Wavefield Extrapolation in Pseudo-depth Domain

Thesis by
Xuxin Ma

Submitted in Partial Fulfillment of the Requirements for the
Degree of
Masters of Science

King Abdullah University of Science and Technology

Physical Sciences and Engineering

Earth Sciences and Engineering

Thuwal, Makkah Province, Kingdom of Saudi Arabia

December, 2011

The thesis of Xuxin Ma is approved by the examination committee.

Committee Chairperson: Tariq Alkhalifah

Committee Member: David Ketcheson

Committee Member: Martin Mai

Copyright ©2011

Xuxin Ma

All Rights Reserved

ABSTRACT

Wavefield extrapolation in pseudo-depth domain

Xuxin Ma

Wave-equation based seismic migration and inversion tools are widely used by the energy industry to explore hydrocarbon and mineral resources. By design, most of these techniques simulate wave propagation in a space domain with the vertical axis being depth measured from the surface. Vertical depth is popular because it is a straightforward mapping of the subsurface space. It is, however, not computationally cost-effective because the wavelength changes with local elastic wave velocity, which in general increases with depth in the Earth. As a result, the sampling per wavelength also increases with depth. To avoid spatial aliasing in deep fast media, the seismic wave is oversampled in shallow slow media and therefore increase the total computation cost. This issue is effectively tackled by using the *vertical time* axis instead of vertical depth. This is because in a vertical time representation, the “wavelength” is essentially time period for vertical rays.

This thesis extends the vertical time axis to the *pseudo-depth* axis, which features distance unit while preserving the properties of the vertical time representation. To explore the potentials of doing wave-equation based imaging in the pseudo-depth domain, a Partial Differential Equation (PDE) is derived to describe acoustic wave in this new domain. This new PDE is inherently anisotropic because the use of a constant vertical velocity to convert between depth and vertical time. Such anisotropy results in lower reflection coefficients compared with conventional space domain modeling results. This feature is helpful to suppress the low wavenumber artifacts in reverse-time migration images, which are caused by the widely used cross-correlation

imaging condition. This thesis illustrates modeling acoustic waves in both conventional space domain and pseudo-depth domain. The numerical tool used to model acoustic waves is built based on the lowrank approximation of Fourier integral operators. To investigate the potential of seismic imaging in the pseudo-depth domain, examples of zero-offset migration are implemented in pseudo-depth domain and compared with conventional space domain imaging results.

ACKNOWLEDGMENTS

I would like to express my sincerest gratitude to my supervisor, Dr Tariq Alkhalifah, who has supported me with guidance and patience throughout my study. I'm thankful to Dr David Ketcheson and Dr Martin Mai and for serving in my committee. I appreciate their times reading the revisions of this thesis. Discussions with Dr Lexing Ying about the low-rank decomposition was helpful to implement this method. I appreciate Dr Christos Saragiotis for proofreading a draft of this thesis.

I'm deeply indebted to my parents. I would not have finished this thesis without their support.

TABLE OF CONTENTS

Abstract	5
Acknowledgement	6
List of Abbreviations	7
List of Symbols	8
List of Illustrations	9
List of Tables	10
I Introduction	11
I.1 Overview of Exploration Seismology	11
I.2 Structure of the Thesis	21
II Acoustic Wave in Pseudo-depth Domain	23
II.1 Vertical Time and Pseudo-depth	23
II.2 Domain Transformation	28
II.3 Extrapolation in the Pseudo-depth Domain	33
III Numerical Methods	43
III.1 Numerical Methods for Seismic Modeling	43
III.2 Spectral Extrapolation	45
III.2.1 Theory	45
III.2.2 Comparison with Finite Difference Method	52
III.3 Low-rank Decomposition	54
IV Seismic Imaging Examples	59
IV.1 Overview of Wave-equation Depth Migration	59
IV.2 Zero-offset Modeling and Migration Example	62
V Discussions	68
V.1 Conclusions	68
V.2 Future Directions	69
A Source Codes	72
B Conference	80
Bibliography	81

LIST OF ABBREVIATIONS

AGC	Automatic Gain Control
AVO	Amplitude Variation with Offset
DMO	Dip Moveout
EM	Electromagnetic
FIO	Fourier Integral Operator
NMO	Normal Moveout
PDE	Partial Differential Equation
PreSDM	Pre-stack Depth Migration
PreSTM	Pre-stack Time Migration
RTM	Reverse-time Migration
ZOM	Zero-offset Migration

LIST OF SYMBOLS

λ	wavelength
σ	lateral gradient of τ
τ	vertical time
v	P-wave velocity
v_0	scaling velocity
v_V	vertical velocity
z	vertical depth
z^*	pseudo-depth

LIST OF ILLUSTRATIONS

I.1	Seismic survey	15
I.2	Seismic dataset	16
I.3	Seismic processing workflow	17
I.4	Multiple reflections	18
I.5	Migration of a seismic section	20
I.6	Migration of a seismic section (continued)	21
II.1	Physical ray and vertical ray in a $v(x)$ medium	24
II.2	Physical ray and vertical ray in a $v(z)$ medium	25
II.3	Sampling of vertical depth axis	26
II.4	Sampling of vertical time axis	27
II.5	Pseudo-depth transformation	30
II.6	Pseudo-depth transformation (continued)	31
II.7	Transformation using different scaling velocities.	32
II.8	Wavefield transformation	33
II.9	Acoustic modeling in the two domains	36
II.10	Acoustic modeling in the two domains (continued)	37
II.11	Dispersion curves of pseudo-depth wave equation	38
II.12	Flat reflector modeling in the two domains	41
II.13	Dipping reflector modeling in the two domains	42
III.1	Numerical dispersion of finite difference and spectral method	53
IV.1	Prestack imaging condition	61
IV.2	Prestack imaging condition (continued)	62
IV.3	Zero-offset modeling	63
IV.4	Zero-offset modeling (continued)	64
IV.5	Zero-offset migrations in two domains	66
IV.6	Zero-offset migrations in two domains (continued)	67

LIST OF TABLES

I.1	Geophysical surveying methods	12
-----	---	----

CHAPTER I

Introduction

I.1 Overview of Exploration Seismology

To explore hydrocarbon and mineral deposits in the Earth, or to produce them efficiently once discovered, it is essential to know the subsurface geology as accurate as possible. For example, drilling a wellbore as deep as thousands of feet beneath the ground is a risky business because of the huge cost involved. To prevent drilling into reservoirs with low hydrocarbon content, it is important to estimate accurately where and how much hydrocarbon is actually present in the subsurfaces before starting the drilling. During production, it is also helpful to track the flow of oil and water in the reservoir rocks in order to extract hydrocarbon with minimum amount of water injection, so as to minimize the economic and environmental cost.

The only way to understand the subsurface geology exactly is to measure rock samples collected from boreholes. Since drilling a borehole is in general very expensive and may cause negative environmental impact, most geophysical surveys are carried out on the Earth's surface. Surface surveys are not direct measurement of subsurface rock properties. One can at best *estimate* the subsurface structure that would most likely result in the surface observations. Mathematically, subsurface mapping belongs to the category of solving *inverse problems*. Solving an inverse problem is essentially the estimation of the coefficients of PDE using observed solutions at certain space-time locations, whereas the *direct problem* is simply solving that PDE. Most inverse problems are more difficult to solve than direct problems because they are inherently ill-posed and nonlinear in most applications (Tarantola, 2005). The ill-

posedness with subsurface mapping problems originates from the non-uniqueness of their solutions. The solution ambiguity may be partially decreased through the work of *geological interpretation*, a process that rules out unlikely solutions by geological constraints (Lines et al., 1988), for example a commonly adopt assumption is that subsurface velocity increases with depth.

Based on different physical principles, geophysicists have developed several methods to map the subsurface structures. Each method measures certain physical properties of the subsurface rocks. Table I.1 summarizes these methods and their related rock properties (Telford et al., 1990).

Geophysical Method	Observed data	Rock properties
Gravity	Strength of gravitational field	Mass density
Magnetic	Strength of magnetic field	Magnetic susceptibility
Electrical	Resistance of the subsurface rock	Electrical conductivity
Electromagnetic	Response to electromagnetic radiation	Electrical conductivity and magnetic permeability
Seismic	Traveltimes of reflected and refracted seismic waves	Elastic moduli

Table I.1: Geophysical surveying methods, the data observed by each method and the rock properties that can be inferred from each method.

Gravity prospecting measures the variations in gravitational field of the Earth. The anomalies in gravitational field suggests greater or lower local mass density than its surrounding rocks. A wide range of geological settings could cause gravitational field anomalies. For example, small scale negative anomalies may be associated with salt domes. Magnetism and gravity have much in common, but magnetic field is more complex in general, mainly due to its dipolar nature where as gravity field is purely monopolar (Telford et al., 1990). Both gravitational and magnetic methods are *passive* prospecting methods, i.e. a naturally occurring field is observed without

applying external source. In fact both gravitational and magnetic forces are *conservative*, by which it means the energy difference between two locations is independent of the path along which the force is exerted. Mathematically, conservative forces can be represented as the gradient of some scalar quantity, called *potential field*. The term *potential method* refers to exploration tools using either gravity or magnetism principles (Blakely, 1995). Potential methods, although limited by resolution and imaging depth, are still often used as secondary tools in exploration because they are much cheaper and faster to implement than the more accurate seismic or electromagnetic surveys (Telford et al., 1990).

Electrical prospecting involves the detection of electric current flow in rocks. The electrical conductivity of subsurface rocks varies enormously from different types of rocks and also with the amount of water or hydrocarbon saturation. The major advantage of electrical method is its ability to differentiate water and hydrocarbon. The two fluids are difficult to distinguish by seismic methods because of their similar elastic moduli. Water, once saturated with salt, is very conductive to electric current, while hydrocarbon is a poor conductor of electric current (Keller and Frischknecht, 1966). Because electric field diffuses quickly in high resistive media such as subsurface rocks, the depth coverage of electrical prospecting methods are often limited. As a result, electrical methods are mostly used in borehole measurements.

Electromagnetic (EM) prospecting records the reflection of EM waves by subsurface rock interfaces. An ore deposit usually has distinctive electrical conductivity and magnetic permeability than its surrounding rocks. The drastic change of these properties form reflectors for EM waves. Unlike its propagation in air which is nearly free from loss of energy, EM waves are *diffusive* in subsurface rocks. Seismic waves in the far-field, in contrast, are non-diffusive waves. Only in the near-field, energy is lost due to permanent deformation of the media. As a result, the depth coverage of EM methods are much less than seismic methods. EM methods are most com-

monly used in ore exploration because they are good reflectors of EM waves and they are also buried shallow in the Earth. Recently, EM methods are gaining increasing attentions among hydrocarbon explorers with the increasing number of subsalt discoveries. The salt and basalt rocks, because of their higher acoustic impedance than shale and sediment rocks, are almost opaque to seismic waves. Seismic methods are therefore difficult to image structures below salt layers. EM waves on the other hand, can penetrate these structures without significant reflection and therefore provides an opportunity to image subsalt structures (Sinha et al., 1990; Young and Cox, 1981).

Although gravity, magnetic, electrical and electromagnetic observations have been made successfully, seismic methods, especially reflection seismology, remains by far the most successful geophysical tool for subsurface imaging. Exploration seismology is an offspring of *seismology*, the study of earthquake motion and the interior of the Earth. When a fracture occurs in the Earth, seismic waves are generated from the fracture surface and causes earthquakes when they propagate to the ground. Seismologists use *seismometers* deployed at various sites to record the ground motion during an earthquake (Lowrie, 2007). The recordings are called *seismograms*.

Exploration seismology adopts similar measurements with a major difference in that the energy source of seismic wave is controlled and artificially sent into the Earth. During a seismic survey, an energy-intensive source is discharged at the surface and generates low-frequency sound waves that travel through the Earth. These sound waves are reflected by subsurface interfaces where there is a change of elastic impedance. The energy source used in onshore seismic surveys are usually explosives or ground-shaking vehicles. A line of receivers, called *geophones* are placed on the Earth's surface to record the ground motion as the reflected sound waves reach the surface (Sheriff and Geldart, 1995). Geophones record the vertical component of particle velocities of the surface media. In offshore surveys, compressed air is used almost exclusively as the energy source (Sheriff and Geldart, 1995). Similar to onshore

surveys, a line of receivers called *hydrophones* are placed on the sea surface to record reflected sound waves. Figure I.1 illustrates an example of an offshore seismic survey. Hydrophones record the pressure change under water during a seismic experiment. Each line of hydrophones form a streamer. Several streamers are towed in parallel by the survey vessel. An airgun shoots high-energy compressed air into the water and generate sound waves. The waves propagate into the Earth and are reflected by subsurface reflectors. The pressure change caused by reflected sound waves is recorded by hydrophones deployed on the sea surface.

Figure I.1: Marine seismic survey. Sound waves are generated by discharging compressed air in the airgun. These waves propagate into the Earth and are reflected by subsurface rock interfaces. The reflected waves are recorded by a line of hydrophones fixed on a streamer. The streamer is towed behind the survey vessel. A buoyant object at the streamer tail keeps the streamer floating on sea surface. (Figure of Schlumberger)

An example seismic dataset is plotted in Figure I.2. The full dataset is a collection of seismograms. Each seismogram is observed from a single shot. The most important information delivered by a seismogram is the traveltimes of seismic events, the elapsed times for sound waves to travel from the source down to the reflectors and back to the receivers on the surface. Using the recorded traveltimes combined with certain assumption about the subsurface structure, geophysicists are able to calculate the depths and angles of subsurface reflectors. This is similar to submarines detecting the distance and shape of other vessels using sound wave echoes. The amplitude

Figure I.2: A marine seismic dataset. (a) The full dataset plotted as a cube with three axis of time, shot position and offset between the source and the receivers. The top surface of the cube is a section at $t = 0$ sec. The front surface is *near-offset* section with offset $h = 1$ m. The right surface is a single shot gather (seismogram) with shot position $s = 875$ m. (b) The same shot gather as the front surface of cube (a), plotted as wiggle traces.

information carried by a seismogram is less important than the traveltimes, because amplitude information is rarely accurate due to the presence of noise.

The term noise in seismic processing refers to any event on the seismic record from which useful information can not be obtained (Sheriff and Geldart, 1995). Seismic noise may be originated from a wide range of causes. For example, noise could be surface waves during land acquisition, nearby vehicles or vessels during acquisition, wave motion of water flow that moves hydrophones off their positions in the acquisition line. Although some noise can be effectively attenuated by stacking gathers or apply proper filtering, the amplitude of seismic data is in general less accurate than its phase information, or traveltimes. As a result, amplitude variations in the seismic data are ignored as a common practice, except in certain applications such as Amplitude Variation with Offset (AVO) in which amplitude is critical to infer the thickness, porosity and water content of a potential reservoir.

To build a subsurface reflectivity image from seismic dataset, a series of steps

are needed to control dataset quality, estimate velocity model and eventually map the reflection events into their correct subsurface locations. The complete seismic processing is composed of three sections. This is illustrated in Figure I.3.

Figure I.3: Basic seismic processing workflow. Seismic data is first collected during Acquisition. The dataset is then processed by pre-processing, velocity estimation and imaging steps to generate a subsurface image. Geological structures are then inferred from the reflectivity image through interpretation.

First, the *raw* seismic dataset acquired from seismic acquisition undergoes a series of *pre-processing* operations before it was migrated. In a homogeneous three-dimensional medium, the energy intensity of sound wave decays proportionally with the square of distance measured from the source. This is known as the *geometric spreading* of spherical waves. Therefore amplitudes of reflection events decay with depths of the reflectors. Automatic Gain Control (AGC) is an operation that removes the effect of geometric spreading by artificially increasing the amplitude of seismic traces with increasing time. Land seismic surveys require a special processing called *static correction*. This is because during land seismic surveys, the geophones are rarely placed on a planar surface. The surface topography effectively shifts the recorded traveltimes of seismic events. Static correction is a measure to restore the

“true” seismogram as if it was recorded on a flat surface. This is often done by applying a time-shift to each trace according to the surface elevation and P-wave velocity at near-surface (Yilmaz, 2001). As explained earlier, a seismic dataset is subject to various types of noise. Stacking gathers and bandpass filtering are common practices to suppress noise. In most cases they are effective to enhance the signal-to-noise ratio. Most seismic imaging algorithms are based on the assumption that the seismic dataset is composed of *primary*¹ reflections only, i.e. seismic events that undergo only one reflection from source to receiver. However, real seismic data records all seismic reflections, including primary reflections and multiple reflections. Figure I.4 is a sketch of the generation of multiples. A seismic dataset with significant multiples,

Figure I.4: Multiple reflections may be caused by sea surface (ghost multiples) due to the high impedance contrast between water and air, or by bouncing between layering of reflectors. (Figure of Schlumberger)

when processed for reflectivity imaging, will cause the multiples to appear as false reflectors in the migration image. This issue is especially important for marine seismic surveys, because the sea surface reflects seismic waves nearly without loss of energy. Multiple removal remains an ongoing research topic. Several methods have been quite successfully applied on field datasets, for example methods based on parabolic Radon transform (Hampson, 1986) and iteratively predicting surface multiples from

¹Seismic events whose energy has been reflected once before being recorded by receivers.

primaries (Berkhout and Verschuur, 1997). At last, a process called *deconvolution* is applied to the seismic dataset to improve visibility of subtle seismic events. During a seismic survey, the energy source sends a mixed-phase disturbance into the subsurface. For visualization purpose and especially to facilitate first-arrival picking during velocity estimation, seismic traces are most suitably presented by zero-phase wavelet, usually a Ricker wavelet. Deconvolution is an operation that attempts to restore the seismogram as if it was recorded with a zero-phase source wavelet.

To construct a subsurface reflectivity image, it is necessary to have a subsurface velocity model. *Velocity estimation* refers to the operation that generates a *smooth* subsurface velocity model. The velocity is smooth because it is a low resolution model that serves as background velocity for imaging process. In fact, if the velocity is known in great detail at every point, there will be no need to image subsurface reflectivity anymore. Traditionally, velocity estimation is done by moveout corrections including Normal Moveout (NMO) and Dip Moveout (DMO). Moveout corrections assume the Earth's subsurface is a stack of flat or dipping layers. The moveout velocity analysis seeks for an interval velocity that flattens the hyperbolic moveout observed on seismograms (Yilmaz, 2001). In complex geological structures where the interval assumption no longer holds, *traveltime tomography* is a more accurate approach to estimate subsurface velocity. Tomography is posed as an inverse problem: the objective is to find a smooth velocity model that minimizes certain cost function, which quantifies the difference between a synthetic traveltime and observed traveltime, while honoring several constraints for example from well-log data or geological knowledge (Lines et al., 1988).

Finally, using the properly pre-processed seismic dataset together with a velocity model, geophysicists are able to map subsurface reflectivity by *seismic migration*. Reflected in surface recordings, a flat reflector appears as a flat event. A dipping reflector, in contrast, appears as a dipping event with a steeper angle located at a

deeper position. For example, Figure I.6(a) is a synthetic seismic section of a syncline reflector illustrated in Figure I.5(b). The reflection events in Figure I.6(a) are apparently mispositioned from the syncline shape and form a “bow-tie” shape. Such mispositioning is due to the fact that recorded reflections are not always generated directly beneath the receiver. The term *migration* originally refers to the operations

Figure I.5: (a) A subsurface velocity model with two velocity layers. Velocities of each layer is 1500 m/sec and 3500 m/sec respectively. (b) Reflectivity of the same velocity model. The reflector has a simple synclinal shape. For visualization purpose, reflectivity image is often plotted as its convolution with a band-limited wavelet.

that relocates a seismic event to its true subsurface location. In the same example, the migration image in Figure I.6(b) clearly shows the mispositioned events are “migrated” to its correct location. The term migration later also refers to other aspects of seismic imaging and is eventually used interchangeably with the word *imaging*.

To prevent velocity estimation from converging to a false velocity model, velocity estimation and imaging are often run iteratively, or so called *migration velocity analysis*. The iteration starts with a simple homogeneous velocity model and gradually increase the number of velocity parameters after each iteration. Eventually, a high resolution velocity and reflectivity models are constructed. Section IV.1 gives a

Figure I.6: (a) Synthetic zero-offset seismic section of the subsurface model in Figure I.5(b). The events recorded by receiver at 150 m (marked by dots) are reflected from two subsurface locations (marked by dots in Figure I.5(b)). (b) The same seismic section after migration. The migration image agrees with original reflectivity model in Figure I.5(b). The artifacts on left and right sides of the image are due to reflection at boundaries.

detailed review of depth migration algorithms.

I.2 Structure of the Thesis

This thesis is divided into three parts. Chapter II introduces the concept of pseudo-depth and how to transform an arbitrary function in space into pseudo-depth domain. Based on this transformation, a PDE is derived to describe acoustic waves in the pseudo-depth domain. The wavefield extrapolated in pseudo-depth domain differs from conventional extrapolation in that the reflection coefficients are in general smaller in the pseudo-depth domain. The cause and application of such low reflection is also discussed in Chapter II. Chapter III discusses time-extrapolation of a wavefield using spectral method. This method is the basic numerical tool used in this thesis. Extrapolations in both conventional space domain and pseudo-depth domain

are solved using a spectral method. Although very accurate, spectral extrapolation is limited by excessive computational cost. The low-rank approximation of the Fourier integral operators provides a way to approximate spectral extrapolation with reasonable computational cost. To investigate the possibility of doing seismic imaging in the pseudo-depth domain, Chapter IV demonstrates examples of Zero-offset Migration (ZOM) in the pseudo-depth domain. To validate the accuracy of pseudo-depth domain imaging, the migration images are compared with those migrated in conventional space domain. Finally, Chapter V concludes this thesis by discussing the benefits and limitations of the pseudo-depth domain extrapolation.

CHAPTER II

Acoustic Wave in Pseudo-depth Domain

II.1 Vertical Time and Pseudo-depth

Seismic imaging methods, depending on the domain in which they work, may be classified as “depth imaging” or “time imaging” (Robein, 2010). Depth imaging refers to the operations which produce a subsurface reflectivity image with the vertical axis being depth z . Time imaging, in contrast, produce a subsurface image with the vertical axis being of time unit, called *vertical time*. The concept of vertical time was introduced by Claerbout (1985b) as the *retarded* time between a vertically moving coordinate system and the fixed spatial coordinate system. The vertical time is defined as the (usually two-way) travelttime between a surface point $(x, z = 0)$ with a subsurface point (x, z) .

$$\tau_{TW}(x, z) = 2 \int_0^z \frac{dz'}{v_V(x, z')}$$

where v_V is the vertical velocity. In isotropic media with acoustic assumption, it equals to P-wave velocity v . In the present study, the vertical time τ is defined as the one-way travelttime

$$\tau(x, z) = \int_0^z \frac{dz'}{v(x, z')} \quad (\text{II.1})$$

This simplifies subsequent equations but retains the features of the two-way vertical time.

Equation (II.1) is integrated on a vertical raypath. In a real subsurface medium, a straight vertical raypath is in general not physical, because it violates Snell's law. Snell's law states that the *lateral slowness* is preserved when seismic waves propagate to an interface of distinct media elasticity, where the rays experience reflection and refraction. (Sheriff and Geldart, 1995). Figure II.1 illustrates the difference between a vertical ray and a bending ray. According to Snell's law, a ray bends closer to the

Figure II.1: (a) A velocity model with $v = 1/(4 \times 10^{-4} + 8 \times 10^{-7}x)$ m/sec, and acoustic wave due to a point source at (600 m, 0 m). (b) The physical ray (red) that honors Snell's law and the vertical ray (green) that defines vertical time τ . After an elapsed time of 0.625 sec, the vertical ray points to (600 m, 710 m) while the physical ray deviates to (788 m, 635 m)

normal direction of the interface when it propagates into a slower medium from a faster medium. In this example the normal direction is the x direction, with velocity decreasing in positive x direction. As a result, a ray started in vertical direction bends toward positive x as it propagates in the medium. The vertical ray (green) does not change direction with velocity variation. The only possibility that such vertical rays are physical is when velocity changes only with depth z , in other words, the subsurface structure is a stack of flat layers. Figure II.2 illustrates an example of an overlapping physical and vertical rays. In this example, because the incident angle is always 0° , the ray does not bent in this medium.

Figure II.2: (a) A velocity model with $v = 595 + 2.38z$ m/sec and acoustic wave due to a point source located at (600 m, 0 m). (b) The physical raypath and vertical raypath are identical in direction and length. Both rays are traced for 0.5 sec traveltime.

The target volume of a seismic survey is about tens of kilometers wide on surface distance and a few kilometers in depth. Since this is a small fraction of the total upper crust (compared to Earth's radius 6370 km), Cartesian coordinate system suffices the purpose of seismic imaging. In three dimensional space, x and y are the two orthogonal axes that define horizontal plane and axis z is depth. The z axis is positive downward because the prospecting target is below the surface. Vertical depth is commonly used for seismic modeling and imaging purposes because it is a straightforward mapping of subsurface space. From the computational point of view, however, this is not an optimal representation. This is because a monochromatic seismic wave changes its wavelength as it propagates in the Earth: wavelength λ changes proportionally with local elastic velocity v by $\lambda = v/f$. Although elasticity of the subsurface rocks are quite inhomogeneous, the wave velocity v is in general increasing with depth z because Lamé parameters of the rocks increases with higher orders of density. As a result, when modeling elastic wave on regularly sampled grids, as commonly practiced in seismic imaging, the sampling of a seismic wave is also increasing with depth z . Most numerical tools require certain minimum spatial sampling of the wavefield in order to avoid spatial aliasing and numerical dispersion. The exact parameterization depends on different numerical schemes and the frequency range of

seismic wave. With classic finite difference, usually 10 to 20 samples per wavelength are needed. The overall sampling rate is determined by the shortest wavelength, which occurs at shallow slow layers. This in general causes over-sampling for deep layers where phase velocities are faster. Such increased sampling is a pure increase of computation cost and it does not enhance resolution of the final seismic image.

For example, consider the two-layer velocity model in Figure II.3, a Ricker wavelet with peak frequency of 10 Hz is propagating vertically in this model. In the shallow

Figure II.3: Left: A two-layer velocity model with a shallow layer of 1500 m/sec and a deep layer of 4000 m/sec. The vertical axis is depth with sampling interval $\Delta z = 10$ m. Right: The waveform of a Ricker wavelet with peak frequency of 10 Hz. The wavelength of this wavelet is 150 m at shallow layer and 400 m at deep layer. As a result, each wavelength has 15 samples at shallow layer and 40 samples at deep layer.

layer each wavelength of a 10 Hz sine has 15 samples. As the wave penetrates into lower layer where the velocity increases to 4000 m/sec, the wavelength of the same sine is also increased and thus each wavelength has 40 samples. For conventional FD, 10 to 20 samples per wavelength is usually sufficient to avoid dispersion. Over-sampling a wavefield does not help the imaging, since the resolution of a subsurface image is of the same scale as the wavelength. The 10 Hz wavelet is apparently oversampled in the deep fast layer. Using larger sampling interval Δz could avoid such oversampling, however it causes aliasing at shallow layer because of insufficient sampling per

Figure II.4: Left: The same velocity model as plotted in Figure II.3, the vertical axis is transformed into vertical time, with sampling interval $\Delta\tau = 6.667$ msec. Right: The waveform of the same Ricker wavelet. The wavelength of this wavelet is 0.1 sec in both layers. As a result, each wavelength is sampled by 15 samples in both layers.

wavelength. A uniformly sampled grid will not resolve both issues at the same time. Non-uniform grids avoid both issues, however, it introduces extra computations due to interpolation between the coarse and dense sampled regions (Berger and Olinger, 1984).

Using the vertical time axis, we can avoid both aliasing and oversampling simultaneously. To see why this is the case, consider the same velocity model with vertical axis being τ defined by Equation (II.1), as illustrated in Figure II.4. The “wavelength” in this domain is 0.1 sec and is independent of media velocity. Therefore, the sampling per wavelength is 15 samples in both shallow and deep layers.

In inhomogeneous media, in general, the sampling per wavelength is not exactly constant even if the vertical axis is vertical time. This is because rays are not exactly vertical unless the velocity does not change in x direction. However, the vertical time axis is still an effective way to reduce the oversampling since the local velocity of subsurface Earth is in general increasing with depth.

Using the vertical time coordinate, Alkhalifah et al. (2001) derived an eikonal equation and an acoustic wave equation in the (x, τ) domain. He later developed

a full time-processing scheme for inhomogeneous media, including ray-tracing and time-extrapolation of acoustic wave (Alkhalifah, 2003, 2005). In the vertical time representation, the Laplacian operator $\nabla^2 = \partial_x^2 + \partial_\tau^2$ features asymmetric units of space and time. It is not convenient to analyzing a differential equation with asymmetric derivatives. It is therefore desirable to express τ in space unit. To this end, the pseudo-depth z^* is introduced as the vertical time τ scaled by a constant velocity v_0

$$z^*(z) = v_0\tau(z) = v_0 \int_0^z \frac{dz'}{v(z')} \quad (\text{II.2})$$

II.2 Domain Transformation

Differentiation of Equation (II.2) with respect to z shows

$$\frac{dz^*}{dz} = \frac{v_0}{v(z)} \quad \text{and} \quad \frac{dz}{dz^*} = \frac{v(z^*)}{v_0} \quad (\text{II.3})$$

Integrating the second equation from $z^* = 0$ to z^* , it follows

$$z(z^*) = \frac{1}{v_0} \int_0^{z^*} v(z^{*'}) dz^{*'} + C \quad (\text{II.4})$$

The integration constant C vanishes because both vertical axis z and z^* are zero at the surface, $z(z^* = 0) = 0$ and $z^*(z = 0) = 0$. Therefore, given pseudo-depth z^* and velocity $v(z^*)$, the physical depth z can be uniquely determined by evaluating (II.4). Equation (II.2) and (II.4) form the mapping equations to transform any space function $f(x, z)$ between the space domain (x, z) and pseudo-depth domain (x, z^*) . Such functions can be a velocity model, reflectivity or P-wavefield and S-wavefield.

To use the transformation given by Equations (II.2) and (II.4) in practice, one needs to consider two other aspects. Firstly, the pseudo-depth z^* computed from Equation (II.2) varies at different surface location (x, y) . Thus, the length of the

pseudo-depth axis for the entire model is determined by the maximum of z^* , i.e. the surface location (x, y) at which the vertical ray is slowest from the surface to the bottom of the model. Secondly, because the transformations are nonlinear operations, the pseudo-depth z^* computed from Equation (II.2) is in general not uniformly sampled. To produce a transformed model on a regularly sampled grid, it is necessary to interpolate the direct transformation result on a regular grid.

In summary, the complete procedure to transform any space function $f(x, z)$ into pseudo-depth domain $f(x, z^*)$ has following steps:

Step 1 Compute pseudo-depth $z^*(x, z)$ from given velocity model $v(x, z)$ using Equation (II.2).

Step 2 Choose the size and sampling interval of a regularly sampled pseudo-depth axis z_r^* such that the maximum value computed from Step 1 lies within the range of z_r^* .

Step 3 Interpolate the input function $f(x, z) = f(x, z^*)$ to the new domain $f(x, z_r^*)$

Similarly, to transform a function in pseudo-depth domain $f(x, z^*)$ back to the space domain, one needs to follow the same procedure with the only difference that in Step 1 Equation (II.4) is used instead of using Equation (II.2).

As an example of the transformation between space domain and pseudo-depth domain, consider the Marmousi velocity model in Figure II.5(a). Extensive faults and folds of this model makes it a challenging case to image accurately. When transformed into pseudo-depth domain using Equation (II.2), the result is shown in Figure II.5(c). The prominent “pull-up” between $x = 4$ km and $x = 7$ km is due to shorter total vertical time in this region. The vertical lines that appear in the bottom of Figure II.5(c) are a consequence of the fact that the maximum vertical time $\tau_{max}(x)$ varies with different surface location x . For example, from $x = 3$ km to $x = 6$ km, the maximum vertical time τ_{max} changed from 1.231 sec to 1.092 sec. The maximum

Figure II.5: (a) The Marmousi velocity model and (b) vertical time computed on this model. (c) Transformation of (a) into pseudo-depth domain using scaling velocity $v_0 = 1500$ m/sec. The vertical layers at the bottom of (b) are because of extrapolating the bottom velocities.

z^* is determined by the maximum value of all $\tau_{max}(x)$. If a vertical ray reaches the bottom of the model with shorter time than τ_{max} , the value of the function beyond bottom of the model is simply a repetition of the value at bottom. This causes the vertical lines appeared in bottom of Figure II.5(c).

If we further apply the inverse transformation given in Equation (II.4) to the pseudo-depth domain velocity model, the result is shown in Figure II.6(a). Figure II.6(b) shows the difference between the inverse transformation result and the original space domain velocity model. Such difference is caused by the interpolation process in the

Figure II.6: (a) The pseudo-depth velocity in Figure II.5(c) transformed back to space domain. (b) The relative error between II.6 (a) and original velocity in Figure II.5(a). The difference is mainly due to interpolation during the forward and inverse transforms. The difference is significant only at individual grids that are located near velocity interfaces.

forward and inverse transforms that may miss or add a few samples in the vicinity of a velocity interface. It may seem significant (20%) but it is mainly localized to individual samples on reflector interfaces. In general, the inverse transformation restores the pseudo-depth domain velocity model to its space domain equivalence.

The choice of v_0 in Equation (II.2) and Equation (II.4) is trivial because it only affects the readings on the pseudo-depth axis. For example, Figure II.7(a) is transformed to pseudo-depth domain using $v_0 = 1500$ m/sec, while Figure II.7(b) is transformed using $v_0 = 15$ m/sec. In both cases the z^* axis is discretized to 751 samples. Using a faster v_0 results in larger Δz^* , a smaller v_0 results in smaller Δz^* sampling interval. The difference between two mapping results using different v_0 is plotted in Figure II.7(c). This difference is less than 0.5% of the model velocity.

The transformation between space domain and pseudo-depth domain is effective for any spatial function, for example velocity, density and wavefield. Figure II.8

Figure II.7: Marmousi model transformed to pseudo-depth domain using (a) $v_0 = 1500$ m/sec and (b) $v_0 = 15$ m/sec. The pseudo-depth axis z^* is discretized into 751 samples in both figures. The sampling interval is $\Delta z^* = 2.6$ m in (a) and $\Delta z^* = 0.026$ m in (b). (c) The small difference between (a) and (b) confirms that choice of scaling velocity does not affect the transformed image.

shows an example of transforming acoustic wave field between the two domains. A point source is placed at (200 m, 150 m) of a medium with velocity changes linearly in x . By applying the forward transformation, the wavefield in Figure II.8(a) is transformed into pseudo-depth domain as illustrated in Figure II.8(b). To go back to the space domain, we simply apply the inverse transform and the results is plotted in Figure II.8(c). Figure II.8(d) shows the difference between the result of inverse transform and the original wavefield. Like transformation of velocity models, such difference is mainly due to the interpolation process and is limited on individual

samples in the vicinity of the velocity interface.

Figure II.8: Transforming wavefield between space domain and pseudo-depth domain. The medium velocity is $v = 1/(5 \times 10^{-3} - 6.25 \times 10^{-6}z)$ m/sec. The wavefield is induced by a points source located at $(x = 200 \text{ m}, z = 200 \text{ m})$. (a) A snapshot of wavefield in space domain. (b) The same wavefield transformed to pseudo-depth domain. (c) Inverse transform of pseudo-depth domain wavefield. (d) Difference between inverse transform (c) and original wavefield (a). The difference is limited to the amplitude of the wavefront while the location of the wavefront is identical after the two transformations.

II.3 Extrapolation in the Pseudo-depth Domain

To find a differential equation that governs the wave propagation in the pseudo-depth domain, it is necessary to transform the differentiation operation to pseudo-depth domain. Let (x^*, z^*) be pseudo-depth domain and (x, z) be the physical space

domain. Since the pseudo-depth mapping is a 1-D operation in the vertical direction, the x -axis is preserved in both domains, $x^* = x$. By applying the chain rule to the partial derivatives $\partial/\partial x$ and $\partial/\partial z$ and utilizing the fact that $\partial x^*/\partial x = 1$ and $\partial x^*/\partial z = 0$, it follows that

$$\partial_x = \partial_{x^*} + \partial_x z^* \partial_{z^*} \quad (\text{II.5a})$$

$$\partial_z = \frac{v_0}{v} \partial_{z^*} \quad (\text{II.5b})$$

The term $\partial z^*/\partial x$ is a measure of the lateral change of vertical time τ . Denote the x -gradient of vertical time τ by σ , by differentiating Equation (II.1) with respect to x

$$\sigma(x, z) = \frac{\partial \tau(x, z)}{\partial x} = \frac{\partial}{\partial x} \int_0^z \frac{1}{v(x, z')} dz' \quad (\text{II.6})$$

σ has slowness unit sec/meter. Denote slowness by $s = 1/v$, then Equation (II.5) can be combined as following

$$\nabla = \begin{pmatrix} \partial_x \\ \partial_z \end{pmatrix} = \begin{pmatrix} 1 & v_0 \sigma \\ 0 & v_0 s \end{pmatrix} \begin{pmatrix} \partial_{x^*} \\ \partial_{z^*} \end{pmatrix} = \mathbf{J} \nabla^* \quad (\text{II.7})$$

The gradient operators in the two domains are related by Jacobian matrix $\mathbf{J}$.

A sound wave in a heterogeneous medium is governed by the acoustic wave equation (Aki and Richards, 1980)

$$\frac{1}{v^2} \frac{\partial^2 P(\mathbf{x}, t)}{\partial t^2} = \nabla^2 P(\mathbf{x}, t) \quad (\text{II.8})$$

where ∇ is the gradient operator and v is the P-wave velocity. Using Equation (II.7),

the Laplacian operator ∇^2 is transformed to pseudo-depth domain as following

$$\begin{aligned}
\nabla^2 &= \nabla^{*T} \mathbf{J}^T \mathbf{J} \nabla^* \\
&= \begin{pmatrix} \partial_{x^*} & \partial_{z^*} \end{pmatrix} \begin{pmatrix} 1 & v_0 \sigma \\ v_0 \sigma & v_0^2 (\sigma^2 + s^2) \end{pmatrix} \begin{pmatrix} \partial_{x^*} \\ \partial_{z^*} \end{pmatrix} \\
&= \partial_{x^*}^2 + v_0 [\partial_{x^*} (\sigma \partial_{z^*}) + \partial_{z^*} (\sigma \partial_{x^*})] + v_0^2 \partial_{z^*}^2 (\sigma^2 + s^2) \partial_{z^*}
\end{aligned} \tag{II.9}$$

Substitute Equation (II.9) into Equation (II.8) reveals the pseudo-depth acoustic wave equation for constant density

$$\frac{1}{v^2} \frac{\partial^2 P}{\partial t^2} = \frac{\partial^2 P}{\partial x^2} + v_0 \left[\frac{\partial}{\partial x} \left(\sigma \frac{\partial P}{\partial z} \right) + \frac{\partial}{\partial z} \left(\sigma \frac{\partial P}{\partial x} \right) \right] + v_0^2 \frac{\partial}{\partial z} \left((\sigma^2 + s^2) \frac{\partial P}{\partial z} \right) \tag{II.10}$$

The superscript $*$ has been dropped for simplicity. Define following spatial Fourier transform of $P(x, z, t)$

$$\hat{P}(k_x, k_z, t) = \int_z \int_x P(x, z, t) e^{i(k_x x + k_z z)} dx dz$$

The differentiation $\partial/\partial x$ then becomes multiplication by ik_x in the Fourier domain. For wavenumbers greater than 1 this implies second order differentiation is greater than first order $|\partial_x^2 \hat{P}| > |\partial_x \hat{P}|$. This allows simplifying Equation (II.10) by dropping terms of first order derivatives of P . The pseudo-depth acoustic wave equation is reduced to

$$\frac{1}{v^2} \frac{\partial^2 P}{\partial t^2} = \frac{\partial^2 P}{\partial x^2} + 2v_0 \sigma \frac{\partial^2 P}{\partial x \partial z} + v_0^2 \left(\sigma^2 + \frac{1}{v^2} \right) \frac{\partial^2 P}{\partial z^2} \tag{II.11}$$

This can be viewed as approximating the true solution by its high spatial frequency components.

The wavefield modeled in pseudo-depth domain, when transformed back to space domain, is expected to agree with conventional acoustic modeling, if Equation (II.11) is correct. To validate Equation (II.11), consider the velocity model plotted in Fig-

ure II.9(a). The velocity features lateral variation. Such x dependence is retained

Figure II.9: (a) A velocity model with $v = 1/(5 \times 10^{-3} - 6.25 \times 10^{-7}x)$ m/sec and (b) its transformation in pseudo-depth domain using $v_0 = 200$ m/sec.

when transformed to pseudo-depth domain, as plotted in Figure II.9(b), since the transformation is essentially a one-dimensional operation in z direction.

A snapshot of conventional acoustic modeling is shown in Figure II.10(a). Using the pseudo-depth domain acoustic wave equation (II.11), a snapshot of the modeling result is shown in Figure II.10(b). Compare to the space domain modeling, the error is around 15%. This error is due to both the interpolation process during the transformation, as explained earlier, and the low reflection nature that will be discussed later. The error may seem profound, however, such error is mainly from the amplitude difference. In exploration geophysics, traveltimes of a seismic event is often considered more important than amplitude. The amplitude of a seismic event is not as accurate as its phase due to the different types of noise during acquisition. During processing, the wavefield is often filtered in different processing steps, this may also change the amplitude of the true seismic wave.

Fourier transform of Equation (II.11) in both space and time domain shows the dispersion relation for pseudo-depth domain acoustic wave equation

$$\omega^2 = v^2 (k_x + v_0 \sigma k_z)^2 + v_0^2 k_z^2 \quad (\text{II.12})$$

Figure II.10: Continued from Figure II.9. (a) Snapshot of conventional acoustic modeling. The source is a point located at (200 m, 200 m). (b) Snapshot of pseudo-depth domain modeling. The source location is mapped to (200 m, 150 m). (c) The pseudo-depth domain wavefield in (b) transformed to space domain. (d) The difference between two snapshots (a) and (c).

the dispersion relation of the conventional acoustic wave equation is simply

$$\omega^2 = v^2 (k_x^2 + k_z^2) \quad (\text{II.13})$$

Equation (II.12) reduces to Equation (II.13) when $\sigma = 0$ and $v_0 = v$. The fact that Equation (II.12) is asymmetric in k_x and k_z implies that the pseudo-depth wave equation (Equation (II.11)) features anisotropy, i.e. the wave propagates at different velocities in different directions. Figure II.11 illustrates the examples of dispersion curves for the pseudo-depth wave equation.

Figure II.11: Dispersion curves computed using Equation (II.12), with scaling velocity $v_0 = 1500$ m/sec, frequency $\omega = 80\pi$. (a) In a homogeneous velocity $v = v_0$, the dispersion relation is a circle, identical to that of the conventional acoustic wave equation. (b) Homogeneous velocity $v = 2000$ m/sec. The dispersion curve is an ellipse due to $v \neq v_0$. (c) Inhomogeneous media with $v = 2000$ m/sec, $\sigma = 5 \times 10^{-4}$ sec/m. The dispersion curve is a tilted ellipse. (d) Inhomogeneous media with $v = 2000$ m/sec, $\sigma = -5 \times 10^{-4}$ sec/m.

In Equation (II.12) if $\sigma = 0$ it reduces to the following form

$$\omega^2 = v^2 k_x^2 + v_0^2 k_z^2 \quad (\text{II.14})$$

This means that the wave propagates at velocity v in the x direction, while propagates at velocity v_0 in the z direction. This is referred to elliptic anisotropy. Physically $\sigma = 0$ means that vertical time τ is a function of depth z only, and thus the velocity is a function of z only. This can be seen by substituting $v = v(z)$ into Equation (II.6). Substitute $\sigma = 0$ into Equation (II.11) shows the pseudo-depth wave equation corre-

sponding in $v = v(z)$ media

$$\frac{\partial^2 P}{\partial t^2} = v^2 \frac{\partial^2 P}{\partial x^2} + v_0^2 \frac{\partial^2 P}{\partial z^2} \quad (\text{II.15})$$

An important feature of the solution of Equation (II.15) is that normal incidence to a flat reflector will not result in any reflection. This is because the phase velocity in vertical direction is fixed at v_0 despite of its true P-wave velocity v . Assuming constant density, the reflection coefficient for a normal incidence is given by

$$r = \frac{v_2 - v_1}{v_2 + v_1} \quad (\text{II.16})$$

In a lateral homogeneous media, the reflection coefficient r vanishes because both v_2 and v_1 equal to v_0 . Figure II.12 shows an example of zero reflection at normal incidence at a flat reflector. The velocity model is composed of two flat layers, with slow velocity 1500 m/sec at shallow and fast velocity 3000 m/sec at depth. Because the vertical time is shorter in fast media, the bottom layer is “compressed” after the transformation to pseudo-depth domain.

If the reflector is not flat, the reflection coefficients are expected to be smaller as the ones with a flat reflector. Figure II.13 shows an example of attenuated reflections on a dipping reflector. In general, the reflection coefficient in the pseudo-depth domain wavefield is smaller than space domain extrapolation.

The low reflection feature provides a potential application in wave-equation based migration. This is related to the commonly used cross-correlation imaging condition (see Section IV.1). Reflections from strong reflectors, for example a salt dome, enhances the image artifacts caused by cross-correlation imaging condition. Conventionally, the non-reflecting wave equation (Baysal et al., 1984) is a measure to reduce unwanted reflections. However, this method artificially modifies the density to compensate velocity contrast. Using the pseudo-depth wave equation, the same

low reflection effect can be achieved without modifying the density model.

Figure II.12: (a) A velocity model with two flat layers. The velocity is 1500 m/sec in the shallow layer and 3000 m/sec in the deep layer. (b) Snapshot of the conventional acoustic modeling result. The wavefield is induced by a point source located at (1000 m, 600 m). (c) The same velocity model transformed into pseudo-depth domain using $v_0 = 1500$ m/sec. (d) Snapshot of pseudo-depth domain modeling. The source location is mapped to (1000 m, 600 m). The wavefield is transformed back to space domain to compare with (b)

Figure II.13: (a) A velocity model has two layers separated by a dipping reflector. Velocity is 1500 m/sec in the shallow layer and 3000 m/sec in the deep layer. (b) Snapshot of conventional space domain modeling. The source is a point located at (1000 m, 600 m). (c) The same velocity model transformed into pseudo-depth domain using $v_0 = 1500$ m/sec. (d) Snapshot of pseudo-depth domain modeling. The source location is mapped to (1000 m, 600 m).

CHAPTER III

Numerical Methods

III.1 Numerical Methods for Seismic Modeling

The full-wave equation for compressional waves (P-waves) in an inhomogeneous medium is written as

$$\partial_t^2 P(\mathbf{x}, t) = \mathcal{L}^2 P(\mathbf{x}, t) + f(\mathbf{x}, t) \quad (\text{III.1})$$

with

$$\mathcal{L}^2 u(\mathbf{x}) = \rho(\mathbf{x}) v^2(\mathbf{x}) \nabla \cdot \left(\frac{1}{\rho(\mathbf{x})} \nabla u(\mathbf{x}) \right)$$

where ∇ is the gradient operator, $P(\mathbf{x}, t)$ is the pressure field, $v(\mathbf{x})$ is the P-wave velocity, $\rho(\mathbf{x})$ is the density and $f(\mathbf{x}, t)$ is the body force. Equation (III.1) is a partial differential equation with second-order derivatives in both space $\mathbf{x}$ and time t . It can be solved directly using well established numerical methods such as *finite difference methods*, *finite element methods*, *pseudospectral methods* and *spectral methods*.

The spatial derivative in Equation (III.1), when expanded into the so called non-conservative form, becomes

$$\mathcal{L}^2 P = v^2 (\nabla^2 P - \rho^{-1} \nabla \rho \cdot \nabla P)$$

Within the range of seismic prospecting, usually several kilometers under the Earth's surface, density ρ varies from about 2.0 g/cm³ to 2.6 g/cm³. Because the density contrast is much smaller than the velocity change in the subsurface, in practice, density is often treated as a constant. By doing so, this operator is simplified to

$$\mathcal{L}^2 = v^2 \nabla^2.$$

Modeling seismic waves is equivalent to solving the following initial-value problem on the model domain $\mathcal{D}$

$$\begin{cases} \partial_t^2 P(\mathbf{x}, t) = v(\mathbf{x})^2 \nabla^2 P(\mathbf{x}, t) + f(\mathbf{x}, t) & \mathbf{x} \in \mathcal{D}, t > 0 \\ P(\mathbf{x}, t = 0) = P_0(\mathbf{x}) & \mathbf{x} \in \mathcal{D} \\ \partial_t P(\mathbf{x}, t = 0) = P_1(\mathbf{x}) & \mathbf{x} \in \mathcal{D} \end{cases} \quad (\text{III.2})$$

Non-reflecting boundary conditions such as one-way wave equation (Clayton and Engquist, 1980) and perfectly matched layers (Berenger, 1994) are usually imposed on the boundary $\partial\mathcal{D}$ to simulate the semi-infinite subsurface space. The top surface is a total reflecting boundary because of the large impedance contrast between air and water (or land). Initial values P_0 and P_1 are usually set to zero because physically the disturbance happens only after a source is applied to the medium. f is a pressure source. For land seismic survey this is usually some mixed-phase wavelet. For marine survey it can be more complex due to reflection from water surface. The actual waveform of the source wavelet is not important for seismic modeling purpose. This is because the phase of seismic wave carries more important information than its amplitude (Sheriff and Geldart, 1995). As a result, $f(t)$ is almost always assumed as a Ricker wavelet because it is minimum-phase and band-limited. A slightly different scenario of seismic modeling is the *exploding reflector modeling* or *zero-offset modeling*, in this case the source is represented by nonzero P_0 while leaving $f = 0$. Zero-offset modeling and migration are described in Section IV.1.

III.2 Spectral Extrapolation

III.2.1 Theory

The full-wave equation (III.1) for constant density can be formulated as a system of differential equations

$$\partial_t \mathbf{u} = \mathbf{H} \mathbf{u} + \mathbf{f} \quad (\text{III.3})$$

where

$$\mathbf{u}(\mathbf{x}, t) = \begin{bmatrix} P(\mathbf{x}, t) \\ \partial_t P(\mathbf{x}, t) \end{bmatrix}, \quad \mathbf{f}(\mathbf{x}, t) = \begin{bmatrix} 0 \\ f(\mathbf{x}, t) \end{bmatrix}, \quad \mathbf{H}(\mathbf{x}) = \begin{bmatrix} 0 & I \\ v^2(\mathbf{x}) \nabla^2 & 0 \end{bmatrix}$$

The system (III.3) is first-order in time and second-order in space. Its solution subject to initial value $\mathbf{u}(\mathbf{x}, t = 0) = \mathbf{u}_0(\mathbf{x})$ can be written as

$$\mathbf{u}(\mathbf{x}, t) = e^{\mathbf{H}(\mathbf{x})t} \mathbf{u}_0(\mathbf{x}) + \int_0^t e^{\mathbf{H}(\mathbf{x})\tau} \mathbf{f}(\mathbf{x}, t - \tau) d\tau \quad (\text{III.4})$$

This formula can not be evaluated directly due to the symbolic operator ∇^2 in matrix $\mathbf{H}$. Because differentiation becomes multiplication in the Fourier domain, we can still use Equation (III.4) to compute the solution in the Fourier domain. Consider constant velocity $v(\mathbf{x}) = v$, taking the spatial Fourier transform of the full-wave equation (III.2), it follows

$$\partial_t^2 \hat{P}(\mathbf{k}, t) = -v^2 |\mathbf{k}|^2 \hat{P}(\mathbf{k}, t) + \hat{f}(\mathbf{k}, t) \quad (\text{III.5})$$

where $|\mathbf{k}| = \sqrt{\mathbf{k} \cdot \mathbf{k}}$ is the L_2 norm of the wavenumber vector $\mathbf{k}$. $\hat{P}(\mathbf{k}, t)$ is the spatial Fourier transform of $P(\mathbf{x}, t)$. The Fourier transform pair in this thesis is defined by

$$\hat{u}(\mathbf{k}) = \mathcal{F}(u) = \int_{\mathbf{x}} u(\mathbf{x}) e^{-i\mathbf{k} \cdot \mathbf{x}} d\mathbf{x} \quad (\text{III.6})$$

$$u(\mathbf{x}) = \mathcal{F}^{-1}(\hat{u}) = \frac{1}{(2\pi)^d} \int_{\mathbf{k}} \hat{u}(\mathbf{k}) e^{i\mathbf{k} \cdot \mathbf{x}} d\mathbf{k} \quad (\text{III.7})$$

where $i = \sqrt{-1}$, $\mathbf{x}$ is d -dimensional space domain coordinate and $\mathbf{k}$ is its Fourier domain coordinate.

Similar to Equation (III.3), we can write the first-order system for Equation (III.5). Because of linearity, we are allowed to consider a single wavenumber $\mathbf{k}$ in the Fourier domain; the system (III.3) is then composed of first-order ordinary differential equations

$$\frac{d\hat{\mathbf{u}}}{dt} = \hat{\mathbf{H}}\hat{\mathbf{u}} + \hat{\mathbf{f}} \quad (\text{III.8})$$

where

$$\hat{\mathbf{u}}(\mathbf{k}, t) = \begin{bmatrix} \hat{P}(\mathbf{k}, t) \\ \partial_t \hat{P}(\mathbf{k}, t) \end{bmatrix}, \quad \hat{\mathbf{f}}(\mathbf{k}, t) = \begin{bmatrix} 0 \\ \hat{f}(\mathbf{k}, t) \end{bmatrix}, \quad \hat{\mathbf{H}}(\mathbf{k}) = \begin{bmatrix} 0 & I \\ -v^2 |\mathbf{k}|^2 & 0 \end{bmatrix}$$

The eigendecomposition of matrix $\hat{\mathbf{H}}$ is

$$\hat{\mathbf{H}} = \frac{1}{2} \begin{bmatrix} I & I \\ \lambda & -\lambda \end{bmatrix} \cdot \begin{bmatrix} \lambda & 0 \\ 0 & -\lambda \end{bmatrix} \cdot \begin{bmatrix} I & \lambda^{-1} \\ I & -\lambda^{-1} \end{bmatrix} \quad (\text{III.9})$$

where $\lambda = iv|\mathbf{k}|$. $\hat{\mathbf{H}}$ has eigenvalues $\pm\lambda$. Since both eigenvalues are imaginary, the so called *evanescent* waves with exponential growth or decay of amplitude are excluded

from the solution to system (III.8). The matrix exponential $e^{\hat{\mathbf{H}}t}$ is computed by

$$\begin{aligned} e^{\hat{\mathbf{H}}t} &= \frac{1}{2} \begin{bmatrix} I & I \\ \lambda t & -\lambda t \end{bmatrix} \cdot \begin{bmatrix} e^{\lambda t} & 0 \\ 0 & e^{\lambda t} \end{bmatrix} \cdot \begin{bmatrix} I & (\lambda t)^{-1} \\ I & -(\lambda t)^{-1} \end{bmatrix} \\ &= \begin{bmatrix} \cos(v|\mathbf{k}|t) & (v|\mathbf{k}|)^{-1} \sin(v|\mathbf{k}|t) \\ -v|\mathbf{k}| \sin(v|\mathbf{k}|t) & \cos(v|\mathbf{k}|t) \end{bmatrix} \end{aligned} \quad (\text{III.10})$$

Substitute Equation (III.10) into Equation (III.4) shows the complete solution

$$\begin{aligned} \hat{P}(\mathbf{k}, t) &= \hat{P}(\mathbf{k}, 0) \cos(v|\mathbf{k}|t) + \partial_t \hat{P}(\mathbf{k}, 0) \frac{\sin(v|\mathbf{k}|t)}{v|\mathbf{k}|} \\ &\quad + \frac{1}{v|\mathbf{k}|} \int_0^t \sin(v|\mathbf{k}|\tau) \hat{f}(\mathbf{k}, t - \tau) d\tau \end{aligned} \quad (\text{III.11})$$

Since particle velocity is zero at $t = 0$, it follows that $\partial_t \hat{P}(\mathbf{k}, 0) = 0$. We can also neglect the source term f because it is more conveniently implemented in the initial value $\hat{P}(\mathbf{k}, 0)$. This simplifies Equation (III.11) to

$$\hat{P}(\mathbf{k}, t) = \hat{P}(\mathbf{k}, 0) \cos(\iota v|\mathbf{k}|t)$$

The cosine function here can be replaced by a complex exponential because the real part is identical in both cases. Thus

$$\hat{P}(\mathbf{k}, t) = \hat{P}(\mathbf{k}, 0) e^{\iota v|\mathbf{k}|t} \quad (\text{III.12})$$

This allows us to compute the solution at time $t + \Delta t$ with the knowledge of the solution at an earlier time t , in other words, *extrapolate* the solution in time

$$\hat{P}(\mathbf{k}, t + \Delta t) = \hat{P}(\mathbf{k}, t) e^{\iota v|\mathbf{k}|\Delta t} \quad (\text{III.13})$$

The space domain solution is computed by taking inverse Fourier transform of (III.13)

$$\begin{aligned} P(\mathbf{x}, t + \Delta t) &= \frac{1}{(2\pi)^d} \int \hat{P}(\mathbf{k}, t) e^{i(\mathbf{k} \cdot \mathbf{x} + v|\mathbf{k}|\Delta t)} d\mathbf{k} \\ &= \mathcal{F}^{-1} \{ \mathcal{F} \{ P(\mathbf{x}, t) \} e^{i v |\mathbf{k}| \Delta t} \} \end{aligned} \quad (\text{III.14})$$

where $\mathcal{F}$ and $\mathcal{F}^{-1}$ stand for forward and inverse spatial Fourier transforms defined by (III.6) and (III.7).

To compute the wavefield extrapolation using Equation (III.14) on a discrete grid, the Fourier transform needs to be discretized. Consider a two-dimensional grid (x_i, z_j)

$$\begin{aligned} x_i &= i\Delta x, \quad i = \{0, \dots, N_x - 1\} \\ z_j &= j\Delta z, \quad j = \{0, \dots, N_z - 1\} \end{aligned}$$

To achieve optimal computation speed of Fast Fourier Transform (FFT), the computation domain is usually expanded by padding zeros to end of each axis, so that the size of the expanded axis can be factored by prime numbers 2, 3 and 5. Most standard FFT solvers can calculate the optimal FFT size. For simplicity, the grid (x_i, z_j) cited here is assumed to be expanded by zero-padding. The wavenumber domain grid is defined as follows

$$\begin{aligned} \xi_p &= \frac{2\pi p}{N_x \Delta x}, \quad p = \{0, \dots, N_x - 1\} \\ \eta_q &= \frac{2\pi q}{N_z \Delta z}, \quad q = \{0, \dots, N_z - 1\} \end{aligned}$$

Define the following two-dimensional discrete Fourier transform (DFT) pair

$$\hat{P}_{pq} = \sum_{j=0}^{N_z-1} \sum_{i=0}^{N_x-1} P_{ij} e^{-\iota(i\xi_p + j\eta_q)} \quad (\text{III.15})$$

$$P_{ij} = \frac{1}{N_x N_z} \sum_{q=0}^{N_z-1} \sum_{p=0}^{N_x-1} \hat{P}_{pq} e^{\iota(i\xi_p + j\eta_q)} \quad (\text{III.16})$$

where $P_{ij} = P(x_i, z_j)$ and $\hat{P}_{pq} = \hat{P}(\xi_p, \eta_q)$ are the discrete samples in space and wavenumber domain. Based on Equation (III.14) and the DFT defined in (III.15) and (III.16), a numerical algorithm to compute the time-extrapolation of the seismic wavefield in constant velocity can be described as follows

- 1 Given solution at current time step: $P_{ij}^n = P(x_i, z_j, t_n)$, compute two-dimensional DFT defined by (III.15): $P_{ij}^n \rightarrow \hat{P}_{pq}^n$. The superscript n indicates time step.
- 2 Extrapolate one time step in the wavenumber domain by computing $\hat{P}_{pq}^{n+1} = \hat{P}_{pq}^n e^{\iota 2\pi v \sqrt{\xi_p^2 + \eta_q^2} \Delta t}$
- 3 Inverse Fourier transform $\hat{P}_{pq}^{n+1}$ by using (III.16) to form the solution at future time step: $\hat{P}_{pq}^{n+1} \rightarrow P_{ij}^{n+1}$

However this method is exact only for constant velocity $v(\mathbf{x}) = v$. This is because the inverse Fourier transform requires its argument to be a function of wavenumber $\mathbf{k}$ only. In a variable velocity model, if we replace v by $v(\mathbf{x}, \mathbf{k})$ it still provides a good approximation provided that time step Δt is sufficiently fine (Etgen and Brandsberg-Dahl, 2009). Thus, the time-extrapolation formula for inhomogeneous velocity is (Etgen and Brandsberg-Dahl, 2009; Wards et al., 2008)

$$P(\mathbf{x}, t + \Delta t) = \frac{1}{(2\pi)^d} \int \hat{P}(\mathbf{k}, t) e^{\iota \mathbf{k} \cdot \mathbf{x}} \phi(\mathbf{x}, \mathbf{k}) d\mathbf{k} \quad (\text{III.17})$$

$$\phi(\mathbf{x}, \mathbf{k}) = e^{\iota v(\mathbf{x}) |\mathbf{k}| \Delta t} \quad (\text{III.18})$$

The phase-shift operator $\phi(\mathbf{x}, \mathbf{k})$ is defined on a mixed-domain of both space and wavenumber. The integral in Equation (III.17) is no longer a standard inverse Fourier transform because of the $\mathbf{x}$ -dependency of phase-shift operator.

Following similar analysis, we can write the first-order system for the pseudo-depth domain acoustic wave equation (II.11). This is in the same form as Equation (III.8), with the difference that

$$\hat{\mathbf{H}}(\mathbf{k}) = \begin{bmatrix} 0 & I \\ -[v^2(\mathbf{x})(k_x + v_0\sigma k_z)^2 + v_0^2 k_z^2] & 0 \end{bmatrix} \quad (\text{III.19})$$

where $\mathbf{k} = [k_x \ k_z]^T$ is the two-dimensional wavenumber coordinate. Similarly the eigenvalues of $\hat{\mathbf{H}}$ are found to be

$$\lambda = \pm \iota \sqrt{v^2(\mathbf{x})(k_x + v_0\sigma k_z)^2 + v_0^2 k_z^2}$$

The time-extrapolation formula (III.17) is still valid for pseudo-depth domain acoustic wave equation, with its phase-shift operator given by

$$\phi(\mathbf{x}, \mathbf{k}) = e^{\iota \sqrt{v^2(\mathbf{x})(k_x + v_0\sigma k_z)^2 + v_0^2 k_z^2} \Delta t} \quad (\text{III.20})$$

Following the same two-dimensional grids (x_i, z_j) defined for constant velocity, a numerical algorithm to compute the time-extrapolation of seismic wavefields in inhomogeneous velocity can be described as follows

- 1 Compute mixed-domain phase-shift operator defined by (III.18)

$$\phi_{ij,pq} = \phi(x_i, z_j, \xi_p, \eta_q) = e^{\iota v_{ij} \sqrt{\xi_p^2 + \eta_q^2} \Delta t}$$

for the conventional acoustic wave equation, or by (III.20)

$$\phi_{ij,pq} = e^{\iota \sqrt{v_{ij}^2(\xi_p + v_0 \sigma_{ij} \eta_q)^2 + v_0^2 \eta_q^2} \Delta t}$$

for pseudo-depth domain wave equation. This step needs to be computed only once, before starting time extrapolation.

- 2 Fourier transform the current solution: $P_{ij}^n \rightarrow \hat{P}_{pq}^n$ by using DFT given by (III.15).
- 3 Compute mixed-domain function $A_{ij,pq} = \hat{P}_{pq}^n \phi_{ij,pq}$ for every space-wavenumber grid $(x_i, z_j, \xi_p, \eta_q)$.
- 4 Form the future solution by calculating summation

$$P_{ij}^{n+1} = \frac{1}{N_x N_z} \sum_{q=0}^{N_z-1} \sum_{p=0}^{N_x-1} A_{ij,pq} e^{\iota(i\xi_p + j\eta_q)}$$

Midpoint time-stepping is popular in seismic modeling because of its second-order accuracy. The mid-point scheme corresponding to Equation (III.17) is found by substituting $t - \Delta t$ into Equation (III.17) and adding it to itself. This follows

$$P(\mathbf{x}, t + \Delta t) - 2P(\mathbf{x}, t) + P(\mathbf{x}, t - \Delta t) = \frac{1}{(2\pi)^d} \int \hat{P}(\mathbf{k}, t) e^{\iota \mathbf{k} \cdot \mathbf{x}} [2 \cos(v(\mathbf{x})|\mathbf{k}|\Delta t) - 2] d\mathbf{k} \quad (\text{III.21})$$

The midpoint time-stepping scheme corresponding to pseudo-depth wave equation is found by using the pseudo-depth phase-shift operator (III.20),

$$P(\mathbf{x}, t + \Delta t) - 2P(\mathbf{x}, t) + P(\mathbf{x}, t - \Delta t) = \frac{1}{(2\pi)^d} \int \hat{P}(\mathbf{k}, t) e^{\iota \mathbf{k} \cdot \mathbf{x}} \left[2 \cos \left(\sqrt{v^2(\mathbf{x}) (k_x + v_0 \sigma k_z)^2 + v_0^2 k_z^2} \Delta t \right) - 2 \right] d\mathbf{k} \quad (\text{III.22})$$

III.2.2 Comparison with Finite Difference Method

Conventionally, the seismic wavefield is modeled using finite difference (FD) methods. FD approximates the spatial and temporal derivatives using Taylor series expansions.

$$P(t + \Delta t) = P(t) + \sum_{k=1}^{\infty} P^{(k)}(t) \frac{\Delta t^k}{k!} \quad (\text{III.23})$$

The midpoint time-stepping based on (III.23) writes

$$\partial_t^2 P = \frac{P^{n+1} - 2P^n + P^{n-1}}{\Delta t^2} - 2 \sum_{k=2}^{\infty} \frac{\Delta t^{2(k-1)}}{(2k)!} \frac{\partial^{2k} P}{\partial t^{2k}} \quad (\text{III.24})$$

where superscripts indicate time steps. Classical FD truncates this equation at $k = 1$ and thus approximates the time derivative with $\mathcal{O}(\Delta t^2)$ accuracy. Using the full-wave equation defined by (III.2), the high order time derivatives ($k \geq 2$) can be computed recursively.

$$\partial^2 P \approx \frac{P^{n+1} - 2P^n + P^{n-1}}{\Delta t^2} + 2 \sum_{k=2}^K \frac{\Delta t^{2(k-1)}}{(2k)!} \mathcal{L}^{2k} P + \mathcal{O}(\Delta t^{2K}) \quad (\text{III.25})$$

Equation (III.25) is often referred to as *Lax-Wendroff* scheme (Dablain, 1986). This method increases the time derivative to $\mathcal{O}(\Delta t^{2K})$ accuracy, however it requires estimation of high order spatial derivatives $\mathcal{L}^{2k}$ at each time step.

The solution computed using both classical FD and Lax-Wendroff high-order schemes suffer from undesirable “ripples” called numerical dispersion noise (Alford et al., 1974). To avoid such dispersion error, the wavefield has to be sampled sufficiently fine in space. For example, Dablain (1986) showed that at least 4 samples are required at Nyquist frequency to eliminate dispersion error for fourth-order FD methods. This requirement effectively imposes a maximum frequency that can be modeled on a given discrete grid. In contrast, solutions computed using spectral method are

nearly free from numerical dispersion. A comparison of numerical dispersions is shown in Figure III.1. A point source is placed at the center of the domain. The wavefield computed using classical FD (Figure III.1(a) and III.1(c)) exhibits strong numerical dispersion. In contrast, the wavefield computed using spectral method (Figure III.1(b) and III.1(d)) is almost free from such dispersion.

Figure III.1: Snapshots of wavefields modeled using (a) classical finite difference method and (b) spectral method. The velocity model is $v = 1000 + 5 \times 10^{-4}((x - 1000)^2 + (z - 1000)^2)$. The source is located at $(x = 1000 \text{ m}, z = 1000 \text{ m})$. The source function is a Ricker wavelet with peak frequency of 32 Hz. Cross-sections of the wavefield at $z = 1000 \text{ m}$ are shown for (c) finite difference method and (d) spectral method. Prominent numerical dispersion exists in the wavefield modeled using finite difference method, the spectral method modeling is free from such dispersion noise.

III.3 Low-rank Decomposition

Despite its high accuracy, extrapolating the wave equation solution using Equation (III.21) is computationally expensive. This is because the phase function $\phi(\mathbf{x}, \mathbf{k})$ is a dense matrix with dimensions N_x -by- N_k , where N_x is the size of space grid and N_k is the size of the wavenumber grid. Consider Fourier transform without padding, $N_x = N_k = N$. Because the integral has spatial dependency $v(\mathbf{x})$, Equation (III.21) can not be implemented as Fast Fourier Transform. Thus, extrapolating one time step using Equation (III.21) costs $\mathcal{O}(N^2)$. In contrast, the computation cost of classical FD is only $\mathcal{O}(N)$. This renders the spectral extrapolation nearly impossible for practical problems. Straightforward implementation of Equation (III.21) is only possible for small-size models.

The time stepping formula Equation (III.17) belongs to the group of so called Fourier Integral Operator (FIO). In general, FIO take the form

$$\Lambda\{f(\mathbf{x})\} = \int_{\mathbf{k}} \hat{f}(\mathbf{k}) a(\mathbf{x}, \mathbf{k}) e^{i\psi(\mathbf{x}, \mathbf{k})} d\mathbf{k} \quad (\text{III.26})$$

where $f(\mathbf{x})$ is any space function, $\hat{f}(\mathbf{k})$ is its Fourier transform defined by (III.6). $\psi(\mathbf{x}, \mathbf{k})$ is the *phase function* and $a(\mathbf{x}, \mathbf{k})$ is the *amplitude function*. The FIO shares a similar form with Fourier transform defined in (III.6) and (III.7), it returns to $f(\mathbf{x})$ when

$$a(\mathbf{x}, \mathbf{k}) = \frac{1}{(2\pi)^d} \quad \text{and} \quad \psi(\mathbf{x}, \mathbf{k}) = \mathbf{x} \cdot \mathbf{k}$$

The spectral time-extrapolation Equation (III.17) is an FIO with

$$a(\mathbf{x}, \mathbf{k}) = \frac{1}{(2\pi)^d} \quad \text{and} \quad \psi(\mathbf{x}, \mathbf{k}) = \mathbf{x} \cdot \mathbf{k} + v(\mathbf{x})|\mathbf{k}|\Delta t \quad (\text{III.27})$$

The summation in Equation (III.26) costs N^2 operations. Candes et al. (2007)

developed a fast algorithm to compute two-dimensional discrete FIO with cost of $TLN \log N$, where $TL \ll N$. T and L are the ranks that approximate the full operator $\mathcal{L}$. The key idea is to partition the wavenumber domain into several angular wedges W_m

$$W_m = \{\mathbf{k} : \frac{2m-1}{M}\pi \leq \arg(\mathbf{k}) < \frac{2m+1}{M}\pi\} \quad (0 \leq m < M) \quad (\text{III.28})$$

where $\arg(\mathbf{k})$ is the angle between $\mathbf{k}$ and vector $[1 \ 0]^T$. Let $\chi_m(\mathbf{k})$ be the indicator function for each wedge W_m

$$\chi_m(\mathbf{k}) = \begin{cases} 1 & \mathbf{k} \in W_m \\ 0 & \mathbf{k} \notin W_m \end{cases}$$

Thus defined, the discrete FIO can be decomposed as a summation of FIO within each wedge (Candes et al., 2007)

$$\Lambda\{f\} = \sum_{m=0}^{M-1} \Lambda_m\{f\} \quad (\text{III.29})$$

$$\Lambda_m\{f(\mathbf{x})\} = \int_{\mathbf{k}} \hat{f}(\mathbf{k}) a(\mathbf{x}, \mathbf{k}) e^{i\psi(\mathbf{x}, \mathbf{k})} \chi_m(\mathbf{k}) d\mathbf{k} \quad (\text{III.30})$$

Let ξ_m be the unit vector pointing from the origin to the center of wedge W_m

$$\xi_m = \left[\cos \frac{2m\pi}{M} \quad \sin \frac{2m\pi}{M} \right]^T.$$

A Taylor expansion of the phase function $\psi(\mathbf{x}, \mathbf{k})$ with respect to point $\xi_m|\mathbf{k}|$ to first order accuracy shows (Candes et al., 2007)

$$\psi(\mathbf{x}, \mathbf{k}) = \psi_m(\mathbf{x}, \mathbf{k}) + \nabla_{\mathbf{k}} \psi(\mathbf{x}, \xi_m) \cdot \mathbf{k} \quad (\text{III.31})$$

where $\psi_m(\mathbf{x}, \mathbf{k})$ is called *residual phase*. The following factorization exists for the product of amplitude function and residual phase (Candes et al., 2007)

$$a(\mathbf{x}, \mathbf{k})e^{\iota\psi_m(\mathbf{x}, \mathbf{k})} = \sum_{n=0}^{\infty} \gamma_{mn}^{(x)}(\mathbf{x})\gamma_{mn}^{(k)}(\mathbf{k}) \quad (\text{III.32})$$

where $\gamma^{(x)}$ is a function of space only, and $\gamma^{(k)}$ is a function of wavenumber only. These factorization functions can be computed by either a deterministic approach or a randomized approach (Candes et al., 2007). The present study uses the randomized approach. The FIO within each wedge is found by plugging Equation (III.31) and Equation (III.32) into Equation (III.30)

$$\Lambda_m\{f(\mathbf{x})\} = \frac{1}{(2\pi)^d} \sum_{n=0}^{\infty} \gamma_{mn}^{(x)}(\mathbf{x}) \int_{\mathbf{k}} \hat{f}(\mathbf{k}) e^{\iota\nabla_{\mathbf{k}}\psi(\mathbf{x}, \xi_m) \cdot \mathbf{k}} \gamma_{mn}^{(k)}(\mathbf{k}) \chi_m(\mathbf{k}) d\mathbf{k} \quad (\text{III.33})$$

Since the spectral time-extrapolation is an FIO with a particular set of amplitude function and phase function (III.27), it is possible to decompose Equation (III.17) following the same way. This is found by substituting Equation (III.27) into Equation (III.33)

$$\begin{aligned} P(\mathbf{x}, t + \Delta t) &= \frac{1}{(2\pi)^d} \sum_{m=0}^{M-1} \sum_{n=0}^{\infty} \gamma_{mn}^{(x)}(\mathbf{x}) e^{\iota v(\mathbf{x}) \Delta t \xi_m} \int_{\mathbf{k}} \hat{P}(\mathbf{k}, t) e^{\iota \mathbf{k} \cdot \mathbf{x}} \gamma_{mn}^{(k)}(\mathbf{k}) \chi_m(\mathbf{k}) d\mathbf{k} \\ &= \sum_{m=0}^{M-1} \sum_{n=0}^{\infty} \gamma_{mn}^{(x)}(\mathbf{x}) e^{\iota v(\mathbf{x}) \Delta t \xi_m} \mathcal{F}^{-1} \{ \hat{P}(\mathbf{k}, t) \gamma_{mn}^{(k)}(\mathbf{k}) \chi_m(\mathbf{k}) \} \end{aligned} \quad (\text{III.34})$$

If the summation of n is truncated, the above equation forms the following low-rank approximation to the exact spectral extrapolation (III.17).

$$P(\mathbf{x}, t + \Delta t) = \sum_{m=0}^{M-1} \sum_{n=0}^{N-1} A_{mn}(\mathbf{x}) \mathcal{F}^{-1} \{ \hat{P}(\mathbf{k}, t) B_{mn}(\mathbf{k}) \} \quad (\text{III.35})$$

where

$$A_{mn}(\mathbf{x}) = \gamma_{mn}^{(x)}(\mathbf{x})e^{\iota v(\mathbf{x})\Delta t \xi_m} \quad \text{and} \quad B_{mn}(\mathbf{k}) = \gamma_{mn}^{(k)}(\mathbf{k})\chi_m(\mathbf{k}) \quad (\text{III.36})$$

Based on Equation (III.35), a practical algorithm to extrapolate wavefield using low-rank approximation can be summarized as follows

- 1 Choose rank M and N . Form factorization (III.32) by using either deterministic or randomized approach. Compute $A_{mn}(\mathbf{x})$ and $B_{mn}(\mathbf{k})$ according to (III.36).
- 2 Take spatial Fourier transform of current solution $P(\mathbf{x}, t) \rightarrow \hat{P}(\mathbf{k}, t)$.
- 3 Multiply with $B_{mn}(\mathbf{k})$
- 4 Inverse Fourier transform the result in Step 3 and multiply with $A_{mn}(\mathbf{x})$
- 5 Repeat Steps 3 and 4 for each $0 \leq m < M$ and $0 \leq n < N$ and sum the results to form the future solution $P(\mathbf{x}, t + \Delta t)$

Step 1 needs to be computed only once, before starting the time-extrapolation. Steps 2 to 5 are repeated for each time sample of the solution. Candes et al. (2007) proposed to choose rank M to be the closest integer to $\sqrt{N_{FFT}}$ where N_{FFT} is the size of discrete Fourier transform to compute $\hat{P}(\mathbf{k}, t)$. This is because by doing so, each wavenumber wedge W_m features the so-called *parabolic scaling*: $length \approx width^2$. It is favored because the resulting stability analysis of this method is independent of the size of the model. Rank N is bounded by the ϵ -separation rank of mixed-domain operator $\exp(\mathbf{x} \cdot \mathbf{k} + v(\mathbf{x})|\mathbf{K}|\Delta t)$ (Candes et al., 2007). For wave equation operators defined in Equation (III.31), the ϵ -rank is computed by

$$r_\epsilon = 1 + \max\{2eA, \log_2(2/\epsilon)\}$$

where A is the bound on space $|\mathbf{x}| < A$, ϵ is the expected error of truncating the

right-hand side of Equation (III.32). Rank N is then chosen to be the closest integer to r_ϵ . Smaller ranks M and N results in more efficient computation. This could be seen from the right-hand side of extrapolation Equation (III.35). The total cost of summation in this equation is proportional to MN . However, the accuracy of the spectral extrapolation decays with decreasing ranks M and N since they are essentially truncations to Taylor series expansion of $\psi(\mathbf{x}, \mathbf{k})$.

CHAPTER IV

Seismic Imaging Examples

IV.1 Overview of Wave-equation Depth Migration

Imaging is the final stage of the seismic processing flow. It attempts to produce an image of the subsurface structure. Seismic imaging can be “time operation” such as Pre-stack Time Migration (PreSTM) or “depth operation” such as Pre-stack Depth Migration (PreSDM) (See section II.1). Today’s seismic imaging is predominantly PreSDM because the lateral homogeneity assumption of time migration is not suitable for most complex geological settings. Depth migration methods are generally classified as “ray-based” imaging or “wave-equation based”. Ray-based methods solve for the traveltimes of propagating wavefronts, instead of the whole wavefield. The subsurface image is then constructed by summing the data along a computed traveltimes curve: this process is often referred to as *Kirchhoff integration*. Ray-based methods are widely used because of a reasonably good accuracy and efficiency to compute. Wave-equation based methods, in contrast, solve the elastic wavefield and construct subsurface images by applying *imaging condition* to the computed wavefields. Because rays are the *asymptotic* approximations to wavefields, it is expected that wave-equation based methods are more accurate than ray-based methods. The popularity of wave-equation based migrations are always limited by their excessive computation cost. Many attempts have been made to reduce the computation costs of wave-equation migrations. For example, *one-way* wave-equation migration has been a successful adaptation of the full *two-way* wave equation with reduced computation cost. In one-way migrations, the wavefield at a certain depth is computed

from the same wavefield at shallower depth, or *extrapolated* from shallow wavefield. The accuracy of one-way methods are in general limited by angles from vertical ray-path. Two-way wave equation, for example Equation (III.1) in acoustic media, is in principle the most accurate imaging method because it solves the complete wavefield, including reflection, diffraction, refraction, multiple arrivals and evanescent waves. Reverse-time migration (RTM) is a two-way wave-equation based depth migration. Reverse-time Migration (RTM) is notable for its ability to image reflectors with arbitrary dipping angles, strong lateral velocity variations and it is straightforward to implement. With the advance of computing power, RTM is becoming the predominant depth migration method.

The seismic wavefield is a function of both space and time, while the subsurface image is a function of space only. In constructing an image, the time-dependency is removed by applying the well-known imaging principle: *“reflectors exist at points in the ground where the first arrival of the downgoing wave is time coincide with an upgoing wave”* (Claerbout, 1971). The wavefield is artificially distinguished as “downgoing” and “upgoing” waves. The downgoing wave is the disturbance generated by the source and propagates into the subsurface. The upgoing wave refers to the energy reflected traveling upwards towards the surface. In Figure IV.1, the downgoing wave is ignited from the source at $t = 0$ and propagates to a reflector A at time $t = 0.6$ sec. At this time snapshot, the value at A becomes the source for the upgoing wave. Because the velocity is faster in the upgoing raypath, it takes 0.4 sec to reach the surface. Figure IV.2 illustrates the concept of imaging condition. Suppose a receiver is buried at point A , the downgoing wave at this point can be found by extrapolating from the source S . Similarly, the upgoing wave is a result of backward-extrapolation from receiver R . By plotting the two wavefields at point A , as in the left figure, clearly the ratio of the two waves at $t = 0.6$ sec is related to the reflection coefficient at A . However, for subsurface imaging purposes the variation of reflection

Figure IV.1: Prestack imaging condition. The downgoing wavefield (Red) is reconstructed by extrapolation from the source (S). The upgoing wavefield (Blue) is reconstructed by backward-extrapolation from the receiver (R).

coefficient on the incident angle is ignored. As a result, the ratio of upgoing wave and downgoing wave is considered as an “approximate” value of the local reflection coefficient. That is

$$r(\mathbf{x}) = \frac{P_U(\mathbf{x}, t)}{P_D(\mathbf{x}, t)} \quad \forall t \quad (P_D \neq 0). \quad (\text{IV.1})$$

A common practice to avoid division by zero is to multiply the denominator by its complex conjugate. If this is further integrated over all time samples, then the reflection coefficient is evaluated by

$$r(\mathbf{x}) = \int \frac{P_U(\mathbf{x}, t) P_D^*(\mathbf{x}, t)}{|P_D(\mathbf{x}, t)|^2 + \epsilon} dt, \quad (\text{IV.2})$$

ϵ is a small number to avoid division by zero. When amplitude of the reflectivity image is not concerned, the whole denominator can be dropped. This yields the well-known *zero-lag cross-correlation* imaging condition

$$r(\mathbf{x}) = \int P_U(\mathbf{x}, t) P_D^*(\mathbf{x}, t) dt \quad (\text{IV.3})$$

Figure IV.2: Prestack imaging condition. Left: The cross-correlation at reflector point A is a high value because both upgoing and downgoing waves arrive at $t = 0.6$ sec. Right: The cross-correlation at non-reflector position B vanishes because different arrival times of upgoing ($t = 0.5$ sec) and downgoing ($t = 0.7$ sec) waves.

The name cross-correlation is due to the fact that the temporal Fourier transform of Equation (IV.3) corresponds to cross-correlation in frequency domain. Similarly, because division is deconvolution in Fourier domain, Equation (IV.2) is referred to as *deconvolution* imaging condition.

IV.2 Zero-offset Modeling and Migration Example

Using the pseudo-depth acoustic wave equation as a tool for doing forward and backward extrapolation, it provides an alternative to the conventional Zero-offset migration (ZOM). It is interesting to see whether ZOM in the pseudo-depth domain is as accurate as it is the conventional space domain, as well as comparing the computational efficiency of migration in the two domains.

To investigate the possibility of doing ZOM in pseudo-depth domain, first we need a zero-offset section. Figure IV.3(a) shows the subsurface velocity model used in this experiment. This model has two velocity layers separated by a reflector featuring flat, dipping and anticline structures. Above the reflector is a slow medium with velocity increasing from 1500 m/sec to 3500 m/sec. Beneath the reflector is a fast medium with 4500 m/sec velocity. Using the exploding reflector modeling methods,

sources are located on each point along the reflector. The strength of each source is proportional to the local reflectivity. Acoustic waves are induced by shooting all the sources simultaneously. Snapshots of the wavefield are plotted in Figure IV.3(b) to Figure IV.3(d). Receivers are placed along the top boundary of the model. Thus

Figure IV.3: (a) A subsurface velocity model featuring two layers. The velocity of the top increases linearly with depth $v(z) = (1500 + z)$ m/sec. The bottom layer is a faster media with $v = 4500$ m/sec. Using exploding reflector method, the modeled wavefield is plotted in (b) to (d). The front surface of each of these cubes shows a time snapshot of the wavefield, at (b) $t = 0$ sec (c) $t = 0.4$ sec and (d) $t = 0.8$ sec.

the synthetic zero-offset section is simply the top surface of the modeled wavefield, this is plotted in Figure IV.4. The original reflectors are apparently mispositioned into “bow-tie” shape in the zero-offset section. The cause of the mispositioning is explained on page 20.

Figure IV.4: The modeling result of Figure IV.3. The zero-offset section is extracted from the top surface $z = 0$ of the cubes shown in Figure IV.3.

The zero-offset modeling is now completed. From now on, we think of the migration problem. This means that the only available information is the zero-offset section. We have no knowledge about the true subsurface velocity as plotted in Figure IV.3(a). Assume Figure IV.4 is a zero-offset data acquired from field experiment. This data, since it is synthetic, is already pre-processed to remove noise and multiples (see Section I.1). The next step is velocity estimation, which in general produces a smoothed subsurface velocity model. For the purpose of validating the accuracy of migration algorithms, the velocity model used here for migration, as plotted in Figure IV.5(a), is simply a smoothed model of true subsurface velocity in Figure IV.3(a). With the zero-offset section and a velocity model available, we can continue to do ZOM. As explained in Section II.1, migrating zero-offset data is essentially back-propagation of zero-offset section in time. Snapshots of the backward extrapolated wavefield are plotted in Figure IV.5(c) and Figure IV.5(e). The final migration image is simply $t = 0$ section of the backpropagated wavefield. Figure IV.6(a) shows the space domain ZOM image.

To do the same migration in pseudo-depth domain, we first need a velocity model in the pseudo-depth domain. This can be done by applying the domain transformation as described in Section II.1. Figure IV.5(b) shows the transformed velocity model. As expected, the fast deep layer is “compressed” in the pseudo-depth domain because of a shorter vertical time. Following the same strategy of backpropagating the same zero-offset section in pseudo-depth domain, we are able to compute a reflectivity image in the pseudo-depth domain. This image, when mapped back to space domain, is plotted in Figure IV.6(b). The backpropagation in pseudo-depth domain is computed by solving pseudo-depth domain wave equation II.10, instead of the conventional acoustic wave equation. Notice no transformation is needed for the zero-offset section in Figure IV.4, since it is not a function of depth. The similarity of the space domain migration image (Figure IV.6(a)), the pseudo-depth domain migration image (Figure IV.6(b)) and the true subsurface reflectivity (Figure IV.6(c)) confirms that ZOM in the pseudo-depth domain is as accurate as conventional space domain migration. The vertical axis of the space domain model has 321 samples, whereas in the pseudo-depth domain the vertical axis has only 257 samples. Ideally, sampling the vertical axis with 20% less number of samples would reduce the computation cost by around 20% as well. However, in this computation example, because of the unequal patching of computation grid in the two domains (see Section III.2), the vertical axis of the pseudo-depth domain computation grid is actually 514 samples. The computation cost is actually increased in pseudo-depth domain in the present study, due to a numerical trick to avoid the artificially imposed periodic boundary condition by using spectral method. This is explained in Section V.2. In general, since the sampling of the vertical axis is reduced in pseudo-depth domain, a corresponding reduction of computation cost should be expected.

Figure IV.5: Zero-offset migration. (a) Migration velocity in space domain. (b) Transformation of (a) to pseudo-depth domain. (c) and (e) are snapshots of the space domain backward extrapolated wavefield with front surface being a snapshot in time, at (c) $t = 1.2$ sec and (e) $t = 0.4$ sec. (b) and (d) are the pseudo-depth domain backward extrapolated wavefield showing snapshots at (d) $t = 1.2$ sec and (f) $t = 0.4$ sec.

Figure IV.6: Zero-offset migration image in (a) space domain and (b) pseudo-depth domain. (c) True subsurface reflectivity model.

CHAPTER V

Discussions

V.1 Conclusions

To summarize, a new space domain called pseudo-depth space is established by transforming the vertical axis from depth to time. By calculating the Jacobian of this coordinate transformation, I derived an acoustic wave equation in the pseudo-depth space. To actually solve this equation in a computer, a numerical scheme is devised based on the technique of low-rank decomposition of Fourier Integral Operator. As a simple application of the pseudo-depth space wave equation, zero-offset modeling and migration has been successfully implemented by doing time-extrapolation using this new wave equation. The migration images are in good agreement with conventional zero-offset migration results.

The pseudo-depth space wavefield differs from conventional acoustic wave in that the reflection coefficients are in general smaller, with the extreme of zero reflection for normal incidence on flat reflectors. This is due to the anisotropic form of the pseudo-depth wave equation. Such low reflection may provide better image quality for depth migration in media with strong impedance contrast, for example the top and flanks of a salt dome. The pseudo-depth wavefield modeling and migration is also promising because it is more cost-effective than conventional regularly sampled space. This is due to the reduced amount of over-sampling at deep fast media as a result of transforming the vertical axis from depth to pseudo-depth.

V.2 Future Directions

Despite its high accuracy, doing wavefield extrapolation using the low-rank approximation in practice still faces several challenges. Firstly, the low-rank approximation using pseudo-depth wave equation as a kernel function that can be unstable when the velocity features strong lateral inhomogeneity. The exact extrapolation operator for a midpoint time stepping is bounded by 2 because of the $2\cos$ term in Equation (III.21). The low-rank approximation decomposes this matrix and its values are no longer guaranteed to be bounded by 2. In the present study, the time step was kept sufficiently small to ensure stability, however the increased computational cost renders this method not yet practical. The solution to this issue is to study the computation in the preprocessing step and design computations that guarantees stability. Secondly, non-reflecting boundary conditions are usually incorporated in seismic modelings to attenuate unwanted reflections at boundaries of the computation domain, for example the paraxial approximation of one-way wave equation (Clayton and Engquist, 1980) and perfectly matched layers (Berenger, 1994). However, most of these boundary conditions are established in space domain. Effective non-reflecting boundary conditions in the wavenumber domain are yet to be studied. Thirdly, Fourier transform automatically assume periodic boundary conditions. As a result, the waves that leave one boundary will be continued on the opposite boundary as if the two-dimensional domain is “wrapped” on a cylinder. A simple trick to tackle this issue is to expand the computation domain by a thin layer with zero velocity. This method works fine with conventional wave equation. With the pseudo-depth domain wave equation, however, because its anisotropy imposes a constant velocity v_0 in the vertical direction, the zero velocity layering is not effective in the vertical direction. As a result, the top/bottom layer needs to be of the same size of the original model. By doing this, the vertical wrapping is resolved, however it also doubles the computation cost of extrapolating

wavefield in pseudo-depth domain. Alternative approaches to eliminate periodicity is needed to avoid wrapping without increasing the computational cost.

Future work is needed to resolve these issues in order to extrapolate wavefield in the pseudo-depth domain with high accuracy and low computational cost.

The pseudo-depth domain wavefield extrapolation has potential applications in full-wave equation based migration and velocity estimation (Alkhalifah, 2003).

APPENDICES

APPENDIX A

Source Codes

Most of the computations in this thesis are programmed by using Madagascar. Madagascar is an open source geophysical computation software that facilitates *reproducible* computational research (Madagascar wiki). Peer review is the backbone of scientific progress. New advancement of scientific research is based on reproducing and verifying previously published research. *The success and credibility of science are anchored in the willingness of scientists to expose their ideas and results to independent testing and replication by others* (APS Ethics and Values). Using Madagascar, a computation work is divided into two parts: a *computation program* that implements the numerical method of the computation, and an *SConstruct* file that manages data flows of the computation. To improve computing efficiency, the computation program is usually coded up by using system languages such as C and Fortran. The SConstruct file is a Python script that links up all the data files and manages input/output for each computation program.

For example, the computation of spectral time-extrapolation is implemented in a C program. A part of this code that does time-stepping is listed in Listing A.1

Listing A.1: Computation program example: spectral time-extrapolation

```

168  /* MAIN LOOP */
    if (verb) sf_warning("nt=%d",nt);
    for (it=0; it<nt; it++) {
    if(verb) sf_warning("it=%d;",it);

172  /* inject source */
    sf_floatread(wv,ns,Fi);
    lint2d_inject(curr,ww,lints);

176  /* matrix multiplication */
    fft2(curr[0],cwave);

```

```

180     for (im = 0; im < m2; im++) {
        for (ik = 0; ik < nk; ik++) {
#ifdef SF_HAS_COMPLEX_H
            cwavem[ik] = cwave[ik]*rr[ik][im];
#else
184         cwavem[ik] = sf_crmul(cwave[ik],rr[ik][im]);
#endif
        }
        ifft2(wave[im],cwavem);
188     }

    for (ix = 0; ix < nx; ix++) {
        for (iz=0; iz < nz; iz++) {
192             i = iz+ix*nz; /* original grid */
            j = iz+ix*nz2; /* padded grid */

            old = curr[ix][iz];
196             for (c = 0, im = 0; im < m2; im++) {
                c += lm[im][i]*wave[im][j];
            }
            curr[ix][iz] = c - prev[ix][iz];
200             prev[ix][iz] = old;
        }

        /* write wavefield */
204         if (!(it % jsnap))
            sf_floatwrite(curr[ix],nz,Fu);

        /* extract data */
208         lint2d_extract(curr,dd,lintr);
    }

    /* write data */
212     sf_floatwrite(dd,nr,Fo);
}

```

The Python script in Listing A.2 defines functions to perform pseudo-depth mapping. The mapping operation is divided into two small programs: the C program `integral1` in line 46 does the integration, another C program `pseudodepth` in line 65 does the interpolation. Such division follows the philosophy of UNIX programming: *write programs that do one thing and do it well*.

Listing A.2: Python interfaces

```

40 def depth(z,v,v0=1500,inv=False):
    """
        v = input velocity
        z = output depth (physical depth if inv==True; pseudo if inv==
            False)
        """
44     if inv:

```

```

        Flow(z,v, '''
            integral1 | math output="input*%g"
            ''' % (1.0/v0))
48     else:
        Flow(z,v, '''
            math output="1.0/input" |
            integral1 | math output="input*%g"
52         ''' % v0)

def mapping(out,inp,depth,n,o,d,inv=False):
    """ pseudo-depth mapping
56     inv = False, depth to pseudodepth
    inv = True,  pseudodepth to depth
    depth = output depth
    """
60     if inv:
        hdr = ' label1=z unit1=m'
    else:
        hdr = ' label1=z* unit1=m'
64     Flow(out,[inp,depth], '''
        pseudodepth
        n=%d o=%g d=%g
        depth=${SOURCES[1]} |
68         put %s
        ''' % (n,o,d,hdr))

```

The entire computation is managed by an SConstruct script. Listing A.3 is the full SConstruct script that implements acoustic RTM in the space domain and pseudo-depth domain.

Listing A.3: SConstruct file for two RTMs

```

from rsf.proj import *
import wave
3
# Depth
DATA = 'french.asc'
Fetch(DATA,'data/french',top=os.environ['SWAGM'],server='local')
7 Flow('french',DATA,'dd form=native | transp | scale rscale=2')
Result('french', '''
    grey color=j allpos=y bias=1000 title="French model"
    scalebar=n barlabel="Depth" barunit="m"
11    labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
    =1
    ''')

# interval velocity
15 nx = 321
dx = 25.6625
nz = 321
dz = 12.83125
19 Flow('vi','french', '''
    window n1=1 f1=30 |
    remap1 n1=%d o1=0 d1=%g |

```

```

unif2 n1=%d d1=%g v00=1500,4000 dvdz=1,0 |
23 put label1=z unit1=m label2=x unit2=m
''' % (nx,dx,nz,dz))
Result('vi','''put d1=%g unit1=km d2=%g unit2=km | scale rscale=.001
|
grey color=j allpos=y bias=1.500 title="Interval velocity"
27 scalebar=n barlabel="\F2 Velocity" barunit="km/sec" barreverse=y
labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
=.5
''' % (.001*dz,.001*dx))

31 # migration velocity
Flow('v','vi','smooth rect1=9 rect2=5 repeat=2')
Result('v','''put d1=%g unit1=km d2=%g unit2=km | scale rscale=.001
|
grey color=j allpos=y bias=1.500 title="Migration velocity"
35 scalebar=n barlabel="\F2 Velocity" barunit="km/sec" barreverse=y
labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
=.5
''' % (.001*dz,.001*dx))

# one-way vertical time
39 Flow('tau','v','math output="1.0/input" | integral1')
Result('tau','''put d1=%g unit1=km d2=%g unit2=km |
grey color=j allpos=y title="\F10 t"
scalebar=y barlabel="\F2 Time" barunit="sec" barreverse=y
43 labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
=.5
''' % (.001*dz,.001*dx))

# new axis
v0 = 1500
47 taumax = 1.38
nzz = 257
dzz = taumax/(nzz-1.0)*v0

# new migration velocity
51 wave.depth('zz','v',v0,False)
wave.mapping('nv','v','zz',nzz,0,dzz,False)
Result('nv','''put d1=%g unit1=km d2=%g unit2=km | scale rscale
=0.001 |
grey color=j allpos=y bias=1.500 title="Migration velocity"
55 scalebar=n barlabel="\F2 Velocity" barunit="km/sec" barreverse=y
labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
=.5
''' % (.001*dzz,.001*dx))

# sigma
59 Flow('sig','tau','''
transp | igrad | math output="input*%g" | transp |
smooth rect1=5 rect2=5 repeat=2
''' % (1.0/dx))

63 wave.mapping('nsig','sig','zz',nzz,0,dzz,False)
Result('nsig','''put d1=%g unit1=km d2=%g unit2=km |
grey color=j title="\F10 s"
scalebar=y barlabel= barunit=
67 labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
=.5
''' % (.001*dzz,.001*dx))

# reflectivity

```

```

71 Flow('ref','vi','''
    igrad | bandpass flo=0.004 fhi=0.008 |
    transp | bandpass fhi=0.016 | transp | scale axis=123
    ''')
75 Result('ref','''put d1=%g unit1=km d2=%g unit2=km |
    grey title="Reflectivity"
    labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
    =.5
    '' % (.001*dz,.001*dx))
79
# expand stratigraphic velocity
nbx = 39
nbz = 39
83 Flow('bz',None,'spike nsp=1 mag=0 n1=%d n2=%d | put o1=0 d1=%g o2=0
    d2=%g' % (nbz,nx,dz,dx))
    Flow('bx',None,'spike nsp=1 mag=0 n1=%d n2=%d | put o1=0 d1=%g o2=0
    d2=%g' % (nz+nbz,nbx,dz,dx))
    Flow('vi2','vi bz bx','put o1=0 | cat axis=1 ${SOURCES[1]} | cat
    axis=2 ${SOURCES[2]} | put o1=0 label1=z unit1=m label2=x unit2=m
    ')
# expand migration velocity
87 Flow('v2','v bz bx','put o1=0 | cat axis=1 ${SOURCES[1]} | cat axis
    =2 ${SOURCES[2]} | put o1=0 label1=z unit1=m label2=x unit2=m')
# expand new migration velocity
nbzz= nzz
Flow('nbz',None,'spike nsp=1 mag=0 n1=%d n2=%d | put o1=0 d1=%g o2=0
    d2=%g' % (nbzz,nx,dzz,dx))
91 Flow('nbx',None,'spike nsp=1 mag=0 n1=%d n2=%d | put o1=0 d1=%g o2=0
    d2=%g' % (nzz+nbzz,nbx,dzz,dx))
    Flow('nv2','nv nbz nbx','put o1=0 | cat axis=1 ${SOURCES[1]} | cat
    axis=2 ${SOURCES[2]} | put o1=0 label1=z* unit1=m label2=x unit2=
    m')
# expanded sigma
Flow('nsig2','nsig nbz nbx','put o1=0 | cat axis=1 ${SOURCES[1]} |
    cat axis=2 ${SOURCES[2]} | put o1=0 label1=z* unit1=m label2=x
    unit2=m')
95
# fft
Flow('fft','v2','''
    rtoc | fft3 axis=1 pad=1 | fft3 axis=2 pad=1 |
99    put label1="kz" label2="kx"
    ''')
Flow('nfft','nv2','''
    rtoc | fft3 axis=1 pad=1 | fft3 axis=2 pad=1 |
103    put label1="kz*" label2="kx"
    ''')

# propagator matrices
107 nt = 2750
    dt = 0.001
# interval velocity
Flow('mi li ri','vi2 fft','''
111    isolr2one seed=2010 dt=%g
    fft=${SOURCES[1]}
    left=${TARGETS[1]}
    right=${TARGETS[2]}
115    '' % dt)

```

```

# migration velocity
Flow('m l r','v2 fft',''
    isolr2one seed=2010 dt=%g
119     fft=${SOURCES[1]}
        left=${TARGETS[1]}
        right=${TARGETS[2]}
        '' % dt)
123 Flow('nm nl nr','nv2 nfft nsig2',''
    isolr2tau seed=2010 dt=%g
        v0=%g
        fft=${SOURCES[1]}
127     sig=${SOURCES[2]}
        left=${TARGETS[1]}
        right=${TARGETS[2]}
        '' % (dt,v0))
131 # stabilize
Flow('lis ibeta','mi li ri',''
    fftstable cmplx=y pad1=1
        nx=%d nz=%d M=1.999999
135     left=${SOURCES[1]}
        right=${SOURCES[2]}
        b=${TARGETS[1]}
        '' % (nx+nbx,nz+nbz))
139 Flow('ls beta','m l r',''
    fftstable cmplx=y pad1=1
        nx=%d nz=%d M=1.9999
        left=${SOURCES[1]}
143     right=${SOURCES[2]}
        b=${TARGETS[1]}
        '' % (nx+nbx,nz+nbz))
Flow('nls nbeta','nm nl nr',''
147     fftstable cmplx=y pad1=1
        nx=%d nz=%d M=1.9999
        left=${SOURCES[1]}
        right=${SOURCES[2]}
151     b=${TARGETS[1]}
        '' % (nx+nbx,nzz+nbzz))

# source coord
155 xs = range(500,7500+140,140)
    zs = 15
    zzs= 15
    nshot = len(xs)
159 print 'nshot=%d' % nshot

# receiver coord
    zr = 15
163 wave.hline('rr',nx,0,dx,zr)
    zzr= 15
    wave.hline('nrr',nx,0,dx,zzr)

167 # source wavelet
    kt = wave.ricker('w_',nt,dt,freq=15)
    Flow('w','w_', 'transp')
    Result('w','transp | window n1=400 | graph grid=y title="total %g
        sec" plotfat=10 plotcol=4 min1=0 label1=t unit1=sec min2=-1 max2
        =1 label2= unit2=' % (nt*dt))

```

```

171 # time extrapolation
def timewave(data,out,inp,left,middle,right,velo,sou,rec,jsnap=5):
    Flow([data,out],[inp,left,middle,right,velo,sou,rec],'''
175         fftwave4
            verb=y cmplx=y pad1=1
            jsnap=%d
            left=${SOURCES[1]}
179         middle=${SOURCES[2]}
            right=${SOURCES[3]}
            vel=${SOURCES[4]}
            sou=${SOURCES[5]}
183         rec=${SOURCES[6]}
            wave=${TARGETS[1]} |
            put label1=x unit1=m
            ''' % (jsnap))

187 # cutting
def cut(out,inp,pseudo=False):
    if pseudo == True:
        Flow(out,inp,'window n1=%d n2=%d | put label1=z* unit1=m
            label2=x unit2=m' % (nzz,nx))
191     else:
        Flow(out,inp,'window n1=%d n2=%d | put label1=z unit1=m
            label2=x unit2=m' % (nz,nx))

# acoustic rtm
195 jsnap = 2
    imgs = []
    nimgs = []
    for ishot in range(nshot):
199         n = '-%02d' % ishot
            # source coord
            wave.point('ss'+n,xs[ishot],zs)
            wave.point('nss'+n,xs[ishot],zxs)
203         # seismogram
            timewave('d'+n,'u2'+n,'w','li','mi','ri','vi2','ss'+n,'rr',jsnap
                )
            Result('d'+n,'''put d1=%g unit1=km |
                transp | halfint inv=y |
207                 grey title=""
                    screenratio=0.75 labelsz=6 titlesz=8
                    ''' % (.001*dx) )
            cut('u'+n,'u2'+n,False)
211         # reverse data
            Flow('d-rev'+n,'d'+n,'reverse which=2 opt=i verb=y')
            # acoustic rtm
            timewave('ds'+n,'us2'+n,'w','ls','m','r','v2','ss'+n,'rr',jsnap)
215         cut('us'+n,'us2'+n,False)
            timewave('dr'+n,'ur2'+n,'d-rev'+n,'ls','m','r','v2','rr','rr',
                jsnap)
            cut('ur'+n,'ur2'+n,False)
            Flow('img'+n,['ur'+n,'us'+n],'reverse which=4 opt=i verb=y |
                xcor2d axis=3 uu=${SOURCES[1]}')
219         imgs.append('img'+n)
            # pseudo-depth rtm
            timewave('nds'+n,'nus2'+n,'w','nls','nm','nr','nv2','nss'+n,'nrr
                ',jsnap)

```

```

223     cut('nus'+n,'nus2'+n,True)
    timewave('ndr'+n,'nur2'+n,'d-rev'+n,'nls','nm','nr','nv2','nrr',
        'nrr',jsnap)
    cut('nur'+n,'nur2'+n,True)
    Flow('nimg'+n,['nur'+n,'nus'+n],'reverse which=4 opt=i verb=y |
        xcor2d axis=3 uu=${SOURCES[1]}')
    nimgs.append('nimg'+n)
227
    # postprocessing
    Flow('stack',imgs,'cat axis=3 ${SOURCES[1:%d]} | stack axis=3' %
        nshot)
    Flow('nstack_',nimgs,'cat axis=3 ${SOURCES[1:%d]} | stack axis=3' %
        nshot)
231 Flow('mut',None,'spike nsp=3 mag=0,1,0 n1=%d n2=%d k1=0,36,290 l1
    =36,290,%d | put o1=0 d1=%g o2=0 d2=%g' % (nz,nx,nz,dz,dx))
    Flow('img','stack mut','''
        bandpass flo=0.006 fhi=0.010 | transp |
        bandpass flo=0      fhi=0.010 | transp |
235     math m=mut.rsfs output="input*m"
        ''')
    Result('img','''put d1=%g unit1=km d2=%g unit2=km |
        grey title="Migration image" pclip=99
239     labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
        =.5
        '' % (.001*dz,.001*dx))
    wave.depth('z','nv',v0,True)
    wave.mapping('nstack','nstack_','z',nz,0,dz,True)
243 Flow('nimg','nstack mut','''
        bandpass flo=0.004 fhi=0.008 | transp |
        bandpass flo=0      fhi=0.008 | transp |
        math m=mut.rsfs output="input*m"
247     ''')
    Result('nimg','''put d1=%g unit1=km d2=%g unit2=km |
        grey title="" pclip=99
        labelsz=8 titlesz=8 labelfat=4 titlefat=4 screenht=8 screenratio
        =.5
251     '' % (.001*dz,.001*dx))

End()

```


APPENDIX B

Conference

Part of this work is communicated in the 73rd European Association of Geoscientists and Engineers Conference & Exhibition, held in Vienna, Austria, in May 23-26, 2011.

Xuxin Ma, Tariq Alkhalifah, “Wavefield extrapolation in pseudo-depth domain”
(Oral presentation)

BIBLIOGRAPHY

- K Aki and PG Richards. *Quantitative Seismology*. W H Freeman and Co, 1980.
- R M Alford, K R Kelly, and D M Boore. Accuracy of finite-difference modeling of the acoustic wave equation. *Geophysics*, 39(6):834–842, 1974.
- T Alkhalifah. τ migration and velocity analysis: Theory and synthetic examples. *Geophysics*, 68(4):1331–1339, 2003.
- T Alkhalifah, S Fomel, and B Biondi. The space-time domain: theory and modelling for anisotropic media. *Geophysical Journal International*, 144(1):105–113, 2001.
- Tariq Alkhalifah. An acoustic wave equation for anisotropic media. *Geophysics*, 65(4):1239–1250, 2000.
- Tariq Alkhalifah. τ -migration and velocity analysis: application to data from the Red Sea. *Geophysical Prospecting*, 53(5):643–653, 2005.
- Tariq Alkhalifah and Sergey Fomel. Source-receiver two-way wave extrapolation for prestack exploding-reflector modeling and migration. *SEG Technical Program Expanded Abstracts*, 29(1):2950–2955, 2010.
- M Bacon, R Simm, and T Redshaw. *3-D Seismic Interpretation*. Cambridge University Press, 2003.
- E Baysal, D D Kosloff, and J W C Sherwood. A two-way nonreflecting wave equation. *Geophysics*, 49(2):132–141, 1984.
- Jean-Pierre Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *Journal of Computational Physics*, 114(2):185 – 200, 1994.
- MJ Berger and J Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of computational Physics*, 53(3):484–512, 1984.
- A J Berkhout and D J Verschuur. Estimation of multiple scattering by iterative inversion, part i: Theoretical considerations. *Geophysics*, 62(5):1586–1595, 1997.
- FJ Billette and S Brandsberg-Dahl. The 2004 BP velocity benchmark. *67th EAGE Conference & Exhibition, Extended Abstracts*, page B035, 2005.
- R J Blakely. *Potential Theory in Gravity and Magnetic Applications*. Cambridge University Press, 1995.
- E Candes, L Demanet, and L Ying. Fast computation of Fourier integral operators. *SIAM Journal on Scientific Computing*, 29(6):2464 – 2493, 2007.
- J F Claerbout. *Fundamentals of Geophysical Data Processing*. Blackwell Scientific Publications, 1985a.

- J F Claerbout. *Imaging the Earth's Interior*. Blackwell Scientific Publications, 1985b.
- Jon F Claerbout. Toward a unified theory of reflector mapping. *Geophysics*, 36(3): 467–481, 1971.
- R W Clayton and B Engquist. Absorbing boundary conditions for wave-equation migration. *Geophysics*, 45(5):895–904, 1980.
- M A Dablain. The application of high-order differencing to the scalar wave equation. *Geophysics*, 51(1):54–66, 1986.
- J T Etgen and S Brandsberg-Dahl. The pseudo-analytical method: Application of pseudo-laplacians to acoustic and acoustic anisotropic wave propagation. *SEG Technical Program Expanded Abstracts*, 28(1):2552–2556, 2009.
- Dan Hampson. Inverse velocity stacking for multiple elimination. *Canadian Journal of Exploration Geophysics*, 22(1):44–55, 1986.
- George V Keller and F C Frischknecht. *Electrical Methods in Geophysical Prospecting*. Pergamon Press, 1966.
- Larry R Lines, Alton K Schultz, and Sven Treitel. Cooperative inversion of geophysical data. *Geophysics*, 53:8–20, 1988.
- William Lowrie. *Fundamentals of Geophysics*. Cambridge University Press, second edition, 2007.
- E Robein. *Seismic Imaging: A Review of the Techniques, their Principles, Merits and Limitations*. EAGE Publications BV, 2010.
- John A Scales. *Theory of Seismic Imaging*. Samizdat Press, 1994.
- Robert E Sheriff and Lloyd P Geldart. *Exploration Seismology*. Cambridge University Press, second edition, 1995.
- M C Sinha, P D Patel, Unsworth M J, Owen T R E, and M G R MacCormack. An active source electromagnetic sounding system for marine use. *Marine Geophysical Research*, 12:59–68, 1990.
- Albert Tarantola. *Inverse Problem Theory and Methods for Model Parameter Estimation*. Society for Industrial and Applied Mathematics, 2005.
- W M Telford, L P Geldart, and R E Sheriff. *Applied Geophysics*. Cambridge University Press, second edition, 1990.
- URL. Ethics and values "what is science?". http://www.aps.org/policy/statements/99_6.cfm, a.
- URL. Madagascar wiki – main page. <http://www.m8r.info>, b.

- B D Wards, G F Margrave, and M P Lamoureux. Phase-shift time-stepping for reverse-time migration. *SEG Technical Program Expanded Abstracts*, 27(1):2262–2266, 2008.
- Yilmaz. *Seismic Data Analysis: Processing, Inversion, and Interpretation of Seismic Data*. Society of Exploration Geophysicists, 2001.
- P D Young and C S Cox. Electromagnetic active source sounding near the east pacific rise. *Geophysical Research Letters*, 8:1043–1046, 1981.