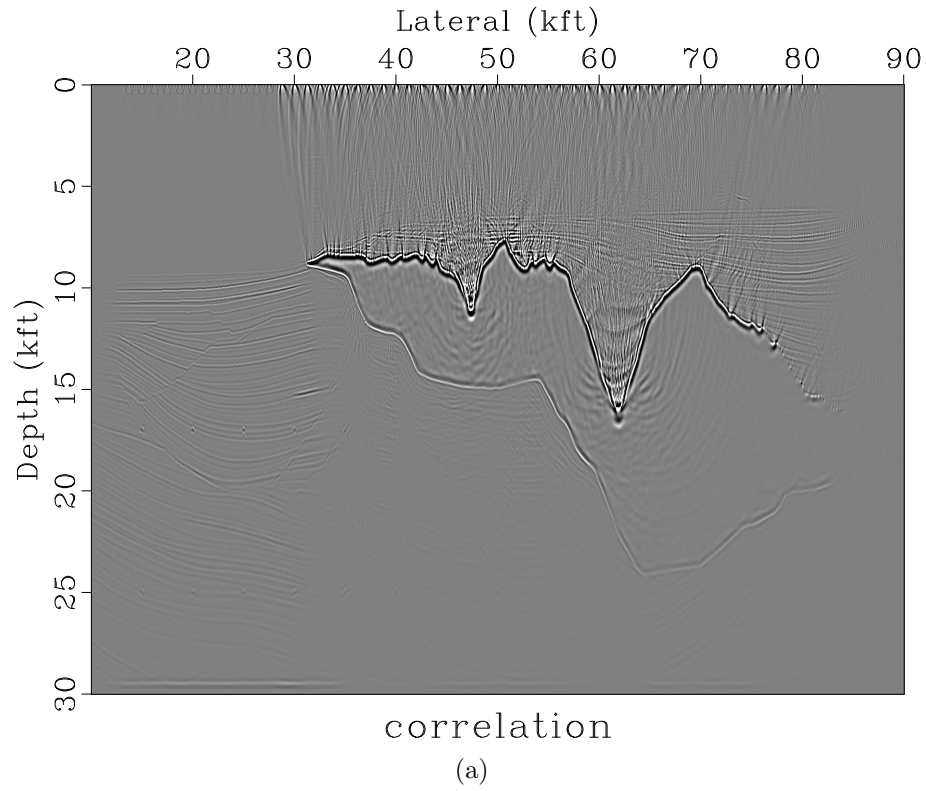
correlation

(a)

normalized_correlation

(b)

Figure 14: RTM result of Sigsbee model using effective boundary saving scheme (staggered grid finite difference). (a) Result of cross-correlation imaging condition. (b) Result of normalized cross-correlation imaging condition.

FULL WAVEFORM INVERSION (FWI)

Time domain FWI was proposed by Tarantola (1984), and developed in Tarantola (1986); Pica et al. (1990). Later, frequency domain FWI was proposed by Pratt et al. (1998). Actually, many authors call it full waveform tomography. (tomography=fwi, imaging=migration) Here, we mainly follow two well-documented paper Pratt et al. (1998) and Virieux and Operto (2009). We define the misfit vector $\Delta \mathbf{p} = \mathbf{p}_{cal} - \mathbf{p}_{obs}$ by the differences at the receiver positions between the recorded seismic data $\mathbf{p}_{obs}$ and the modelled seismic data $\mathbf{p}_{cal} = \mathbf{f}(\mathbf{m})$ for each source-receiver pair of the seismic survey. Here, in the simplest acoustic velocity inversion, $\mathbf{m}$ corresponds to the velocity model to be determined. The objective function taking the least-squares norm of the misfit vector $\Delta \mathbf{p}$ is given by

$$E(\mathbf{m}) = \frac{1}{2}\Delta \mathbf{p}^{\dagger}\Delta \mathbf{p} = \frac{1}{2}\Delta \mathbf{p}^{T}\Delta \mathbf{p}^{*} = \frac{1}{2}\sum_{r=1}^{ng}\sum_{s=1}^{ns}\int_{0}^{t_{\max}} dt |p_{cal}(\mathbf{x}_r, t; \mathbf{x}_s) - p_{obs}(\mathbf{x}_r, t; \mathbf{x}_s)|^2$$

(64)

where ns and ng are the number of sources and geophones, $\dagger$ denotes the adjoint and $*$ the complex conjugate, while $\mathbf{f}(\cdot)$ indicates the forward modeling of the wave propagation. The recorded seismic data is only a small subset of the whole wavefield.

The minimum of the misfit function $E(\mathbf{m})$ is sought in the vicinity of the starting model $\mathbf{m}_0$. FWI is essentially a local optimization. In the framework of the Born approximation, we assume that the updated model $\mathbf{m}$ of dimension M can be written as the sum of the starting model $\mathbf{m}_0$ plus a perturbation model $\Delta \mathbf{m}$: $\mathbf{m} = \mathbf{m}_0 + \Delta \mathbf{m}$. In the following, we assume that $\mathbf{m}$ is real valued.

A second-order Taylor-Lagrange development of the misfit function in the vicinity of $\mathbf{m}_0$ gives the expression

$$E(\mathbf{m}_0 + \Delta \mathbf{m}) = E(\mathbf{m}_0) + \sum_{i=1}^{M}\frac{\partial E(\mathbf{m}_0)}{\partial m_i}\Delta m_i + \frac{1}{2}\sum_{i=1}^{M}\sum_{j=1}^{M}\frac{\partial^2 E(\mathbf{m}_0)}{\partial m_i \partial m_j}\Delta m_i \Delta m_j + O(||\Delta \mathbf{m}||^3)$$

(65)

Taking the derivative with respect to the model parameter m_i results in

$$\frac{\partial E(\mathbf{m})}{\partial m_i} = \frac{\partial E(\mathbf{m}_0)}{\partial m_i} + \sum_{j=1}^{M}\frac{\partial^2 E(\mathbf{m}_0)}{\partial m_j \partial m_i}\Delta m_j, i = 1, 2, \ldots, M.$$

(66)

Briefly speaking, it is

$$\frac{\partial E(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial E(\mathbf{m}_0)}{\partial \mathbf{m}} + \frac{\partial^2 E(\mathbf{m}_0)}{\partial \mathbf{m}^2}\Delta \mathbf{m}$$

(67)

Thus,

$$\Delta \mathbf{m} = -\left(\frac{\partial^2 E(\mathbf{m}_0)}{\partial \mathbf{m}^2}\right)^{-1}\frac{\partial E(\mathbf{m}_0)}{\partial \mathbf{m}} = -\mathbf{H}^{-1}\nabla E_{\mathbf{m}}$$

(68)

where

$$\nabla E_{\mathbf{m}} = \frac{\partial E(\mathbf{m}_0)}{\partial \mathbf{m}} = \left[\frac{\partial E(\mathbf{m}_0)}{\partial m_1}, \frac{\partial E(\mathbf{m}_0)}{\partial m_2}, \dots, \frac{\partial E(\mathbf{m}_0)}{\partial m_M} \right]^T \tag{69}$$

and

$$\mathbf{H} = \frac{\partial^2 E(\mathbf{m}_0)}{\partial \mathbf{m}^2} = \left[\frac{\partial^2 E(\mathbf{m}_0)}{\partial m_i \partial m_j} \right] = \begin{bmatrix} \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_1^2} & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_1 m_2} & \cdots & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_1 m_M} \\ \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_2 m_1} & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_2^2} & \cdots & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_2 m_M} \\ \vdots & & \ddots & \vdots \\ \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_M m_1} & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_M m_2} & \cdots & \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_M^2} \end{bmatrix}. \tag{70}$$

$\nabla E_{\mathbf{m}}$ and $\mathbf{H}$ are the gradient vector and the Hessian matrix, respectively.

The Newton, Gauss-Newton, and steepest-descent methods

In terms of Eq. (64),

$$\begin{aligned} \frac{\partial E(\mathbf{m})}{\partial m_i} &= \frac{1}{2} \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int dt \left[\left(\frac{\partial p_{cal}}{\partial m_i} \right) (p_{cal} - p_{obs})^* + \left(\frac{\partial p_{cal}}{\partial m_i} \right)^* (p_{cal} - p_{obs}) \right] \\ &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int dt \mathtt{Re} \left[\left(\frac{\partial p_{cal}}{\partial m_i} \right)^* \Delta p \right] (\Delta p = p_{cal} - p_{obs}) \\ &= \mathtt{Re} \left[\left(\frac{\partial \mathbf{p}_{cal}}{\partial m_i} \right)^\dagger \Delta \mathbf{p} \right] = \mathtt{Re} \left[\left(\frac{\partial \mathbf{f}(\mathbf{m})}{\partial m_i} \right)^\dagger \Delta \mathbf{p} \right], i = 1, 2, \dots, M. \end{aligned} \tag{71}$$

That is to say,

$$\nabla E_{\mathbf{m}} = \nabla E(\mathbf{m}) = \frac{\partial E(\mathbf{m})}{\partial \mathbf{m}} = \mathtt{Re} \left[\left(\frac{\partial \mathbf{f}(\mathbf{m})}{\partial \mathbf{m}} \right)^\dagger \Delta \mathbf{p} \right] = \mathtt{Re} \left[\mathbf{J}^\dagger \Delta \mathbf{p} \right] \tag{72}$$

where $\mathtt{Re}$ takes the real part, and $\mathbf{J} = \frac{\partial \mathbf{p}_{cal}}{\partial \mathbf{m}} = \frac{\partial \mathbf{f}(\mathbf{m})}{\partial \mathbf{m}}$ is the Jacobian matrix, i.e., the sensitivity or the Fréchet derivative matrix.

Differentiation of the gradient expression (71) with respect to the model parameters gives the following expression for the Hessian $\mathbf{H}$:

$$\begin{aligned} \mathbf{H}_{i,j} &= \frac{\partial^2 E(\mathbf{m})}{\partial m_i \partial m_j} = \frac{\partial}{\partial m_j} \left(\frac{\partial E(\mathbf{m})}{\partial m_i} \right) = \frac{\partial}{\partial m_j} \mathtt{Re} \left[\left(\frac{\partial \mathbf{p}_{cal}}{\partial m_i} \right)^\dagger \Delta \mathbf{p} \right] \\ &= \frac{\partial}{\partial m_j} \mathtt{Re} \left[\left(\frac{\partial \mathbf{p}_{cal}}{\partial m_i} \right)^T \Delta \mathbf{p}^* \right] = \mathtt{Re} \left[\frac{\partial}{\partial m_j} \left(\frac{\partial \mathbf{p}_{cal}}{\partial m_i} \right)^T \Delta \mathbf{p}^* \right] + \mathtt{Re} \left[\frac{\partial \mathbf{p}_{cal}^\dagger}{\partial m_i} \frac{\partial \mathbf{p}_{cal}}{\partial m_j} \right] \end{aligned} \tag{73}$$

In matrix form

$$\mathbf{H} = \frac{\partial^2 E(\mathbf{m})}{\partial \mathbf{m}^2} = \mathtt{Re} \left[\mathbf{J}^\dagger \mathbf{J} \right] + \mathtt{Re} \left[\frac{\partial \mathbf{J}^T}{\partial \mathbf{m}^T} (\Delta \mathbf{p}^*, \Delta \mathbf{p}^*, \dots, \Delta \mathbf{p}^*) \right]. \tag{74}$$

In many cases, this second-order term is neglected for nonlinear inverse problems. In the following, the remaining term in the Hessian, i.e., $\mathbf{H}_a = \mathrm{Re}[\mathbf{J}^\dagger\mathbf{J}]$, is referred to as the approximate Hessian. It is the auto-correlation of the derivative wavefield. Eq. (68) becomes

$$\Delta\mathbf{m} = -\mathbf{H}^{-1}\nabla E_{\mathbf{m}} = -\mathbf{H}_a^{-1}\mathrm{Re}[\mathbf{J}^\dagger\Delta\mathbf{p}]. \tag{75}$$

The method which solves equation (74) when only $\mathbf{H}_a$ is estimated is referred to as the Gauss-Newton method. To guarantee th stability of the algorithm (avoiding the singularity), we can use $\mathbf{H} = \mathbf{H}_a + \eta\mathbf{I}$, leading to

$$\Delta\mathbf{m} = -\mathbf{H}^{-1}\nabla E_{\mathbf{m}} = -(\mathbf{H}_a + \eta\mathbf{I})^{-1}\mathrm{Re}\left[\mathbf{J}^\dagger\Delta\mathbf{p}\right]. \tag{76}$$

Alternatively, the inverse of the Hessian in Eq. (68) can be replaced by $\mathbf{H} = \mathbf{H}_a = \mu\mathbf{I}$, leading to the gradient or steepest-descent method:

$$\Delta\mathbf{m} = -\mu^{-1}\nabla E_{\mathbf{m}} = -\alpha\nabla E_{\mathbf{m}} = -\alpha\mathrm{Re}\left[\mathbf{J}^\dagger\Delta\mathbf{p}\right], \alpha = \mu^{-1}. \tag{77}$$

At the k-th iteration, the misfit function can be presented using the 2nd-order Taylor-Lagrange expansion

$$E(\mathbf{m}_{k+1}) = E(\mathbf{m}_k - \alpha_k\nabla E(\mathbf{m}_k)) = E(\mathbf{m}_k) - \alpha_k\langle\nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k)\rangle + \frac{1}{2}\alpha_k^2\nabla E(\mathbf{m}_k)^\dagger\mathbf{H}_k\nabla E(\mathbf{m}_k). \tag{78}$$

Setting $\frac{\partial E(\mathbf{m}_{k+1})}{\partial\alpha_k} = 0$ gives

$$\alpha_k = \frac{\nabla E(\mathbf{m}_k)^\dagger\nabla E(\mathbf{m}_k)}{\nabla E(\mathbf{m}_k)^\dagger\mathbf{H}_k\nabla E(\mathbf{m}_k)} \overset{\mathbf{H}_k := \mathbf{H}_a = \mathbf{J}_k^\dagger\mathbf{J}_k}{=\!=\!=} \frac{\nabla E(\mathbf{m}_k)^\dagger\nabla E(\mathbf{m}_k)}{\langle\mathbf{J}_k\nabla E(\mathbf{m}_k), \mathbf{J}_k\nabla E(\mathbf{m}_k)\rangle} \tag{79}$$

Conjugate gradient (CG) implementation

The gradient-like method can be summarized as

$$\mathbf{m}_{k+1} = \mathbf{m}_k + \alpha_k\mathbf{d}_k. \tag{80}$$

The conjugate gradient (CG) algorithm decreases the misfit function along the conjugate gradient direction:

$$\mathbf{d}_k = \begin{cases} -\nabla E(\mathbf{m}_0), & k = 0 \\ -\nabla E(\mathbf{m}_k) + \beta_k\mathbf{d}_{k-1}, & k \geq 1 \end{cases} \tag{81}$$

There are many ways to compute β_k:

$$
\begin{cases}
\beta_k^{HS} = \dfrac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle} \\[12pt]
\beta_k^{FR} = \dfrac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \nabla E(\mathbf{m}_{k-1}), \nabla E(\mathbf{m}_{k-1}) \rangle} \\[12pt]
\beta_k^{PRP} = \dfrac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle}{\langle \nabla E(\mathbf{m}_{k-1}), \nabla E(\mathbf{m}_{k-1}) \rangle} \\[12pt]
\beta_k^{CD} = -\dfrac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_{k-1}) \rangle} \\[12pt]
\beta_k^{DY} = \dfrac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle}
\end{cases}
\tag{82}
$$

To achieve best convergence rate, in practice we suggest to use a hybrid scheme combing Hestenes-Stiefel and Dai-Yuan:

$$
\beta_k = \max(0, \min(\beta_k^{HS}, \beta_k^{DY})).
\tag{83}
$$

Iterating with Eq. (80) needs to find an appropriate α_k. Here we provide two approaches to calculate α_k.

- Approach 1: Currently, the objective function is

$$
E(\mathbf{m}_{k+1}) = E(\mathbf{m}_k + \alpha_k \mathbf{d}_k) = E(\mathbf{m}_k) + \alpha_k \langle \nabla E(\mathbf{m}_k), \mathbf{d}_k \rangle + \frac{1}{2}\alpha_k^2 \mathbf{d}_k^\dagger \mathbf{H}_k \mathbf{d}_k.
\tag{84}
$$

Setting $\frac{\partial E(\mathbf{m}_{k+1})}{\partial \alpha_k} = 0$ gives

$$
\alpha_k = -\frac{\langle \mathbf{d}_k, \nabla E(\mathbf{m}_k) \rangle}{\mathbf{d}_k^\dagger \mathbf{H}_k \mathbf{d}_k} \overset{\mathbf{H}_k := \mathbf{H}_a = \mathbf{J}_k^\dagger \mathbf{J}_k}{=} -\frac{\langle \mathbf{d}_k, \nabla E(\mathbf{m}_k) \rangle}{\langle \mathbf{J}_k \mathbf{d}_k, \mathbf{J}_k \mathbf{d}_k \rangle}.
\tag{85}
$$

- Approach 2: Recall that

$$
\mathbf{f}(\mathbf{m}_k + \alpha_k \mathbf{d}_k) = \mathbf{f}(\mathbf{m}_k) + \frac{\partial \mathbf{f}(\mathbf{m}_k)}{\partial \mathbf{m}}\mathbf{d}_k + O(||\mathbf{d}_k||^2) = \mathbf{f}(\mathbf{m}_k) + \alpha_k \mathbf{J}_k \mathbf{d}_k + O(||\mathbf{d}_k||^2).
\tag{86}
$$

Using the 1st-order approximation, we have

$$
\begin{aligned}
E(\mathbf{m}_{k+1}) &= \frac{1}{2}||\mathbf{f}(\mathbf{m}_k + \alpha_k \mathbf{d}_k) - \mathbf{p}_{obs}||^2 \\
&\approx \frac{1}{2}||\mathbf{f}(\mathbf{m}_k) + \alpha_k \mathbf{J}_k \mathbf{d}_k - \mathbf{p}_{obs}||^2 = \frac{1}{2}||\mathbf{f}(\mathbf{m}_k) - \mathbf{p}_{obs} + \alpha_k \mathbf{J}_k \mathbf{d}_k||^2 \\
&= E(\mathbf{m}) + \alpha_k \langle \mathbf{J}_k \mathbf{d}_k, \mathbf{f}(\mathbf{m}_k) - \mathbf{p}_{obs} \rangle + \frac{1}{2}\alpha_k^2 \langle \mathbf{J}_k \mathbf{d}_k, \mathbf{J}_k \mathbf{d}_k \rangle.
\end{aligned}
\tag{87}
$$

Setting $\frac{\partial E(\mathbf{m}_{k+1})}{\partial \alpha_k} = 0$ gives

$$
\alpha_k = \frac{\langle \mathbf{J}_k \mathbf{d}_k, \mathbf{p}_{obs} - \mathbf{f}(\mathbf{m}_k) \rangle}{\langle \mathbf{J}_k \mathbf{d}_k, \mathbf{J}_k \mathbf{d}_k \rangle}.
\tag{88}
$$

In fact, Eq. (88) can also be obtained from Eq. (85) in terms of Eq. (72): $\nabla E_{\mathbf{m}} = \mathbf{J}^\dagger \Delta \mathbf{p}$.

In terms of Eq. (86), the term $\mathbf{J}_k \mathbf{d}_k$ is computed conventionally using a 1st-order-accurate finite difference approximation of the partial derivative of $\mathbf{f}$:

$$\mathbf{J}_k \mathbf{d}_k = \frac{\mathbf{f}(\mathbf{m}_k + \epsilon \mathbf{d}_k) - \mathbf{f}(\mathbf{m}_k)}{\epsilon} \tag{89}$$

with a small parameter ϵ. In practice, we chose an ϵ such that

$$\max(\epsilon |\mathbf{d}_k|) \leqslant \frac{\max(|\mathbf{m}_k|)}{100}. \tag{90}$$

Fréchet derivative

Recall that the basic acoustic wave equation can be specified as

$$\frac{1}{v^2(\mathbf{x})} \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} - \nabla^2 p(\mathbf{x}, t; \mathbf{x}_s) = f_s(\mathbf{x}, t; \mathbf{x}_s). \tag{91}$$

where $f_s(\mathbf{x}, t; \mathbf{x}_s) = f(t')\delta(\mathbf{x} - \mathbf{x}_s)\delta(t - t')$. The Green's function $\Gamma(\mathbf{x}, t; \mathbf{x}_s, t')$ is defined by

$$\frac{1}{v^2(\mathbf{x})} \frac{\partial^2 \Gamma(\mathbf{x}, t; \mathbf{x}_s, t')}{\partial t^2} - \nabla^2 \Gamma(\mathbf{x}, t; \mathbf{x}_s, t') = \delta(\mathbf{x} - \mathbf{x}_s)\delta(t - t'). \tag{92}$$

Thus the integral representation of the solution can be given by

$$\begin{aligned}
p(\mathbf{x}_r, t; \mathbf{x}_s) &= \int_V d\mathbf{x} \int dt' \Gamma(\mathbf{x}_r, t; \mathbf{x}, t') f(\mathbf{x}, t'; \mathbf{x}_s) \\
&= \int_V d\mathbf{x} \int dt' \Gamma(\mathbf{x}_r, t - t'; \mathbf{x}, 0) f(\mathbf{x}, t'; \mathbf{x}_s) \text{(Causility of Green's function)} \\
&= \int_V d\mathbf{x} \Gamma(\mathbf{x}_r, t; \mathbf{x}, 0) * f(\mathbf{x}, t; \mathbf{x}_s)
\end{aligned} \tag{93}$$

where $*$ denotes the convolution operator.

A perturbation $v(\mathbf{x}) \to v(\mathbf{x}) + \Delta v(\mathbf{x})$ will produce a field $p(\mathbf{x}, t; \mathbf{x}_s) + \Delta p(\mathbf{x}, t; \mathbf{x}_s)$ defined by

$$\frac{1}{(v(\mathbf{x}) + \Delta v(\mathbf{x}))^2} \frac{\partial^2 [p(\mathbf{x}, t; \mathbf{x}_s) + \Delta p(\mathbf{x}, t; \mathbf{x}_s)]}{\partial t^2} - \nabla^2 [p(\mathbf{x}, t; \mathbf{x}_s) + \Delta p(\mathbf{x}, t; \mathbf{x}_s)] = f_s(\mathbf{x}, t; \mathbf{x}_s) \tag{94}$$

Note that

$$\frac{1}{(v(\mathbf{x}) + \Delta v(\mathbf{x}))^2} = \frac{1}{v^2(\mathbf{x})} - \frac{2\Delta v(\mathbf{x})}{v^3(\mathbf{x})} + O(\Delta^2 v(\mathbf{x})) \tag{95}$$

Eq. (94) subtracts Eq. (91), yielding

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 \Delta p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2} - \nabla^2 \Delta p(\mathbf{x},t;\mathbf{x}_s) = \frac{\partial^2[p(\mathbf{x},t;\mathbf{x}_s) + \Delta p(\mathbf{x},t;\mathbf{x}_s)]}{\partial t^2}\frac{2\Delta v(\mathbf{x})}{v^3(\mathbf{x})} \quad (96)$$

Using the Born approximation, Eq. (96) becomes

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 \Delta p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2} - \nabla^2 \Delta p(\mathbf{x},t;\mathbf{x}_s) = \frac{\partial^2 p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2\Delta v(\mathbf{x})}{v^3(\mathbf{x})} \quad (97)$$

Again, based on integral representation, we obtain

$$\Delta p(\mathbf{x}_r,t;\mathbf{x}_s) = \int_V d\mathbf{x}\Gamma(\mathbf{x}_r,t;\mathbf{x},0) * \frac{\partial^2 p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2\Delta v(\mathbf{x})}{v^3(\mathbf{x})}. \quad (98)$$

Gradient computation

According to the previous section, it follows that

$$\frac{\partial p_{cal}}{\partial v_i(\mathbf{x})} = \int_V d\mathbf{x}\Gamma(\mathbf{x}_r,t;\mathbf{x},0)*\ddot{p}(\mathbf{x},t;\mathbf{x}_s)\frac{2}{v^3(\mathbf{x})} = \int_V d\mathbf{x}\Gamma(\mathbf{x}_r,t;\mathbf{x},0)*\frac{\partial^2 p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2}{v^3(\mathbf{x})}. \quad (99)$$

The convolution guarantees

$$\int dt[g(t) * f(t)]h(t) = \int dt f(t)[g(-t) * h(t)]. \quad (100)$$

Then, Eq. (71) becomes

$$\frac{\partial E(\mathbf{m})}{\partial m_i} = \sum_{r=1}^{ng}\sum_{s=1}^{ns}\int dt \text{Re}\left[\left(\frac{\partial p_{cal}}{\partial m_i}\right)^* \Delta p\right] (\Delta p = p_{cal} - p_{obs})$$

$$= \sum_{r=1}^{ng}\sum_{s=1}^{ns}\int_0^{t_{max}} dt \text{Re}\left[\left(\int_V d\mathbf{x}\Gamma(\mathbf{x}_r,t;\mathbf{x},0) * \frac{\partial^2 p(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2}{v^3(\mathbf{x})}\right)^* \Delta p(\mathbf{x}_r,t;\mathbf{x}_s)\right]$$

$$= \sum_{r=1}^{ng}\sum_{s=1}^{ns}\int_0^{t_{max}} dt \text{Re}\left[\left(\frac{\partial^2 p_{cal}(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2}{v^3(\mathbf{x})}\right)^* (\int_V d\mathbf{x}\Gamma(\mathbf{x}_r,-t;\mathbf{x},0) * \Delta p(\mathbf{x}_r,t;\mathbf{x}_s))\right]$$

$$= \sum_{r=1}^{ng}\sum_{s=1}^{ns}\int_0^{t_{max}} dt \text{Re}\left[\left(\frac{\partial^2 p_{cal}(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2}{v^3(\mathbf{x})}\right)^* (\int_V d\mathbf{x}\Gamma(\mathbf{x}_r,0;\mathbf{x},t) * \Delta p(\mathbf{x}_r,t;\mathbf{x}_s))\right]$$

$$= \sum_{r=1}^{ng}\sum_{s=1}^{ns}\int_0^{t_{max}} dt \text{Re}\left[\left(\frac{\partial^2 p_{cal}(\mathbf{x},t;\mathbf{x}_s)}{\partial t^2}\frac{2}{v^3(\mathbf{x})}\right)^* p_{res}(\mathbf{x}_r,t;\mathbf{x}_s)\right]$$

$$(101)$$

where $p_{res}(\mathbf{x},t;\mathbf{x}_s)$ is a time-reversal wavefield produced using the residual $\Delta p(\mathbf{x}_r,t;\mathbf{x}_s)$ as the source. It follows from reciprocity theorem

$$p_{res}(\mathbf{x},t;\mathbf{x}_s) = \int_V d\mathbf{x}\Gamma(\mathbf{x}_r,0;\mathbf{x},t) * \Delta p(\mathbf{x}_r,t;\mathbf{x}_s) = \int_V d\mathbf{x}\Gamma(\mathbf{x},0;\mathbf{x}_r,t) * \Delta p(\mathbf{x}_r,t;\mathbf{x}_s). \quad (102)$$

satisfying

$$\frac{1}{v^2(\mathbf{x})}\frac{\partial^2 p_{res}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} - \nabla^2 p_{res}(\mathbf{x}, t; \mathbf{x}_s) = \Delta p(\mathbf{x}_r, t; \mathbf{x}_s). \tag{103}$$

It is noteworthy that an input f and the system impulse response function g are exchangeable in convolution. That is to say, we can use the system impulse response function g as the input, the input f as the impulse response function, leading to the same output. In the seismic modeling and acquisition process, the same seismogram can be obtained when we shot at the receiver position $\mathbf{x}_r$ when recording the seismic data at the position $\mathbf{x}$.

Numerical results

I use the Marmousi model for the benchmark test, as shown in the top panel of Figure 15. FWI tacitly requires a good starting model incorporated with low frequency information. Therefore, we use a starting model (bottom panel of Figure 15) obtained by smoothing the original model 20 times with a 5x5 window.

The FWI is carried out for 300 iterations. We record all the updated velocity to make sure the velocity refinement is going on during the iterative procedure. The updated velocity model at the iteration 1, 20, 50, 100, 180 and 300 are displyed in Figure 17. Figure 18 presents the decreasing misfit function in iterations. As can be seen from the Figures 17 and 18, the velocity model changes significantly at the early stage. Later iterations in FWI make some improvement on small details for the velocity model.

ACKNOWLEDGEMENT

REFERENCES

Baysal, E., D. D. Kosloff, and J. W. Sherwood, 1983, Reverse time migration: Geophysics, **48**, 1514–1524.

Berenger, J.-P., 1994, A perfectly matched layer for the absorption of electromagnetic waves: Journal of computational physics, **114**, 185–200.

Biondi, B., 2006, 3d seismic imaging: Society of Exploration Geophysicists.

Bjorck, A., 1996, Numerical methods for least squares problems: Society for Industrial and Applied Mathematics.

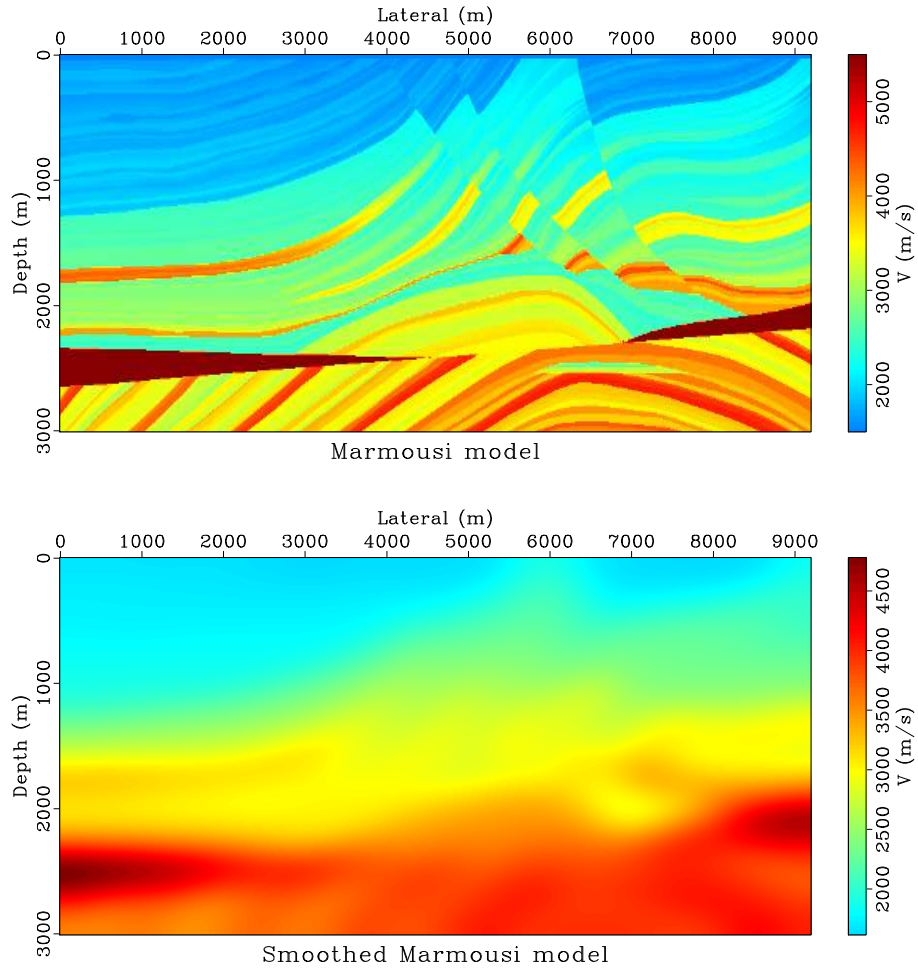Carcione, J. M., G. C. Herman, and A. Ten Kroode, 2002, Seismic modeling: Geophysics, **67**, 1304–1325.

Figure 15: Top: The original Marmousi is downsampled with a factor of 3 along depth and lateral direction. The shots are generated according to the subsampled Marmousi model. Bottom: The starting model of FWI for Marmousi model, which is obtained by smoothing the original model 20 times with a 5x5 window.
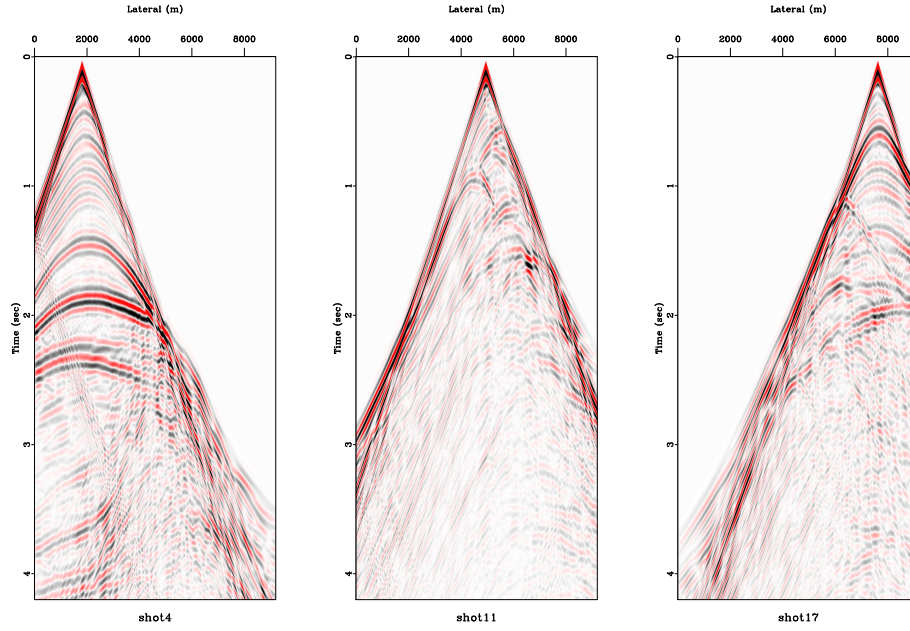
Figure 16: 21 shots were deployed in the FWI. Here, shot 4, 11 and 17 are shown from left to right
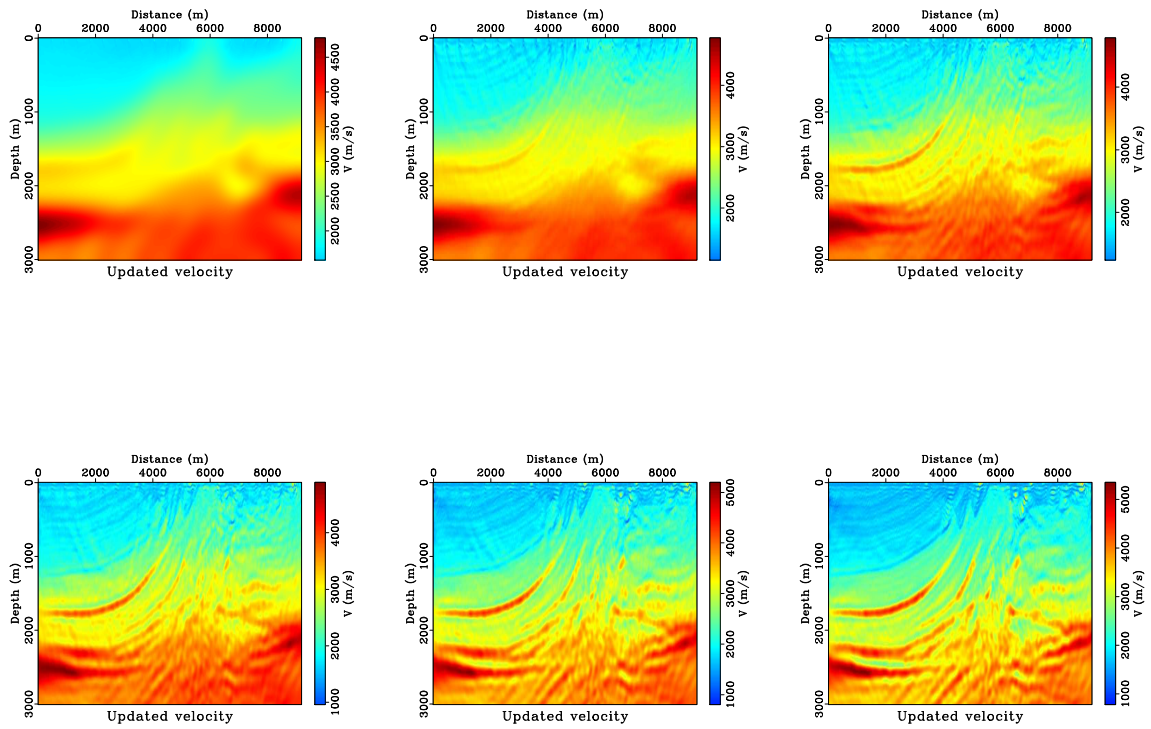

Figure 17: The updated velocity model at the iteration 1, 20, 50, 100, 180 and 300.
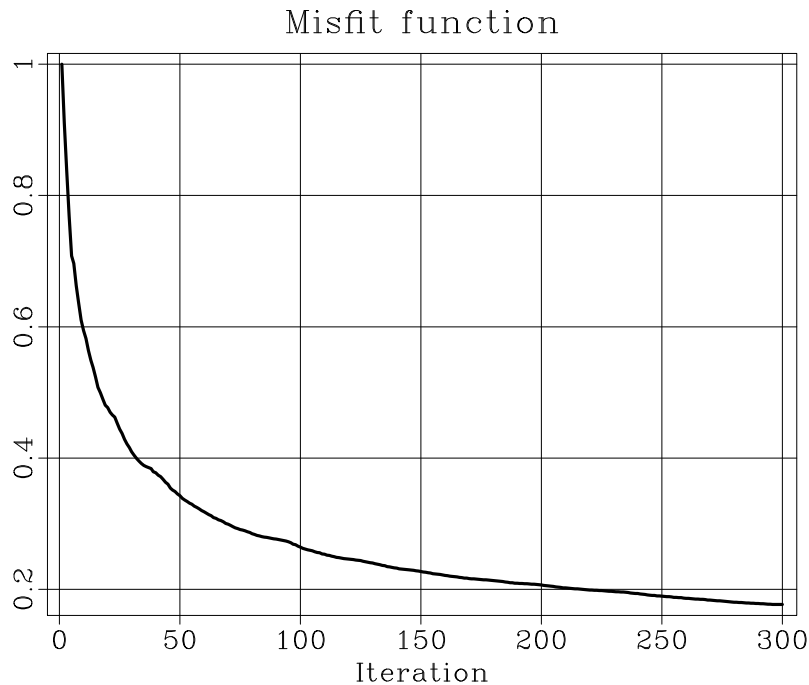
Misfit function

Figure 18: The misfit function decreases with the iterations.

Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: Geophysics, **50**, 705–708.

Claerbout, J., 1986, Imaging the earth's interior: Geophysical Journal of the Royal Astronomical Society, **86**, 217–217.

Claerbout, J. F., 1971, Toward a unified theory of reflector mapping: Geophysics, **36**, 467–481.

Clapp, R. G., 2009, Reverse time migration with random boundaries: 79th Annual International Meeting, SEG Expanded Abstracts, 2809–2813.

Clapp, R. G., H. Fu, and O. Lindtjorn, 2010, Selecting the right hardware for reverse time migration: The Leading Edge, **29**, 48–58.

Clayton, R., and B. Engquist, 1977, Absorbing boundary conditions for acoustic and elastic wave equations: Bulletin of the Seismological Society of America, **67**, 1529–1540.

Collino, F., and C. Tsogka, 2001, Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media: Geophysics, **66**, 294–307.

DiMarco, S., R. Reid, A. Jochens, W. Nowlin Jr, M. Howard, et al., 2001, General characteristics of currents in the deepwater gulf of mexico: Presented at the Offshore Technology Conference, Offshore Technology Conference.

Dussaud, E., W. W. Symes, P. Williamson, L. Lemaistre, P. Singer, B. Denel, and A. Cherrett, 2008, Computational strategies for reverse-time migration: SEG Annual meeting.

Engquist, B., and A. Majda, 1977, Absorbing boundary conditions for numerical

simulation of waves: Proceedings of the National Academy of Sciences, **74**, 1765–1766.

Guitton, A., B. Kaelin, and B. Biondi, 2006, Least-squares attenuation of reverse-time-migration artifacts: Geophysics, **72**, S19–S23.

Guitton, A., A. Valenciano, D. Bevc, and J. Claerbout, 2007, Smoothing imaging condition for shot-profile migration: Geophysics, **72**, S149–S154.

Komatitsch, D., and R. Martin, 2007, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation: Geophysics, **72**, SM155–SM167.

Liu, G., Y. Liu, L. Ren, and X. Meng, 2013a, 3d seismic reverse time migration on gpgpu: Computers & Geosciences, **59**, 17 – 23.

Liu, H., R. Ding, L. Liu, and H. Liu, 2013b, Wavefield reconstruction methods for reverse time migration: Journal of Geophysics and Engineering, **10**, 015004.

McMechan, G., 1983, Migration by extrapolation of time-dependent boundary values: Geophysical Prospecting, **31**, 413–420.

Pica, A., J. Diet, and A. Tarantola, 1990, Nonlinear inversion of seismic reflection data in a laterally invariant medium: Geophysics, **55**, 284–292.

Pratt, G., C. Shin, et al., 1998, Gauss–newton and full newton methods in frequency–space seismic waveform inversion: Geophysical Journal International, **133**, 341–362.

Roden, J. A., and S. D. Gedney, 2000, Convolutional pml (cpml): An efficient fdtd implementation of the cfs-pml for arbitrary media: Microwave and optical technology letters, **27**, 334–338.

Symes, W. W., 2007, Reverse time migration with optimal checkpointing: Geophysics, **72**, SM213–SM221.

Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: Geophysics, **49**, 1259–1266.

——, 1986, A strategy for nonlinear elastic inversion of seismic reflection data: Geophysics, **51**, 1893–1903.

Versteeg, R., 1994, The marmousi experience: Velocity model determination on a synthetic complex data set: The Leading Edge, **13**, 927–936.

Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: Geophysics, **74**, WCC1–WCC26.

Yang, P., J. Gao, and B. Wang, 2014, {RTM} using effective boundary saving: A staggered grid {GPU} implementation: Computers & Geosciences.

Yoon, K., and K. J. Marfurt, 2006, Reverse-time migration using the poynting vector: Exploration Geophysics, **37**, 102–107.

Yoon, K., C. Shin, S. Suh, L. R. Lines, and S. Hong, 2003, 3d reverse-time migration using the acoustic wave equation: An experience with the seg/eage data set: The Leading Edge, **22**, 38–41.