

See discussions, stats, and author profiles for this publication at: <https://www.researchgate.net/publication/262936022>

A numerical tour of wave propagation

Article · May 2014

CITATIONS

2

READS

1,210

1 author:



[Pengliang Yang](#)

Harbin Institute of Technology

36 PUBLICATIONS 517 CITATIONS

[SEE PROFILE](#)

Some of the authors of this publication are also working on these related projects:



Downsampling Plus Interpolation for Wavefield Reconstruction by Reverse Propagation [View project](#)

A numerical tour of wave propagation

Pengliang Yang
E-mail: ypl.2100@gmail.com

May 1, 2014

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 delineation summarized in this tutorial has been implemented in Madagascar with Matlab, C and CUDA programming, which will benefit readers' further study.

Contents

1	Basic wave equation	1
1.1	Acoustic wave equation	2
1.2	Elastic wave equation	2
2	Forward modeling	5
2.1	Taylor and Páde expansion	5
2.2	Approximate the wave equation	6
2.3	Absorbing boundary condition (ABC)	6
2.3.1	Clayton-Enquist boundary condition	6
2.3.2	Sponge ABC	6
2.4	Perfectly Matched Layer (PML)	8
2.4.1	Split PML (SPML) for acoustics	8
2.4.2	Nonsplit Convolutional-PML (CPML) for acoustics	9
2.4.3	Nonsplit PML (NPML) for elastics	10
2.5	Discretization	10
2.5.1	Higher-order approximation of staggered-grid finite difference	11
2.5.2	Discretization of SPML	13
2.5.3	Discretization of NPML	14
3	Reverse time migration (RTM)	14
3.1	Brief overview	14
3.2	RTM implementation	15
3.3	Imaging condition	15
3.4	Computation strategies and boundary saving	15
3.5	Numerical examples	17
4	Full waveform inversion (FWI)	18
4.1	The Newton, Gauss-Newton, and steepest-descent methods	21
4.2	Conjugate gradient (CG) implementation	22
4.3	Fréchet derivative	23
4.4	Gradient computation	24

1 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

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

- Constitutive law

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

1.1 Acoustic wave equation

Acoustics is a special case of gas dynamics (sound waves in gases and fluids) and linear elastodynamics. Note that elastodynamics is a more accurate representation of earth dynamics, but more 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. t . Substituting Eq. (2) into the differentiation of Eq. (1) gives

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

We introduce $v(\mathbf{x}) = \sqrt{\frac{\kappa(\mathbf{x})}{\rho(\mathbf{x})}}$ (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 ($\rho(\mathbf{x}) := \rho$), we obtain the 2nd-order equation

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

where $\nabla^2 = \nabla \cdot \nabla = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2}$. In 2D case, it is

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

where $f_s(\mathbf{x}, t; \mathbf{x}_s) = \rho v^2(\mathbf{x}) \frac{\partial \tilde{S}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t}$. A snap 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

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

We use the Hess VTI model shown in Figure 3 and Figure 4. We perform 1000 steps of modeling with time interval $\Delta t = 0.001s$, and capture the wavefield at $kt = 0.9s$, as shown in Figure 5.

1.2 Elastic wave equation

In elastic wave equation, the modulus $\kappa(\mathbf{x})$ corresponds to two Lamé parameters: $\lambda + 2\mu = \rho v_p^2$ and $\mu = \rho v_s^2$, in which v_p and v_s denote the P- and S-wave velocity. The elastic wave equation can be written as

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

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 6, in which the grid size is 200x200, the spatial interval is $\Delta x = \Delta z = 5m$, and the velocities are chosen to be $V_p = 2km/s$, $V_s = V_p/\sqrt{2}$.

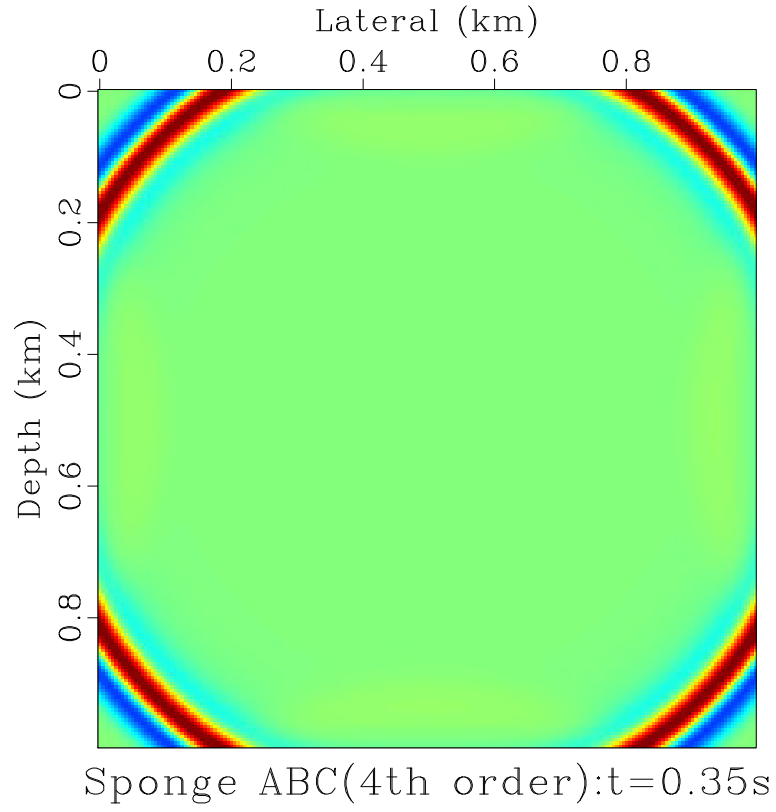


Figure 1: A snap of acoustic wavefield obtained at $t=0.35s$ with 4-th order finite difference scheme and the sponge absorbing boundary condition.

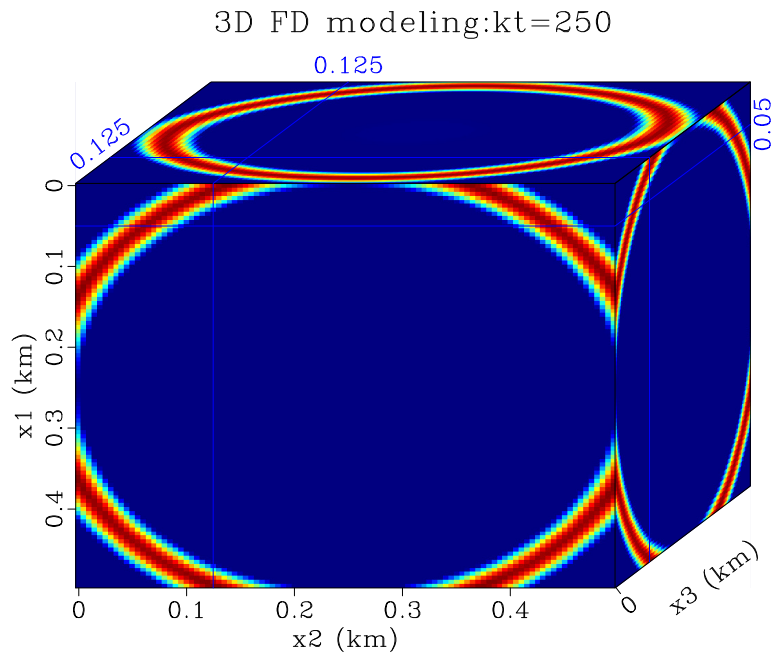


Figure 2: A wavefield snap recorded at $kt=250$, 300 steps modeled.

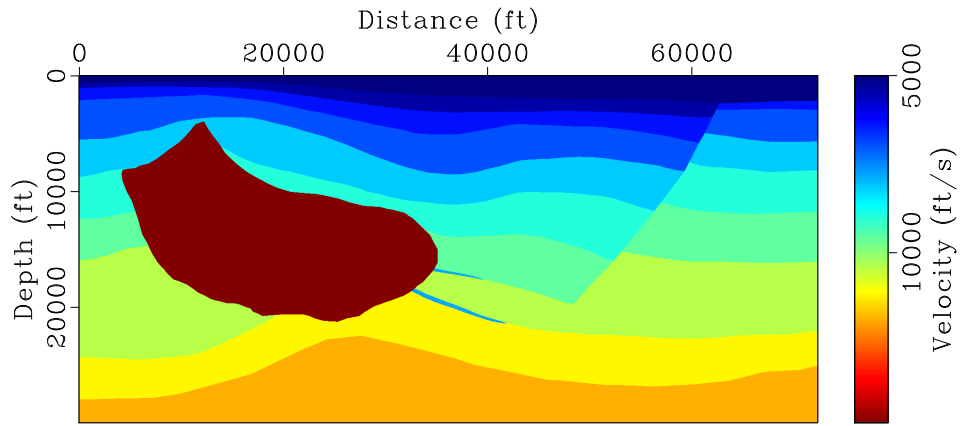


Figure 3: Hess VTI model: velocity component 1.

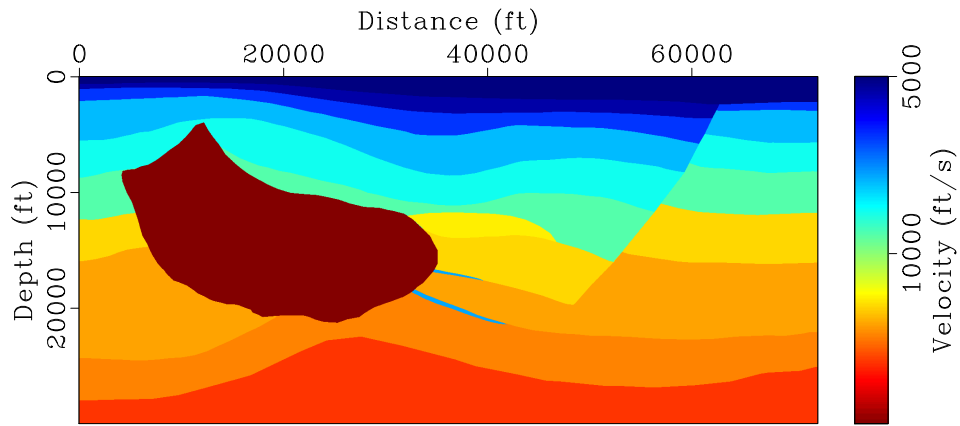


Figure 4: Hess VTI model: velocity component 2.

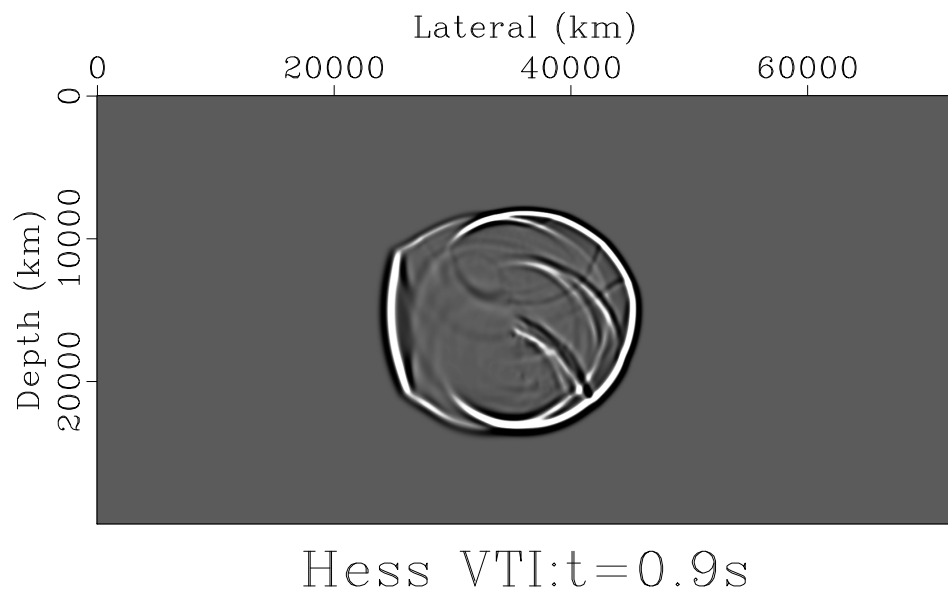


Figure 5: Wavefield at $kt = 0.9s$, 1000 steps of modeling with time interval $\Delta t = 0.001s$ performed.

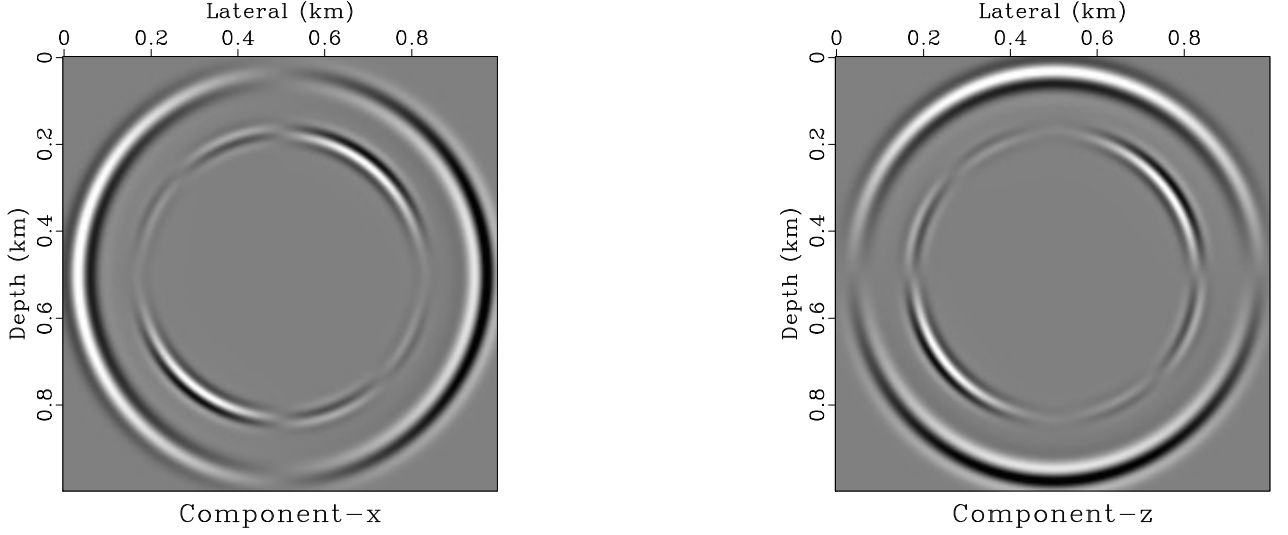


Figure 6: Two components of elastic wave propagation at $kt = 270$, $nt = 300$ modeled with $\Delta t = 0.001$. Grid size=200x200, $\Delta x = \Delta z = 5m$, $V_p = 2km/s$, $V_s = V_p/\sqrt{2}$

2 Forward modeling

2.1 Taylor and Páde expansion

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

$$f(x+h) = f(x) + \frac{\partial f(x)}{\partial x}h + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2}h^2 + \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3}h^3 + \dots \quad (10)$$

A popular example is

$$(1+x)^\alpha = 1 + \alpha x + \frac{\alpha(\alpha-1)}{2!}x^2 + \dots + \frac{\alpha(\alpha-1)\dots(\alpha-n+1)}{n!}x^n + \dots \quad (11)$$

Here we mainly consider the following expansion formula:

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

The Páde expansion of Eq. (12) should be:

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

we provide an informal derivation:

$$\begin{aligned} y = (1-x)^{\frac{1}{2}} &\Rightarrow x = 1 - y^2 = (1-y)(1+y) \Rightarrow 1-y = \frac{x}{1+y} \\ y &= 1 - \frac{x}{1+y} = 1 - \frac{x}{1 + (1 - \frac{x}{1+y})} = 1 - \frac{x/2}{1 - \frac{x/2}{1+y}} \\ &= 1 - \frac{x/2}{1 - \frac{x/2}{2 - \frac{x}{1+y}}} = 1 - \frac{x/2}{1 - \frac{x/4}{1 - \frac{x/2}{1+y}}} = \dots \end{aligned}$$

The 1st-order Páde expansion is:

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

The 2nd-order Páde expansion is:

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

And the 3rd-order one is:

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

2.2 Approximate the wave equation

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

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

$$\frac{\omega^2}{v^2} = k_x^2 + k_z^2. \quad (17)$$

The down-going wave equation in Fourier domain is

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

Using the different order Páde expansions, we have:

$$\left\{ \begin{array}{l} \text{1st - order : } k_z = \frac{\omega}{v} \left(1 - \frac{v^2 k_x^2}{2\omega^2} \right) \\ \text{2nd - order : } k_z = \frac{\omega}{v} \left(1 - \frac{\frac{2v^2 k_x^2}{\omega^2}}{4 - \frac{v^2 k_x^2}{\omega^2}} \right) \\ \text{3rd - order : } k_z = \frac{\omega}{v} \left(1 - \frac{\frac{v^2 k_x^2}{2\omega^2} - \frac{v^4 k_x^4}{8\omega^4}}{1 - \frac{v^2 k_x^2}{2\omega^2}} \right) \\ \text{4th - order : } k_z = \frac{\omega}{v} \left(1 - \frac{\frac{v^2 k_x^2}{2\omega^2} - \frac{v^4 k_x^4}{4\omega^4}}{1 - \frac{3v^2 k_x^2}{4\omega^2} + \frac{v^4 k_x^4}{16\omega^4}} \right) \end{array} \right. \quad (19)$$

The corresponding time domain equations are:

$$\left\{ \begin{array}{l} \text{1st - order : } \frac{\partial^2 p}{\partial t \partial z} + \frac{v}{2} \frac{\partial^2 p}{\partial x^2} - \frac{1}{v} \frac{\partial^2 p}{\partial t^2} = 0 \\ \text{2nd - order : } \frac{\partial^3 p}{\partial t^2 \partial z} - \frac{v^2}{4} \frac{\partial^3 p}{\partial x^2 \partial z} - \frac{1}{v} \frac{\partial^3 p}{\partial t^3} + \frac{3v}{4} \frac{\partial^3 p}{\partial x^2 \partial t} = 0 \\ \text{3rd - order : } \frac{\partial^4 p}{\partial t^3 \partial z} - \frac{v^2}{2} \frac{\partial^4 p}{\partial x^2 \partial t \partial z} - \frac{1}{v} \frac{\partial^4 p}{\partial t^4} + v \frac{\partial^4 p}{\partial x^2 \partial t^2} - \frac{v^3}{8} \frac{\partial^4 p}{\partial x^4} = 0 \\ \text{4th - order : } \frac{\partial^5 p}{\partial t^4 \partial z} - \frac{3v^2}{4} \frac{\partial^5 p}{\partial x^2 \partial t^2 \partial z} - \frac{v^4}{16} \frac{\partial^5 p}{\partial x^4 \partial z} + \frac{1}{v} \frac{\partial^5 p}{\partial t^5} + \frac{5v}{4} \frac{\partial^5 p}{\partial x^2 \partial t^3} + \frac{5v^3}{16} \frac{\partial^5 p}{\partial t \partial x^4} = 0 \end{array} \right. \quad (20)$$

2.3 Absorbing boundary condition (ABC)

2.3.1 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 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 7), Figure 8 displays 10 shots data volume obtained the Clayton-Enquist boundary condition.¹

2.3.2 Sponge ABC

The sponge ABC was proposed by Cerjan et al. (1985). The principle is very simple: attenuating the reflections exponentially in the extended artificial boundary (Figure 9) area by multiplying a factor less $d(u)$ than 1. Commonly, we use the factor

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

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.

¹The forward modeling part using CUDA programming can also be found in FWI codes, /RSFSR-C/user/pyang/cuda_fw_kernels.cu.

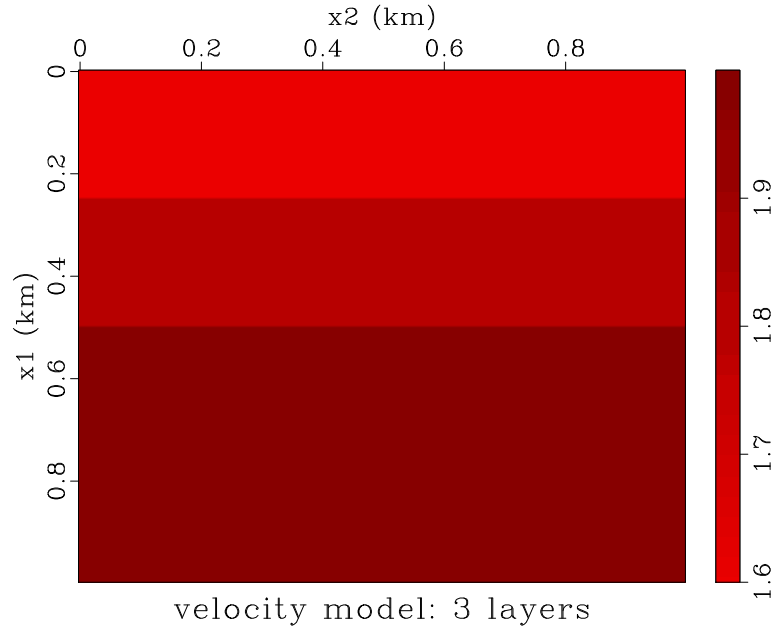


Figure 7: Velocity model: 3 layers

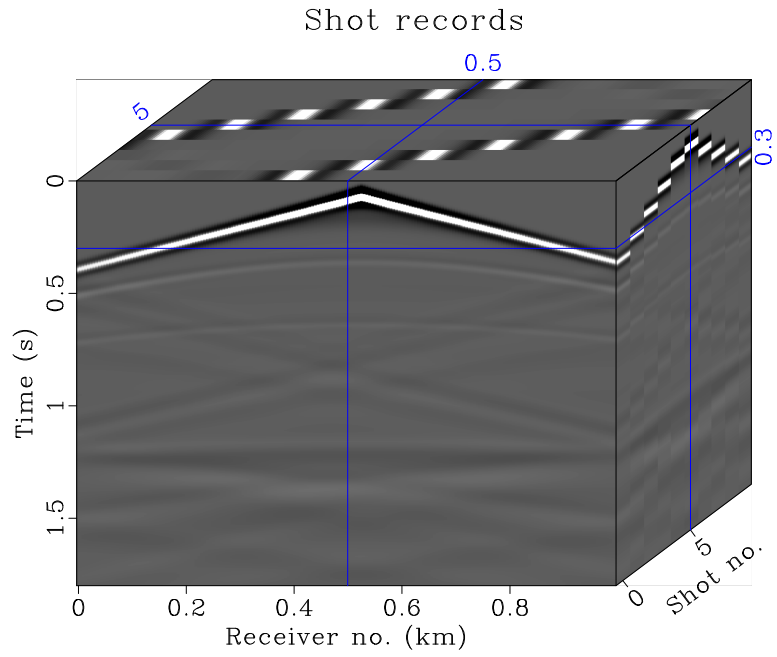


Figure 8: 10 shots data volume obtained using the Clayton-Enquist boundary condition.

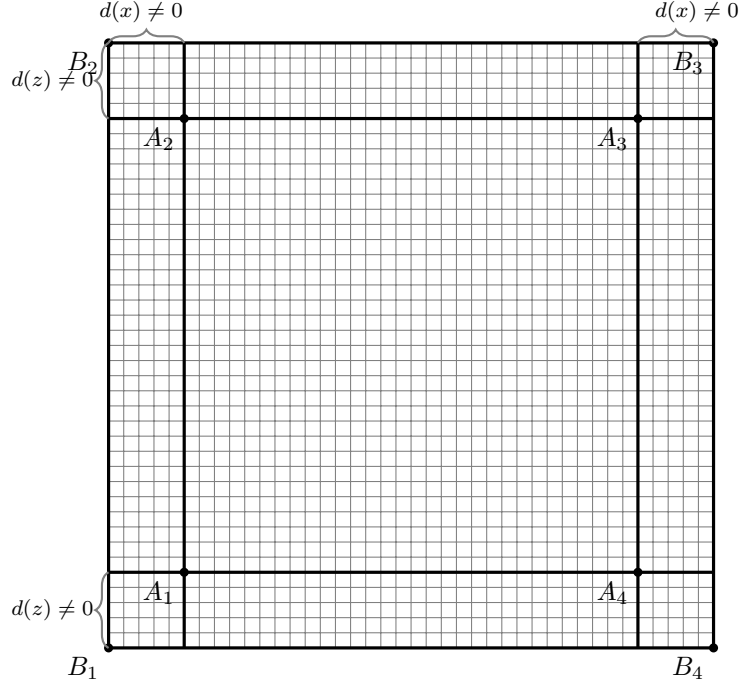


Figure 9: A schematic diagram of extended artificial boundary area. $A_1A_2A_3A_4$ is the original model zone, which is extended to be $B_1B_2B_3B_4$ with artificial boundary. In the extended boundary area, the attenuation coefficient $d(u) \neq 0$; In the model zone $A_1A_2A_3A_4$, $d(u) = 0$, $u = x, z$.

2.4 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).

2.4.1 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 . Then the acoustic wave equation becomes

$$\begin{cases} p = p_x + p_z \\ \frac{\partial p_x}{\partial t} = v^2 \frac{\partial v_x}{\partial x} \\ \frac{\partial p_z}{\partial t} = v^2 \frac{\partial v_z}{\partial z} \\ \frac{\partial v_x}{\partial t} = \frac{\partial p}{\partial x} \\ \frac{\partial v_z}{\partial t} = \frac{\partial p}{\partial z} \end{cases} \quad (22)$$

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

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

where $d(x)$ and $d(z)$ are the ABC coefficients designed to attenuate the reflection in the boundary zone, see Figure 9. There exists many forms of ABC coefficients function. In the absorbing layers, we use the following

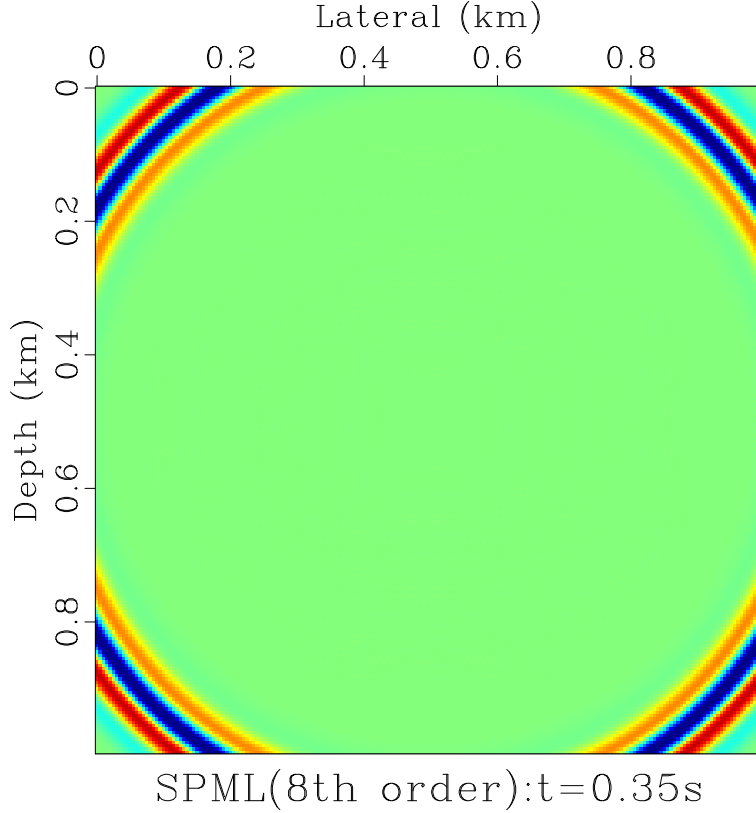


Figure 10: Wavefield snap of SPML with 8th order finite difference

model for the damping parameter $d(x)$ (Collino and Tsogka, 2001):

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

where L indicates the PML thickness; 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 extraction using Poynting vector, see Section 3.3.

A numerical example of SPML using 8th order staggered finite difference scheme is given in Figure 10.

2.4.2 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 Maxwell's 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 combine two convolution terms

into the above equations:

$$\begin{cases} \frac{\partial^2 p}{\partial t^2} = v^2 (Px + Pz) \\ Px = \frac{\partial Ax}{\partial x} + \Psi_x \\ Pz = \frac{\partial Az}{\partial z} + \Psi_z \\ Ax = \frac{\partial p}{\partial x} + \Phi_x \\ Az = \frac{\partial p}{\partial z} + \Phi_z \end{cases} \quad (25)$$

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:

$$\begin{cases} \Psi_x^n = b_x \Psi_x^{n-1} + (b_x - 1) \partial_x^{n-1/2} Ax \\ \Psi_z^n = b_z \Psi_z^{n-1} + (b_z - 1) \partial_z^{n-1/2} Az \\ \Phi_x^n = b_x \Phi_x^{n-1} + (b_x - 1) \partial_x^{n-1/2} Px \\ \Phi_z^n = b_z \Phi_z^{n-1} + (b_z - 1) \partial_z^{n-1/2} Pz \end{cases} \quad (26)$$

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).

2.4.3 Nonsplit PML (NPML) for elastics

The nonsplit PML for elastic implementation is

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

where the auxiliary variables are governed via the following relation

$$\begin{cases} \frac{\partial \Omega_{xx}}{\partial t} + d(x) \Omega_{xx} = d(x) \frac{\partial \tau_{xx}}{\partial x}, \frac{\partial \Omega_{xz}}{\partial t} + d(z) \Omega_{xz} = d(z) \frac{\partial \tau_{xz}}{\partial z} \\ \frac{\partial \Omega_{zx}}{\partial t} + d(x) \Omega_{zx} = d(x) \frac{\partial \tau_{xz}}{\partial x}, \frac{\partial \Omega_{zz}}{\partial t} + d(z) \Omega_{zz} = d(z) \frac{\partial \tau_{zz}}{\partial z} \\ \frac{\partial \Psi_{xx}}{\partial t} + d(x) \Psi_{xx} = d(x) \frac{\partial v_x}{\partial x}, \frac{\partial \Psi_{xz}}{\partial t} + d(z) \Psi_{xz} = d(z) \frac{\partial v_x}{\partial z} \\ \frac{\partial \Psi_{zx}}{\partial t} + d(x) \Psi_{zx} = d(x) \frac{\partial v_z}{\partial x}, \frac{\partial \Psi_{zz}}{\partial t} + d(z) \Psi_{zz} = d(z) \frac{\partial v_z}{\partial z} \end{cases} \quad (28)$$

2.5 Discretization

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

$$\begin{cases} f(x+h) = f(x) + \frac{\partial f(x)}{\partial x} h + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} h^2 + \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3} h^3 + \dots \\ f(x-h) = f(x) - \frac{\partial f(x)}{\partial x} h + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} h^2 - \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3} h^3 + \dots \end{cases} \quad (29)$$

It leads to

$$\begin{cases} \frac{f(x+h) + f(x-h)}{2} = f(x) + \frac{1}{2!} \frac{\partial^2 f(x)}{\partial x^2} h^2 + \frac{1}{4!} \frac{\partial^4 f(x)}{\partial x^4} h^4 + \dots \\ \frac{f(x+h) - f(x-h)}{2} = \frac{\partial f(x)}{\partial x} h + \frac{1}{3!} \frac{\partial^3 f(x)}{\partial x^3} h^3 + \frac{1}{5!} \frac{\partial^5 f(x)}{\partial x^5} h^5 + \dots \end{cases} \quad (30)$$

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

$$\begin{cases} \frac{\partial f(x)}{\partial x} = \frac{f(x + \Delta x/2) - f(x - \Delta x/2)}{\Delta x} + O(\Delta x^2) \\ f(x) = \frac{f(x + \Delta x/2) + f(x - \Delta x/2)}{2} + O(\Delta x^2) \end{cases} \quad (31)$$

2.5.1 Higher-order approximation of staggered-grid finite difference

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

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

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

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

Thus, taking first N terms means

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

In matrix form,

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

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

$$\mathbf{V} = \begin{bmatrix} 1 & 1 & \dots & 1 \\ x_1 & x_2 & \dots & x_N \\ \vdots & \vdots & \ddots & \vdots \\ x_1^{N-1} & x_2^{N-1} & \dots & x_N^{N-1} \end{bmatrix} \quad (36)$$

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

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

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

```

1 function c=staggered_fdcoeff(NJ)
2 % Computing 2*N-order staggered-grid FD coefficients (NJ=2N)
3 % Example:
4 %     format long
5 %     NJ=10;
6 %     c=staggered_fdcoeff(NJ)
7
8 N=NJ/2;
9 x=zeros(N,1);
10 b=zeros(N,1);    b(1)=1;
11 c=b;
12 for k=1:N
13     x(k)=(2*k-1)^2;
14 end
15
16 for k=1:N-1
17     for i=N:-1:k+1
18         b(i)=b(i)-x(k)*b(i-1);
19     end
20 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

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

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, we need at least 10 points in one minimum wavelength spatially to avoid dispersion artefacts:

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

while in 4th-order finite difference scheme, it can be relaxed to around 5 points per minimum wavelength

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

2.5.2 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 $[ix\Delta x, iz\Delta z]$

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

That is to say,

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

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

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

Thus, we have

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

In summary,

$$\left\{ \begin{array}{l} v_x^{k+\frac{1}{2}}[ix + \frac{1}{2}, iz] = \frac{1 - 0.5\Delta t d[ix]}{1 + 0.5\Delta t d[ix]} v_x^{k-\frac{1}{2}}[ix + \frac{1}{2}, iz] + \\ \frac{1}{1 + 0.5\Delta t d[ix]} \frac{\Delta t}{\Delta x} (p^k[ix + 1, iz] - p^k[ix, iz]) \\ v_z^{k+\frac{1}{2}}[ix, iz + \frac{1}{2}] = \frac{1 - 0.5\Delta t d[iz]}{1 + 0.5\Delta t d[iz]} v_z^{k-\frac{1}{2}}[ix, iz + \frac{1}{2}] + \\ \frac{1}{1 + 0.5\Delta t d[iz]} \frac{\Delta t}{\Delta z} (p^k[ix, iz + 1] - p^k[ix, iz]) \\ p_x^{k+1}[ix, iz] = \frac{1 - 0.5\Delta t d[ix]}{1 + 0.5\Delta t d[ix]} p_x^k[ix, iz] + \\ \frac{1}{1 + 0.5\Delta t d[ix]} \frac{\Delta t v^2[ix, iz]}{\Delta x} (v_x^{k+\frac{1}{2}}[ix + \frac{1}{2}, iz] - v_x^{k+\frac{1}{2}}[ix - \frac{1}{2}, iz]) \\ p_z^{k+1}[ix, iz] = \frac{1 - 0.5\Delta t d[iz]}{1 + 0.5\Delta t d[iz]} p_z^k[ix, iz] + \\ \frac{1}{1 + 0.5\Delta t d[iz]} \frac{\Delta t v^2[ix, iz]}{\Delta z} (v_z^{k+\frac{1}{2}}[ix, iz + \frac{1}{2}] - v_z^{k+\frac{1}{2}}[ix, iz - \frac{1}{2}]) \\ p^{k+1}[ix, iz] = p_x^{k+1}[ix, iz] + p_z^{k+1}[ix, iz] \end{array} \right. \quad (45)$$

If we define:

$$b' = \frac{1 - 0.5\Delta t d}{1 + 0.5\Delta t d}, b = \exp(-\Delta t d) \quad (46)$$

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

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

Then, we have

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

2.5.3 Discretization of NPML

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

$$\frac{\partial f}{\partial t} + df = \gamma. \quad (49)$$

The analytic solution of this equation is

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

In discrete form,

$$\begin{aligned} f(k\Delta t) &= -\frac{1}{d}e^{-dk\Delta t} + \frac{1}{d}\gamma, \\ f((k+1)\Delta t) &= -\frac{1}{d}e^{-d(k+1)\Delta t} + \frac{1}{d}\gamma. \end{aligned} \quad (51)$$

Thus,

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

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

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

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:

$$\left\{ \begin{aligned} \Omega_{xx}^{k+1} &= b_x\Omega_{xx}^k + (1 - b_x)\frac{\partial \tau_{xx}^{k+1/2}}{\partial x}, \Omega_{xz}^{k+1} = b_z\Omega_{xz}^k + (1 - b_z)\frac{\partial \tau_{xz}^{k+1/2}}{\partial z} \\ \Omega_{zx}^{k+1} &= b_x\Omega_{zx}^k + (1 - b_x)\frac{\partial \tau_{zx}^{k+1/2}}{\partial x}, \Omega_{zz}^{k+1} = b_z\Omega_{zz}^k + (1 - b_z)\frac{\partial \tau_{zz}^{k+1/2}}{\partial z} \\ \Psi_{xx}^{k+1} &= b_x\Psi_{xx}^k + (1 - b_x)\frac{\partial v_x^{k+1/2}}{\partial x}, \Psi_{xz}^{k+1} = b_z\Psi_{xz}^k + (1 - b_z)\frac{\partial v_x^{k+1/2}}{\partial z} \\ \Psi_{zx}^{k+1} &= b_x\Psi_{zx}^k + (1 - b_x)\frac{\partial v_z^{k+1/2}}{\partial x}, \Psi_{zz}^{k+1} = b_z\Psi_{zz}^k + (1 - b_z)\frac{\partial v_z^{k+1/2}}{\partial z} \end{aligned} \right. \quad (54)$$

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 procedure 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).

3 Reverse time migration (RTM)

3.1 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).

3.2 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.

3.3 Imaging condition

The cross-correlation imaging condition can be expressed as

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

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

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

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

There exist a better way to carry out the illumination compensation as suggested in Guitton et al. (2007)

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

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

$$\mathbf{S} = \mathbf{v}p = \nabla p \frac{dp}{dt} = (v_x p, v_z p). \quad (58)$$

Here, we denote S_s and S_r as the source wavefield and receiver wavefield Poynting vector. It is important to mention that the technique of Poynting vector also provides a convenient tool for RTM angle domain common image gather (ADCIG) extraction, however, we do not provide the implementation for the time being. The angle between the incident wave and the reflected wave can be obtained:

$$\gamma = \arccos \frac{\mathbf{S}_s \cdot \mathbf{S}_r}{|\mathbf{S}_s| |\mathbf{S}_r|} \quad (59)$$

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

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

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

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

where

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

Costa et al. (2009) modified the weight as

$$W(\theta) = \cos^3(\theta/2). \quad (63)$$

3.4 Computation strategies and boundary saving

There are some 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

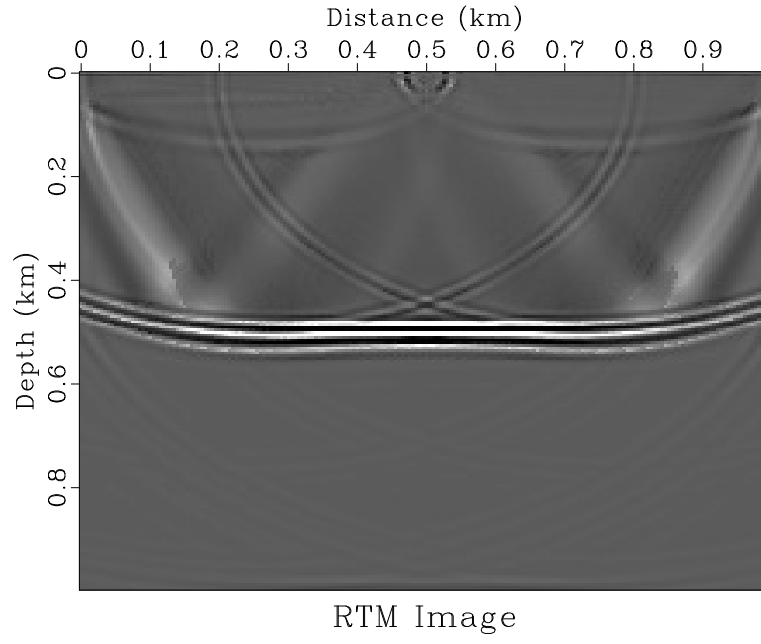


Figure 11: The RTM image by stacking all ADCIGs.

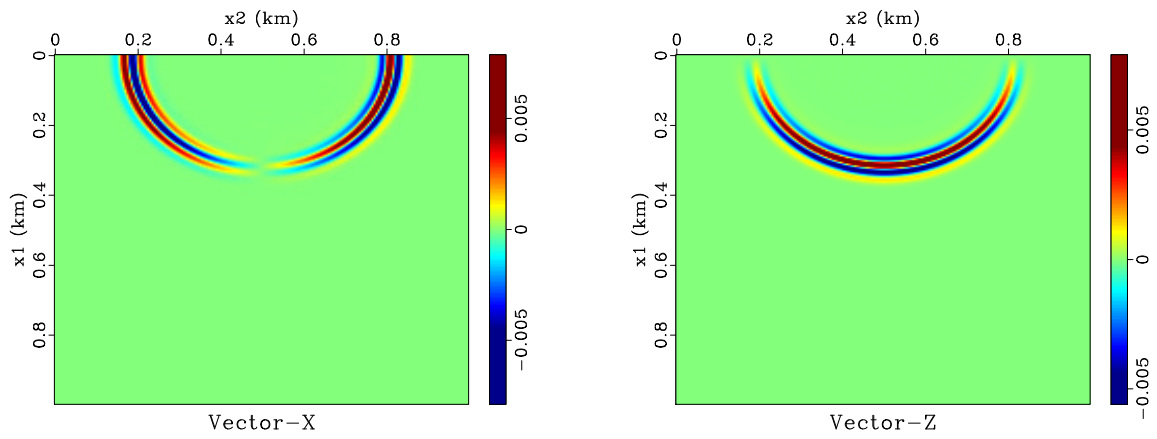


Figure 12: The x and z components indicate the directional Poynting vectors are working.

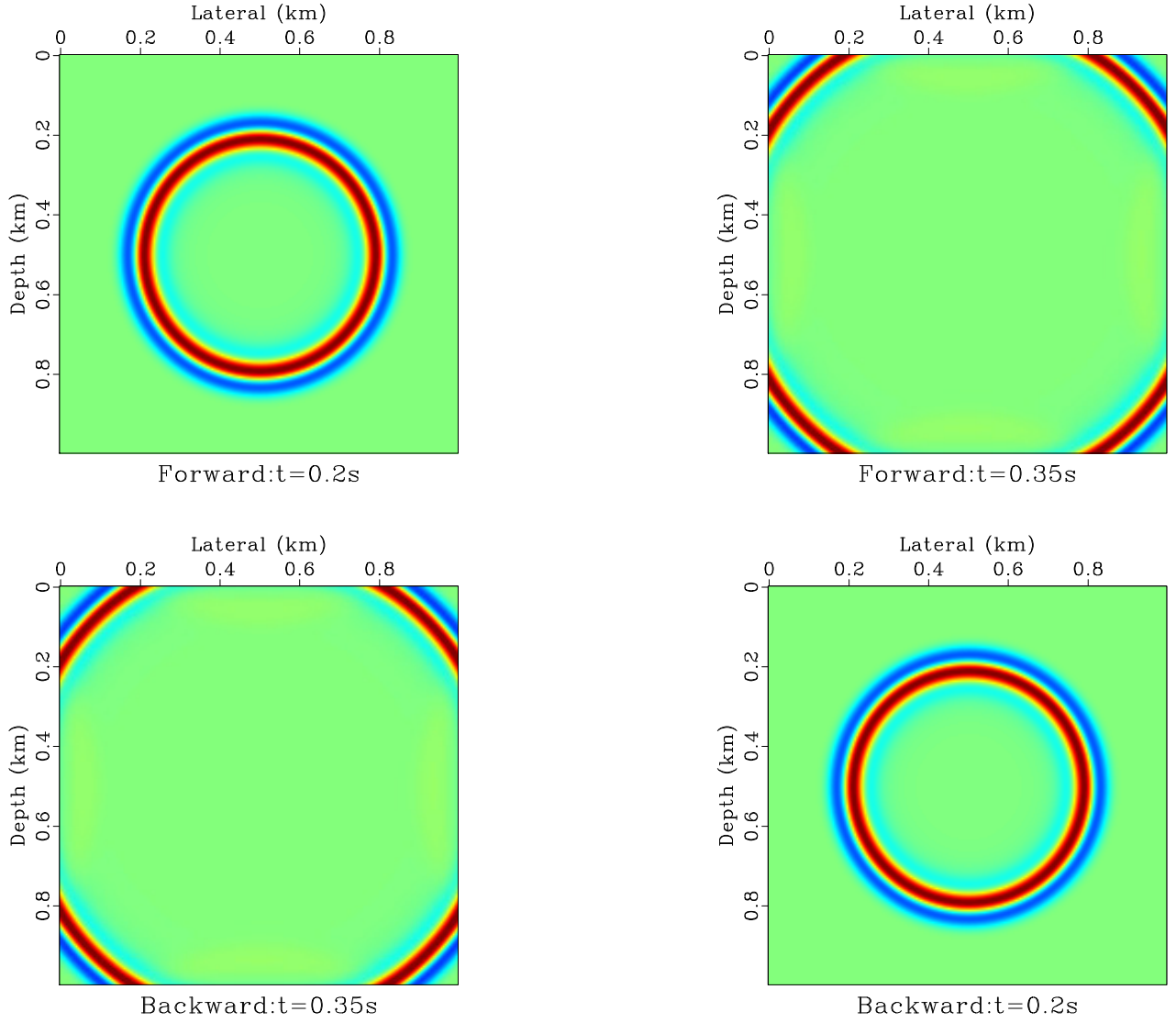


Figure 13: The forward modeled wavefield can be exactly reconstructed using effective boundary saving.

remodeling, using the saved wavefield snaps 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). Of course, 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) introduced the concept of effective boundary for regular and staggered grid finite difference. 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 side in the model for perfect reconstruction. The concept of effective boundary saving does not depend on C or GPU implementation. However, it is of special value for GPU implementation, 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 13. Effective boundary saving for staggered grid finite difference is adopted in /RSFSRC/user/pyang/Mgpurtm.cu.

3.5 Numerical examples

I show my GPU-based RTM result for two benchmark models: Marmousi model and the sigsbee model.²

²The GPU-based RTM codes is available in Madagascar /RSFSRC/user/pyang/Mgpurtm.cu. This code uses CPML boundary condition to obtain high quality imaging result.

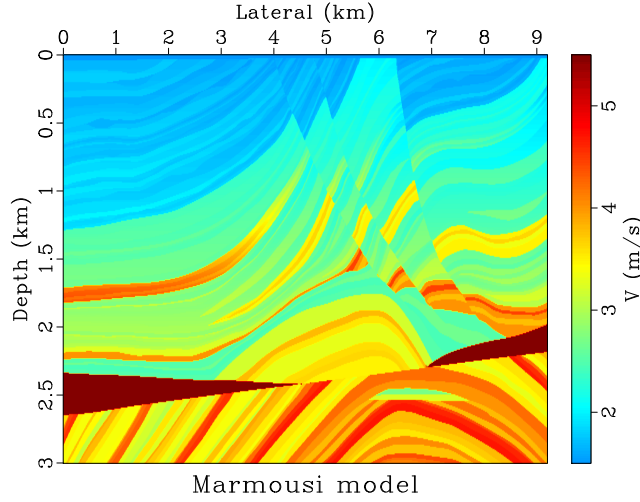


Figure 14: The Marmousi velocity model.

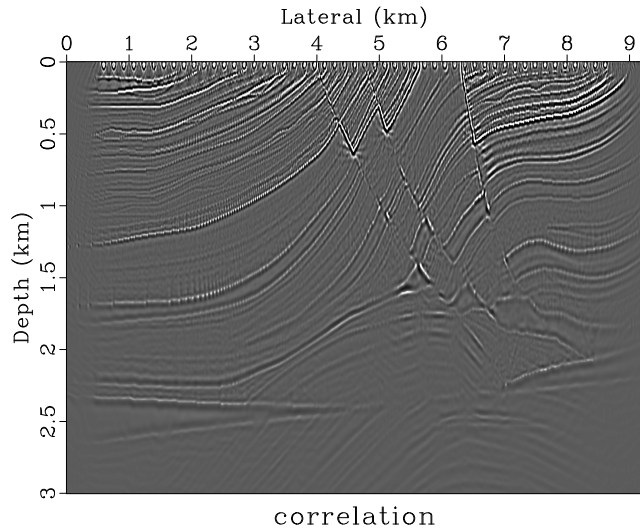


Figure 15: RTM result of Marmousi model using effective boundary saving scheme (staggered grid finite difference): Cross-correlation imaging condition.

The Marmousi model is shown in Figure 14. 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 15 and 16 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.

Sigsbee model is also shown in Figure 17. 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 18 and 19. 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.

4 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

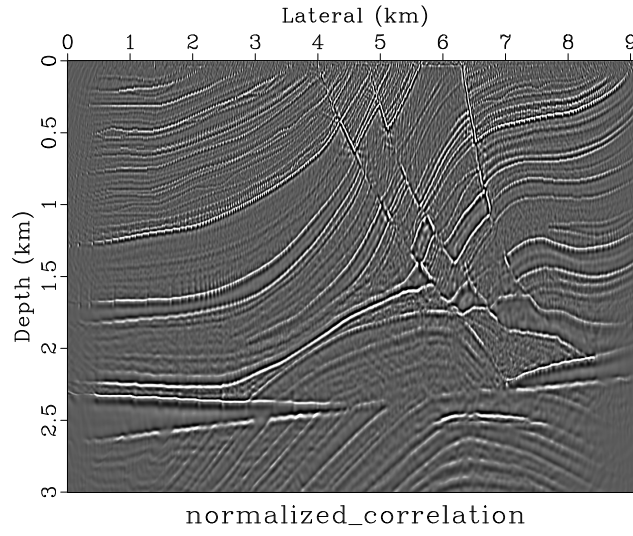


Figure 16: RTM result of Marmousi model using effective boundary saving scheme (staggered grid finite difference): Cross-correlation imaging condition.

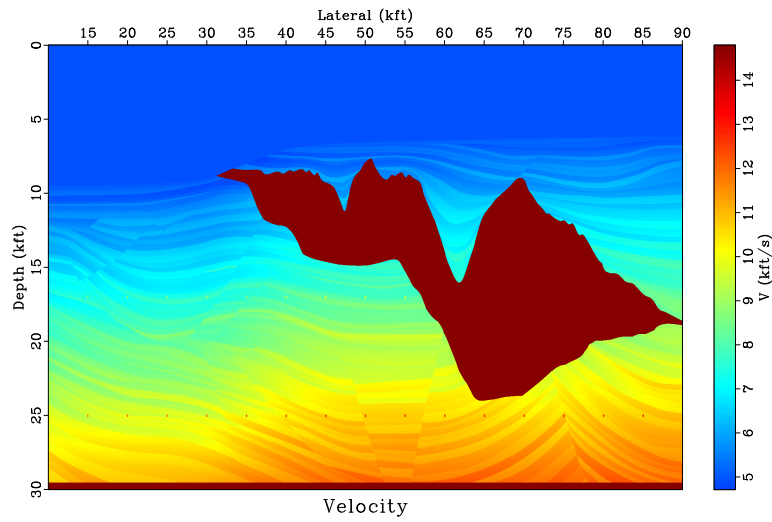


Figure 17: The Sigsbee velocity model.

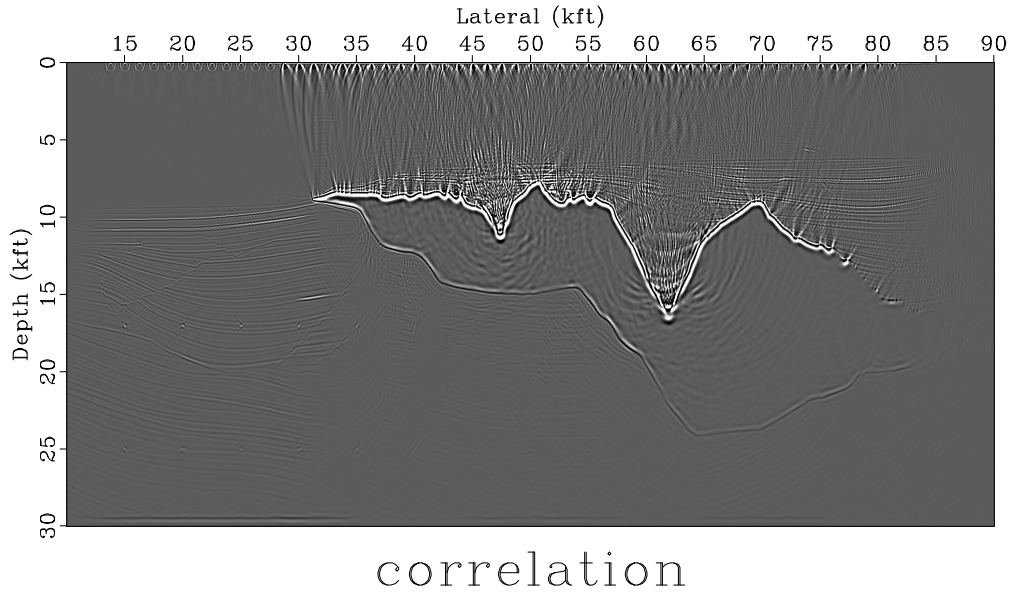


Figure 18: RTM result of Sigsbee model using effective boundary saving scheme (staggered grid finite difference): Cross-correlation imaging condition.

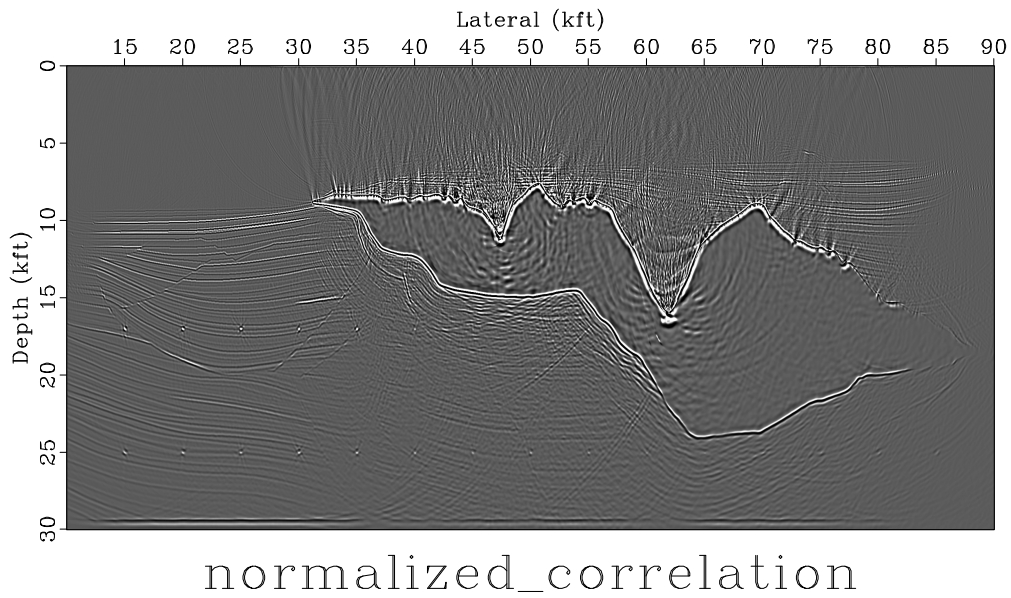


Figure 19: RTM result of Sigsbee model using effective boundary saving scheme (staggered grid finite difference): Normalized cross-correlation imaging condition.

tomography. 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

$$E(\mathbf{m}) = \frac{1}{2} \Delta \mathbf{p}^\dagger \Delta \mathbf{p} = \frac{1}{2} \Delta \mathbf{p}^T \Delta \mathbf{p}^* = \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int_0^{t_{\max}} dt |p_{cal}(\mathbf{x}_r, t; \mathbf{x}_s) - p_{obs}(\mathbf{x}_r, t; \mathbf{x}_s)|^2 \quad (64)$$

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 . The 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

$$E(\mathbf{m}_0 + \Delta \mathbf{m}) = E(\mathbf{m}_0) + \sum_{i=1}^M \frac{\partial E(\mathbf{m}_0)}{\partial m_i} \Delta m_i + \frac{1}{2} \sum_{i=1}^M \sum_{j=1}^M \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_i \partial m_j} \Delta m_i \Delta m_j + O(\|\Delta \mathbf{m}\|^3) \quad (65)$$

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

$$\frac{\partial E(\mathbf{m})}{\partial m_i} = \frac{\partial E(\mathbf{m}_0)}{\partial m_i} + \sum_{j=1}^M \frac{\partial^2 E(\mathbf{m}_0)}{\partial m_j \partial m_i} \Delta m_j, i = 1, 2, \dots, M. \quad (66)$$

Briefly speaking, it is

$$\frac{\partial E(\mathbf{m})}{\partial \mathbf{m}} = \frac{\partial E(\mathbf{m}_0)}{\partial \mathbf{m}} + \frac{\partial^2 E(\mathbf{m}_0)}{\partial \mathbf{m}^2} \Delta \mathbf{m} \quad (67)$$

Thus,

$$\Delta \mathbf{m} = - \left(\frac{\partial^2 E(\mathbf{m}_0)}{\partial \mathbf{m}^2} \right)^{-1} \frac{\partial E(\mathbf{m}_0)}{\partial \mathbf{m}} = -\mathbf{H}^{-1} \nabla E_{\mathbf{m}} \quad (68)$$

where

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

and

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

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

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

In terms of Eq. (64),

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

That is to say,

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

where 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} :

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

In matrix form

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

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

$$\Delta \mathbf{m} = -\mathbf{H}^{-1} \nabla E_{\mathbf{m}} = -\mathbf{H}_a^{-1} \text{Re}[\mathbf{J}^\dagger \Delta \mathbf{p}]. \quad (75)$$

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

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

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:

$$\Delta \mathbf{m} = -\mu^{-1} \nabla E_{\mathbf{m}} = -\alpha \nabla E_{\mathbf{m}} = -\alpha \text{Re} [\mathbf{J}^\dagger \Delta \mathbf{p}], \quad \alpha = \mu^{-1}. \quad (77)$$

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

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

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

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

4.2 Conjugate gradient (CG) implementation

The gradient-like method can be summarized as

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

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

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

There are many ways to compute β_k :

$$\left\{ \begin{aligned} \beta_k^{HS} &= \frac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle} \\ \beta_k^{FR} &= \frac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \nabla E(\mathbf{m}_{k-1}), \nabla E(\mathbf{m}_{k-1}) \rangle} \\ \beta_k^{PRP} &= \frac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle}{\langle \nabla E(\mathbf{m}_{k-1}), \nabla E(\mathbf{m}_{k-1}) \rangle} \\ \beta_k^{CD} &= -\frac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_{k-1}) \rangle} \\ \beta_k^{DY} &= \frac{\langle \nabla E(\mathbf{m}_k), \nabla E(\mathbf{m}_k) \rangle}{\langle \mathbf{d}_{k-1}, \nabla E(\mathbf{m}_k) - \nabla E(\mathbf{m}_{k-1}) \rangle} \end{aligned} \right. \quad (82)$$

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

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

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

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

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

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

- Approach 2: Recall that

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

Using the 1st-order approximation, we have

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

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

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

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} :

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

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

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

4.3 Fréchet derivative

Recall that the basic acoustic wave equation can be specified as

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

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

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

Thus the integral representation of the solution can be given by

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

where $*$ denotes the convolution operator.

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

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

Note that

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

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

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

4.4 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

$$\begin{aligned} \frac{\partial E(\mathbf{m})}{\partial m_i} &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int dt \text{Re} \left[\left(\frac{\partial p_{cal}}{\partial m_i} \right)^* \Delta p \right] (\Delta p = p_{cal} - p_{obs}) \\ &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int_0^{t_{\max}} dt \text{Re} \left[\left(\int_V d\mathbf{x} \Gamma(\mathbf{x}_r, t; \mathbf{x}, 0) * \frac{\partial^2 p(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} \frac{2}{v^3(\mathbf{x})} \right)^* \Delta p(\mathbf{x}_r, t; \mathbf{x}_s) \right] \\ &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int_0^{t_{\max}} dt \text{Re} \left[\left(\frac{\partial^2 p_{cal}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} \frac{2}{v^3(\mathbf{x})} \right)^* \left(\int_V d\mathbf{x} \Gamma(\mathbf{x}_r, -t; \mathbf{x}, 0) * \Delta p(\mathbf{x}_r, t; \mathbf{x}_s) \right) \right] \\ &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int_0^{t_{\max}} dt \text{Re} \left[\left(\frac{\partial^2 p_{cal}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} \frac{2}{v^3(\mathbf{x})} \right)^* \left(\int_V d\mathbf{x} \Gamma(\mathbf{x}_r, 0; \mathbf{x}, t) * \Delta p(\mathbf{x}_r, t; \mathbf{x}_s) \right) \right] \\ &= \sum_{r=1}^{ng} \sum_{s=1}^{ns} \int_0^{t_{\max}} dt \text{Re} \left[\left(\frac{\partial^2 p_{cal}(\mathbf{x}, t; \mathbf{x}_s)}{\partial t^2} \frac{2}{v^3(\mathbf{x})} \right)^* p_{res}(\mathbf{x}_r, t; \mathbf{x}_s) \right] \end{aligned} \quad (101)$$

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 reciprocal 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

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

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} .

An FWI using Marmousi model (size=240x737) has been shown in Figure 20. ³

³Our GPU-based FWI using Clayton-Enquist boundary condition can be found in Madagascar open software: /RSFSR-C/user/pyang/Mgpufwi.cu. This code uses Clayton-Enquist boundary condition due to its small saving burden on device, compared with PML absorbing boundary condition.

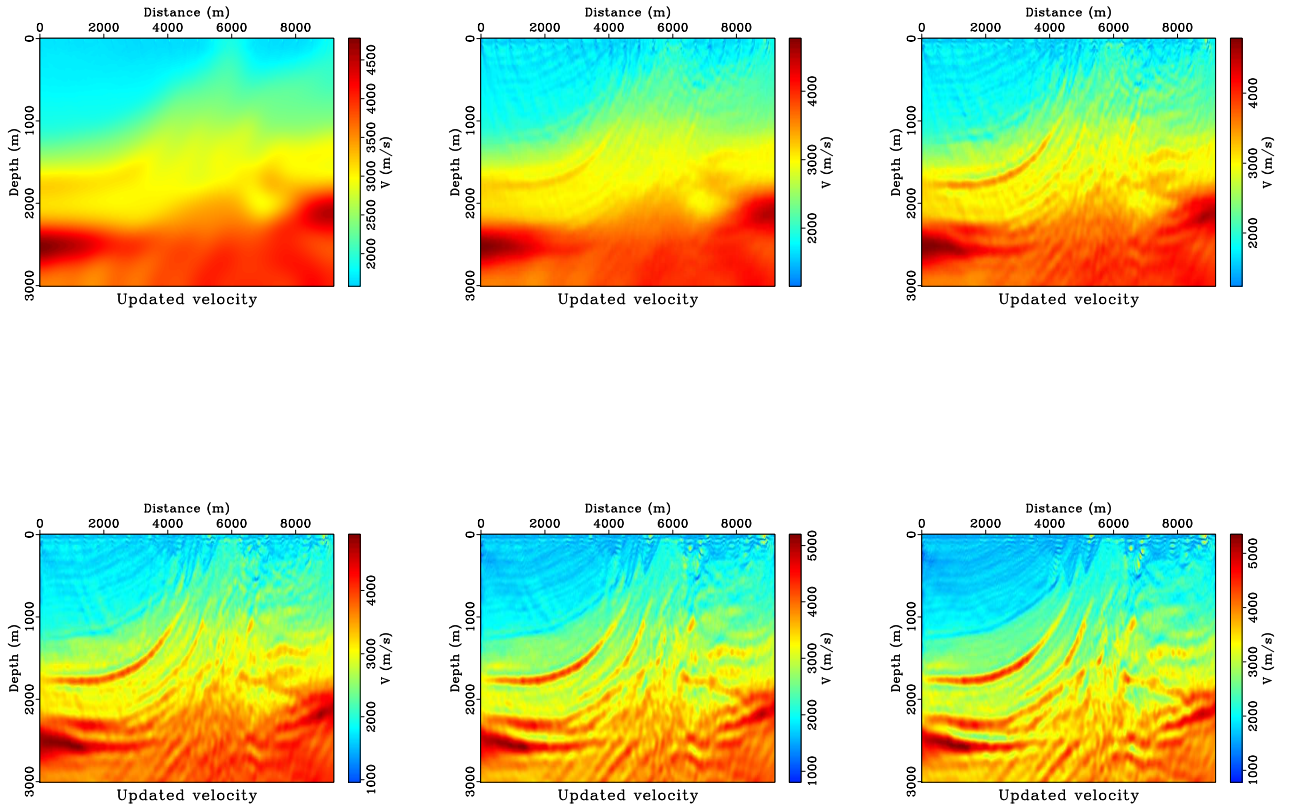


Figure 20: FWI for Mamrousi model: The updated velocity model at the iteration 1, 20, 50, 100, 180 and 300.

References

- Baysal, E., Kosloff, D. D., and Sherwood, J. W. (1983). Reverse time migration. *Geophysics*, 48(11):1514–1524.
- Berenger, J.-P. (1994). A perfectly matched layer for the absorption of electromagnetic waves. *Journal of computational physics*, 114(2):185–200.
- Biondi, B. (2006). *3D seismic imaging*. Number 14. Society of Exploration Geophysicists.
- Bjorck, A. (1996). *Numerical methods for least squares problems*. Number 51. Society for Industrial and Applied Mathematics.
- Cerjan, C., Kosloff, D., Kosloff, R., and Reshef, M. (1985). A nonreflecting boundary condition for discrete acoustic and elastic wave equations. *Geophysics*, 50(4):705–708.
- Claerbout, J. F. (1971). Toward a unified theory of reflector mapping. *Geophysics*, 36(3):467–481.
- Clapp, R. G. (2009). Reverse time migration with random boundaries. In *79th Annual International Meeting, SEG Expanded Abstracts*, volume 28, pages 2809–2813.
- Clapp, R. G., Fu, H., and Lindtjorn, O. (2010). Selecting the right hardware for reverse time migration. *The Leading Edge*, 29(1):48–58.
- Collino, F. and Tsogka, C. (2001). Application of the perfectly matched absorbing layer model to the linear elastodynamic problem in anisotropic heterogeneous media. *Geophysics*, 66(1):294–307.
- Dussaud, E., Symes, W. W., Williamson, P., Lemaistre, L., Singer, P., Denel, B., and Cherrett, A. (2008). Computational strategies for reverse-time migration. *SEG Annual meeting*.
- Engquist, B. and Majda, A. (1977). Absorbing boundary conditions for numerical simulation of waves. *Proceedings of the National Academy of Sciences*, 74(5):1765–1766.
- Guitten, A., Kaelin, B., and Biondi, B. (2006). Least-squares attenuation of reverse-time-migration artifacts. *Geophysics*, 72(1):S19–S23.

- Guittou, A., Valenciano, A., Bevc, D., and Claerbout, J. (2007). Smoothing imaging condition for shot-profile migration. *Geophysics*, 72(3):S149–S154.
- Komatitsch, D. and Martin, R. (2007). An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. *Geophysics*, 72(5):SM155–SM167.
- Liu, G., Liu, Y., Ren, L., and Meng, X. (2013a). 3d seismic reverse time migration on gpgpu. *Computers & Geosciences*, 59(0):17 – 23.
- Liu, H., Ding, R., Liu, L., and Liu, H. (2013b). Wavefield reconstruction methods for reverse time migration. *Journal of Geophysics and Engineering*, 10(1):015004.
- McMechan, G. (1983). Migration by extrapolation of time-dependent boundary values. *Geophysical Prospecting*, 31(3):413–420.
- Pica, A., Diet, J., and Tarantola, A. (1990). Nonlinear inversion of seismic reflection data in a laterally invariant medium. *Geophysics*, 55(3):284–292.
- Pratt, G., Shin, C., et al. (1998). Gauss-newton and full newton methods in frequency-space seismic waveform inversion. *Geophysical Journal International*, 133(2):341–362.
- Roden, J. A. and Gedney, S. D. (2000). Convolutional pml (cpml): An efficient fdtd implementation of the cfs-pml for arbitrary media. *Microwave and optical technology letters*, 27(5):334–338.
- Symes, W. W. (2007). Reverse time migration with optimal checkpointing. *Geophysics*, 72(5):SM213–SM221.
- Tarantola, A. (1984). Inversion of seismic reflection data in the acoustic approximation. *Geophysics*, 49(8):1259–1266.
- Tarantola, A. (1986). A strategy for nonlinear elastic inversion of seismic reflection data. *Geophysics*, 51(10):1893–1903.
- Virieux, J. and Operto, S. (2009). An overview of full-waveform inversion in exploration geophysics. *Geophysics*, 74(6):WCC1–WCC26.
- Yang, P., Gao, J., and Wang, B. (2014). {RTM} using effective boundary saving: A staggered grid {GPU} implementation. *Computers & Geosciences*.
- Yoon, K. and Marfurt, K. J. (2006). Reverse-time migration using the poynting vector. *Exploration Geophysics*, 37(1):102–107.
- Yoon, K., Shin, C., Suh, S., Lines, L. R., and Hong, S. (2003). 3d reverse-time migration using the acoustic wave equation: An experience with the seg/eage data set. *The Leading Edge*, 22(1):38–41.