

**Master's Thesis**  
**Department of Civil and Earth Resources Engineering**  
**Graduate School of Engineering**  
**Kyoto University**  
**February 2010**

---



# Virtual world in Geophysics; synthesized data by interferometry and simulation

Engineering Geology  
Department of Civil and Earth Resources Engineering  
Graduate School of Engineering  
Kyoto University

Norimitsu Nakata

## **Abstract**

Importance of virtual world is getting bigger in Geophysics and the core topics are interferometry and numerical experiments. Interferometry recovers impulse response that is virtual shot record between two points that are source or receiver. Interferometry is helpful because it can apply the hard place to deploy active sources. I propose cross-coherence interferometry and obtain useful subsurface information from virtual shot gather made by traffic noise. Numerical simulation is one of three main tools for research now; other tools are mathematics and field data processing. I use simulation for verification of availability of time-lapse seismic survey when storing  $CO_2$  and make reasonable indicator of relationship between receivers and noises. It is, however, difficult for taking much time to simulate three-dimensional full elastic wave propagation. I propose epoch-making method using Graphic Processing Unit and I success to reduce much computation time. These are all leading edge of "Virtual world in Geophysics" and really useful.

# Contents

<b>Acknowledgement</b>	<b>xiv</b>
<b>Introduction</b>	<b>1</b>
<b>Part I Interferometry by cross-coherence</b>	<b>4</b>
<b>Chapter I.1 Introduction</b>	<b>5</b>
<b>Chapter I.2 Theory of Interferometry</b>	<b>7</b>
I.2.1 Cross-correlation and deconvolution .....	8
I.2.2 Cross-coherence .....	9
<b>Chapter I.3 Advantages and Disadvantages of the cross-coherence method</b>	<b>14</b>
I.3.1 Correction of the amplitude variation among traces .....	14
I.3.2 Suppressing the influence of noise .....	16
I.3.3 Removing the amplitude information .....	19
<b>Chapter I.4 Simulation</b>	<b>20</b>
I.4.1 Simulation of noise-free data .....	20
I.4.2 Simulation of noise contaminated data .....	22
<b>Chapter I.5 Case Study</b>	<b>26</b>

---

<b>Chapter I.6</b>	<b>Conclusion</b>	<b>35</b>
<b>Appendix</b>	<b>Higher order terms in the cross-coherence method</b>	<b>37</b>
<b>Part II</b>	<b>Indicators for survey designing for seismic time-lapse monitoring of <math>CO_2</math> injection to subsurface aquifer</b>	<b>43</b>
<b>Chapter II.1</b>	<b>Introduction</b>	<b>44</b>
<b>Chapter II.2</b>	<b>Simulated data processing</b>	<b>46</b>
II.2.1	Ray tracing simulation using synthetic model .....	46
II.2.2	Ray tracing simulation using realistic model .....	55
<b>Chapter II.3</b>	<b>Discussion</b>	<b>77</b>
II.3.1	Importance of migration in ray tracing data .....	77
II.3.2	Indicators of survey design using NRMS and RMS .....	77
II.3.3	Application of seismic interferometry .....	84
<b>Chapter II.4</b>	<b>Conclusion</b>	<b>87</b>
<b>Part III</b>	<b>Three-dimensional elastic wave propagating simulation using FDTD method on GPU</b>	<b>89</b>
<b>Chapter III.1</b>	<b>Introduction</b>	<b>90</b>
<b>Chapter III.2</b>	<b>FDTD computation by staggered grid on GPU</b>	<b>94</b>
III.2.1	Programing for CUDA .....	94
III.2.2	Example of CUDA .....	95
III.2.3	Mounting FDTD on GPU .....	99
<b>Chapter III.3</b>	<b>Discussion</b>	<b>109</b>

<b>Chapter III.4 Conclusion</b>	<b>112</b>
<b>Appendix Staggered grid</b>	<b>114</b>
<b>Conslusion</b>	<b>120</b>
<b>References</b>	<b>121</b>

# List of Figures

I.2.1	The raypath of the term $H_{AB}^5$ . The triangles show two receivers, $r_A$ (gray) and $r_B$ (white). The black square is the scattering point $x_s$ . $r_B$ is a pseudo point source. This wavefield is excited at $r_B$ propagates to $x_s$ as shown by the dotted arrow, and is reflected and returns to $r_B$ as shown by the dashed arrow. Then the wavefield is reflected at $r_B$ toward $r_A$ as shown by the solid arrow. ....	12
I.4.1	Flat horizontal two-layer model for FDTD simulation. Volumes $V_1$ and $V_2$ have velocities $1500m/s$ and $2000m/s$ , respectively. There are two receivers, $r_A$ (gray triangle) at $(x,z) = (1500, 900)$ and $r_B$ (white triangle) at $(3500, 900)$ . The depth of the line of sources at $50m$ increments is $500m$ . The lateral edge positions are $500$ and $4500m$ . The simulated volume is $5000 \times 5000m$ and the thickness of the first layer is $2500m$ .....	21
I.4.2	Interferometry gathers constructed by interferometry using the of (a) cross-correlation, (b) deconvolution, and (c) cross-coherence methods between $r_A$ and $r_B$ in simulated data and the arrival times of the wavefields for (d) cross-correlation, (e) deconvolution, and (f) cross-coherence. The labels "C", "D", and "H" in these figures correspond to equations I.2.5, I.2.8 and I.2.14, respectively. ....	23

I.4.3 The effect of noise added to the simulation data before applying (a) cross-correlation, (b) deconvolution, and (c) cross-coherence interferometry. One trace of these figures is made from the stack to source position of Figure I.4.2. The time range is from 0.9 to 4.0s in Figure I.4.2. In each figure, the signal-to-noise ratio is different between traces. No noise is added to the leftmost trace. The second trace from the left has a signal-to-noise ratio of 20. From this trace, the signal-to-noise ratio is reduce by one for each successive trace. The rightmost trace’s signal-to-noise ratio is 1g ..... 25

I.5.1 (a)The location of survey line. I obtained the traffic noise data at Gunma, Japan. There are some crossing and parallel roads and trains. The survey line is almost straight along the river. (b)Observed gather. I set the receiver number from south to north. .... 27

I.5.2 (a), (c)Observed data in time and frequency domain, respectively. (b), (d)normalized data in time and frequency domain, respectively. (b) shows  $D(\omega)$  and (d) shows  $\frac{D(\omega)}{|D(\omega)|}$ ... 28

I.5.3 Pseudo shot gathers made by interferometry. (a)cross-correlation, (b)deconvolution, and (c)cross-coherence. The source point of these gathers is receiver number 60. I do not apply some filters to these displayed data..... 30

I.5.4 Subsurface structure made by CMP stack method and migration using reflection wave. (a)cross-correlation, (b)deconvolution (c)cross-coherence interferometry and (d) is made by using active source. These figures are applied 3-5-35-40 Hz band pass filter..... 31

I.5.5 (a)pseudo shot gather that are made by cross-coherence interferometry and that source point is receiver number 190. (b)phase velocity, (c)dispersion curve and (c)interval velocity that are made by (a). In (c), the blue  $\times$  is picking point of fundamental mode of surface wave and the red line is the dispersion curve from inversion. In (d), this interval velocity is estimated by inversion. The red thickness line is the estimated velocity and the blue dashed lines are the range of error. .... 33

I.5.6 Short time window FFT section from pseudo shot gathers by (a)cross-correlation, (b)deconvolution, and (c)cross-coherence interferometry. Transverse axis is in time domain, and vertical axis is in frequency domain. These values are normalized by maximum value of these section..... 34

I.5.7 Phase velocity section. I compare max amplitude values between three methods in each frequency range. Black, blue and red points are picked from cross-correlation, deconvolution and cross-coherence sections, respectively. .... 34

I.A.1 (a) $(G_s(r_A, s)G_0^*(r_A, s))^{2a-b-n}$ . This wavefield is excited at  $r_A$  and propagates to  $x_s$  (dashed arrow) and reflects at  $x_s$  and returns to  $r_A$  (dotted arrow). (b) $\{G_s(r_A, s)G_0^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)\}^{m-2c+d}$ . This wavefield is excited at  $r_A$  and propagates to  $r_B$  through  $x_s$  (dashed arrow). And this wavefield reflects at  $r_B$  and returns to  $r_A$  through  $x_s$  again (dashed arrow).  $r_B$  behaves free-point boundary..... 41

II.2.1 (a)Synthetic model for ray tracing. The size of this model is  $10 \times 10 \times 3km$  and this has 9 layers. (b)P wave velocity slice at  $Y = 5km$ . .... 47

II.2.2  $CO_2$  injection portion. This is round shape and size is  $800 \times 50m$ . This portion is put between two layers. .... 47

II.2.3 Sources and Receivers. I show the densest pattern. (a)ground plan of sources and receivers. Red point is source and blue point is receiver. This survey takes orthogonal geometry and I can get a variety of offsets without near. (b)Survey and layers..... 49

II.2.4 The number of folds. The range of this square is  $4 \times 4km$  at the center of the model. .... 49

II.2.5 Shot gather. The shot point is  $(x, y, z) = (5, 5, 0km)$  and I show all receivers gather. Because I simulate only primary wave from each layer, I have no data after about 2.0sec. This shot gather is made by ray tracing, this do not simulate surface wave or diffraction wave. The label of horizontal axis is receiver number and there are 121 receivers by each receiver line. .... 50

II.2.6 CMP gather after NMO correction. Left figure is noise free, middle figure is added scattering noise and right figure is added both scattering and white noises. ....	51
II.2.7 DMO stack section of after high saturated $CO_2$ injection. Left figure is noise free, middle figure is added scattering noise and right figure is added both scattering and white noises.....	51
II.2.8 Migration sections. before, after1 and after2 is before injection, after injection that are high saturated $CO_2$ and after injection that are low saturated $CO_2$ , respectively. (a), (b) and (c) densest survey (Pattern A). (d), (e) and (f) dense sources and sparse receivers (Pattern B). (g), (h) and (i) sparsest survey (Pattern C). ....	52
II.2.9 The RMS amplitude that time gate is 32ms whose center is the top of aquifer. Top, middle and bottom line figures are Pattern A, B and C, respectively. Left, middle and right row figures are before $CO_2$ injection, after high saturated $CO_2$ injection, and after low saturated $CO_2$ injection, respectively. ....	53
II.2.10 NRMS of noise free data. White color shows 200 and black color shows 0. Top and bottom line figures are high and low $CO_2$ saturated. Left, middle and right row figures are Pattern A, B and C. ....	54
II.2.11 NRMS of data that are added scattering and white noise. The arrangement of figure is same as II.2.10.....	55
II.2.12 Migration sections of real 3D seismic data. (a) East-West line. (b) North-South line. The broken line (a) and solid line (b) show the well that observe some properties. ....	56
II.2.13 Synthetic seismogram of logging data. The type of data from left are time and depth index, P wave velocity, density, acoustic impedance, reflection coefficient, 35Hz ricker wavelet, wavelet that are extracted from seismic data around well, synthetic data that are convoluted extracted wavelet, synthetic data of ricker wavelet, seismic trace around well, opposite phase of synthetic data of extracted wavelet, and opposite phase of synthetic data of ricker wavelet.....	57

II.2.14 Depth of grid of one horizon. After picking horizon, I take grids whole range of model. The unit of right colorbar is  $m$ . The grid in the center of figure is real 3D seismic survey range. .... 57

II.2.15 Simulated sections of 2D FDTD method. The models are (a)observed logging data and (b)blocking data from observed data (Table II.2.3), and this is assumed horizontally multi-layered structure. The sources are top of the sea and horizon distance is  $1,000m$ , and receivers are bottom of the sea. .... 59

II.2.16 Identification of wave that is existed in Figure II.2.15(b). No. 1 wave that are attached arrow is multiple between top and bottom of the sea. No. 2, 3 and 4 waves are P wave primary, P-S converted and S wave primary reflection wave from layer 2, reflectively. No. 5 and 6 waves are primary and multiple P reflected wave from layer 4 and No. 7 is the primary P reflected wave from layer 5 that are target horizon in this research. .... 59

II.2.17 Realistic model. The size of this model is  $4.7 \times 4.5 \times 3km$  and this has 11 layers. .... 60

II.2.18 P wave velocity slice. (a)X-Z slice at  $Y = 2.25km$  and (b)Y-Z slice at  $X = 2.35km$ . .... 61

II.2.19 The injection portion of injection 1 that size is  $200 \times 5m$ . This portion is center of X – Y plane and put between two layers. .... 61

II.2.20 The densest pattern of sources and receivers in realistic model. (a)survey with layers. The vertical distance between source and receivers is  $50m$ (b). .... 62

II.2.21 The number of folds. Top figures are survey geometries. Red point is source and blue point is receiver. Bottom figures show the number of folds at all CMP. The size of CMP is  $2.5 \times 2.5km$ . .... 62

II.2.22 CMP gathers. This gathers line keeping the distance of offset. (1)P wave primary only, (2)P wave primary and coherent noise that are multiple, P-S converted and S wave, (3)adding white noise from (2), (4)adding scattering noise from (3), (5)applying NMO correction to (4), (6)applying the handling that are remove the multiples to (5), (7)applying DMO correction to (6). .... 64

II.2.2 DMO stack sections and Migration sections that are sliced at  $CMPY = 90$ . Top and bottom line figures are DMO stack and Migration section, respectively. (a) and (e) are before injection, (b) and (f) are smallest injection area, (c) and (g) are wide but thin injection area, and (d) and (h) are middle width and thickness. .... 65

II.2.2 DMO stack sections. (a) P wave primary only, (b) Primary wave and coherent noise, and (c) Adding white noise that are  $S/N = 0.5$  and scattering noise to (b). The slice position is same as Figure II.2.23. .... 66

II.2.2 DMO section of  $S/N = 4$ . Top and bottom line figures are using exact velocity and 95% velocity, respectively. The receiver geometry of (a) and (d) are Pattern A, (b) and (e) are Pattern B, and (c) and (f) are Pattern C. The slice position is same as Figure II.2.23. .... 67

II.2.2 DMO section of  $S/N = 2$ . The geometry and slice position of figures are same as Figure II.2.25. .... 68

II.2.2 DMO section of  $S/N = 0.5$ . The geometry and slice position of figures are same as Figure II.2.25. .... 69

II.2.2 NRMS at of DMO stack. The time gate of NRMS is  $50ms$  and the center of this gate is the top of aquifer. Top and bottom line figures are P wave primary only and adding coherent noise, respectively. The size of injection is (a) and (d) are smallest (injection 1), (b) and (e) are wide but thin (injection 2), and (c) and (f) are middle width and thickness (injection 3). .... 70

II.2.2 NRMS and RMS of receiver geometry Pattern A. Top and bottom line figures are NRMS and RMS, respectively. The signal to noise ratio of (a) and (d) are 4, (b) and (e) are 2, and (c) and (f) are 0.5. .... 71

II.2.3 NRMS and RMS of receiver geometry Pattern B. The figure geometries are same as Figure II.2.29. .... 72

II.2.3 NRMS and RMS of receiver geometry Pattern C. The figure geometries are same as Figure II.2.29. ....	73
II.2.3 NRMS and RMS of receiver geometry Pattern A and using 95% velocity. The figure geometries are same as Figure II.2.29. ....	74
II.2.3 NRMS and RMS of receiver geometry Pattern B and using 95% velocity. The figure geometries are same as Figure II.2.29. ....	75
II.2.3 NRMS and RMS of receiver geometry Pattern C and using 95% velocity. The figure geometries are same as Figure II.2.29. ....	76
II.3.1 The computation method of one-dimensional averaging. I take average by green arrow direction and pink arrow direction. The blue points of each graph are NRMS value of each point that are not averaging. The red line of each graph is average of blue points and red circles are injection point. ....	79
II.3.2 One-dimensional averaging value. ID numbers correspond to Table II.3.1. Blue crosses are NRMS and Red crosses are RMS. ....	79
II.3.3 The computation method of two-dimensional averaging. I take grids on the NRMS or RMS horizontal section and take average in each grid. Blue circles on the averaging figure show injection portion. The color between before and after averaging does not indicate same value. ....	81
II.3.4 The rating value of two-dimensional averaging. Blue crosses are NRMS and Red crosses are RMS. ....	82
II.3.5 (a) NRMS and (b) DNRMS. ....	83
II.3.6 The rating value of one-dimensional averaging of NRMS (blue crosses) and DNRMS (red crosses). ....	83
II.3.7 The rating value of two-dimensional averaging of NRMS (blue crosses) and DNRMS (red crosses). ....	84

II.3.8	The concept of seismic interferometry using multiples for construct the pseudo shot record. ....	86
III.1.1	Comparison of (a) computing performance and (b) memory bandwidth between GPU and CPU. Green and blue lines denote GPU and CPU, respectively. ....	93
III.2.1	Hardware implementation of GPU. Device is connected at PCI Express. GPU is the computation unit on the GPU board and includes processors (green) and memories (orange). Device memory could be found outside of the GPU. ....	96
III.2.2	The concept of GPU programming. The colors in this figure corresponds to the same explanation as stated in Figure III.2.1. One thread is calculated at one processor, one block is being input to one multiprocessor. ....	96
III.2.3	Snapshots of particle motion. This figure shows 40,000 particles inside the model for several time steps. Black denotes the area inside the model, in which the particles are displaced. ....	99
III.2.4	Thirteen grid positions for computing one grid on the center of grids of next step. A cube shows one grid. Thirteen grids are needed (white and red cubes) for calculating one grid of next step (red cube). Two times from 13 grids to 1 grid for updating the data which calculated velocity and displacement, respectively should be simulated. ....	101
III.2.5	Copied grids to shared memory including wing area. White cubes are wing area's cube and red cubes are calculating area for next time steps. The block size is $4 \times 4 \times 2$ . The number of grids of wing area are 4 times bigger than one of calculating area. ....	101
III.2.6	Computation time of 2D elastic wave propagating simulation. GPU Case A (blue line) used only device memory, GPU Case B (red line) used both device and shared memory and GPU Case C (green line) used only shared memory. The value of this plot is shown in Table III.2.3. The fastest time of each grid and case without distinction of blocksize is chosen. ....	103

III.2.7 Snapshots of 3D elastic propagating simulation by 0.15 sec timestep. On the depth at  $z = 120$  one can see the reflection wave of P wave at (d) and S wave at (e)..... 105

III.2.8 Shot gather of Figure III.2.4. The receivers are set at  $(y, z) = (250, 0)$  and every 5  $x$  grid of 25m. .... 106

III.2.9 Comparison of Z-component displacement between simulation result and analytic solution. Black line is analytical value and red line is simulating value. The distance between source and receiver point is 1200m. .... 107

III.2.10 The slice of model for  $CO_2$  injection simulation. The sky blue area in middle of the model shows  $CO_2$ . .... 107

III.2.11 The shot gather of receiver line. (a)before injection, (b)after injection, (c)difference of (a) and (b). The amplitude of (a) and (b) is same, and (c) is quite small. .... 108

III.A.1 (a) absorbed factor. Five cases that are different  $\alpha$  and the number of grids are shown. (b) the ratio of amplitude between direct and reflected from absorbed boundary wave. This ratio is calculated that reflected wave amplitude divided by direct wave amplitude. The label of  $x$  axis shows the  $\alpha$  and the number of grids. The numbers in the graph are exact values of each case. This amplitude is observed at the received point, which is 100 grids far from absorbed boundary and 50 grids far from source point. RMS amplitude for calculation and the calculation length is 40ms which corresponds to the length of source wavelet is used. .... 118

III.A.2 Grid dispersion by too high frequency. (b)100Hz of input source wavelet is too high to get stable result in this model. .... 119

# List of Tables

II.2.1 The properties of synthetic model.....	48
II.2.2 The number of sources and receivers in synthetic model .....	48
II.2.3 The properties of realistic model .....	58
II.2.4 The number of sources and receivers in realistic model .....	60
II.3.1 The relationship of ID number and the variety of data.....	80
II.3.2 The comparison of value in each analyzing method .....	85
III.2.1 The comparison of computation time between CPU and GPU. This computation time do not include the data output time.....	98
III.2.2 Computation time of 2D elastic wave propagating simulation on CPU.....	100
III.2.3 Computation time of 2D elastic wave propagating simulation on GPU.....	102
III.2.4 Computation time of 3D elastic wave propagating simulation .....	102
III.3.1 Comparison of computation time of 2D elastic wave propagating simulation by coa- lescing .....	110
III.3.2 Additional memory copy time by GPU computing of 3D elastic wave propagating simulation .....	110

## Acknowledgement

I would like to acknowledge to my supervisors in my university, first and maximum. Because I will continue a student as Ph.D. candidate and I am shy, my acknowledgement is little euphemistic. Prof. Toshifumi Matsuoka brought me so many chances, knowledge and freedoms. He does never restrict my research by his idea but teach me everything he know and give me good advices. Our discussions are usually in my car over once per week and the style is brainstorming. We discuss much type of things from various aspects and I sometimes flash great idea into my mind. He also authorize me great experiences in the outside of university. Because I get many chances, the contents of this thesis do not covered my all works in my Master candidate term. Associate Prof. Yasuhiro Yamada showed pure interest in Geology like a child and this is the reason to choose this laboratory. This is also true for Associate Prof. Xue Ziqiu, his enthusiasms to  $CO_2$  stimulate my interest and this topic is including this thesis. Assistant Prof. Takeshi Tsuji is a great scientist for me and also my first target. His keen advices that are covered by many trivial things are really helpful. Assistant Prof. Kyosuke Onishi who is in Akita Univ. now are good advice about seismic interferometry and give me a quite valuable traffic noise data that are used in interferometry part.

Laboratory's members are all argufies and I learn many things. I avoid to say thank individually because we are a big group.

Mr. Akihisa(May-Q) Takahashi, Susumu Abe, Kenichi Akama, Kazuya Shiraiishi and all other JGI, Inc. members accepted me for three times, five months. I can improve my basically skill for Geophysicist. Carbon Dioxide Capture and Storage (CCS) part is a co-reseach with JGI.

Prof. Roel Snieder in Colorado School of Mines is a great educator as same as scientist. I stayed CSM for two months and research cross-coherence seismic interferometry. I want to show great thanks to Roel who accepted me and continue contacts even now. Because the meeting with him is tough but great time that are difficult to say, I aspire keeping contacts.

I show special thanks to my mother, father and brothers who give me joy time. Finally, I great appreciate to Rie's support, too.

# Introduction

According to Oxford Advanced Learner's Compass, the word "virtual" is defined as "1) almost or very nearly the thing described, so that any slight difference is not important. 2) made to appear to exist by the use of computer software, for example on the Internet." I usually consider that "virtual" is computer simulation. However, the simulation is second meaning of "virtual", and first one is more abstract. When I translate "virtual" to one Japanese word, the meaning of this word is possibly "imaginary". I know the meaning of "virtual" and "imaginary" is difference, but I cannot remind the suitable word. Thus, the meaning of "virtual" has a nuance that is difficult to say in Japanese. We dare to say, it is "it is not fact but it seems to be true". Recently the method of "virtual" is often to use in anywhere, especially in the computer. It is often not consist between hardwares and softwares; for examples between the number of physical cores in CPU and cores that are recognized by operating system because of "virtual" that is Hyper-Threading Technology. Another example, the amount of memory increases by using HDD as "virtual" memory.

Therefore, I often experience a benefit of "virtual" in my life. How about in Geophysics? There are two main topics of "virtual" in Geophysics such as (1) interferometry and (2) simulation. Because "virtual" is not a real but never a fib, the calculation in "virtual world" is also useful in Geophysics. The importance of "virtual" in Geophysics is developed simultaneously with the development of computers. In this thesis, I research these two things. The purpose of this thesis is reveal a potential of "virtual world" what "virtual" can make clear, what "virtual" can do, and what changing after now. This thesis has three parts and each part is the portion of "virtual world" in Geophysics.

When I hear the term of "virtual", I remember "Virtual source method" soon. It is possibly that I often consider seismic interferometry. Virtual source method can consider virtual source point, however it is not mean that this source is worse than active source (Mehta et al., 2008a). Herein, I treat that virtual source and seismic interferometry is same method. We can say briefly that seismic interferometry can construct the Green's function between two points by cross-correlation. Thus, I set pseudo source to place that are difficult to put real source and I can get shot record that is only gotten by seismic interferometry. Virtual Real Source (VRS) is one of applications of seismic interferometry (Behura, 2007, ). This method deconvolutes wavefield that are constructed from seismic interferometry by active source wavefield and extract source wavelet. The wavefield constructed by seismic interferometry has power spectrum of source wavelet and it is impossible to get this wavefield using active source. Sometimes, the virtual data is more superior to the real data.

In the first part of this thesis, I described about seismic interferometry. Recently, some papers which applied interferometry to real observed data were published including seismic sections ((Draganov et al., 2009)). In this study, I propose a new algorithm interferometry called as "cross-coherence". We applied this algorithm for noisy data in the field. This method can retrieve more stable reflection structures under existence of severe noise. We estimate the availability of cross-coherence from all aspects of mathematics, numerical experience, and data processing.

Seismic interferometry uses real data to construct virtual data. On the other hand, simulation like a modeling is little difference in the method of constructing to virtual data. Although I can control some degrees the initial model or variation of time step in simulation, it is not meaningful that this simulation is quite alienate from real phenomena that is not virtual. We should keep close to real as possible as I can that is depended on computation time, technique and cost. The modeling methods are ray tracing, Finite-Different Time-Domain method (FDTD) or Finite-Element method (FEM), and I obtain accurate answer when I take latter one. However, it takes much time for calculation in FDTD and FEM. Presently, I have no choice in 3D modeling except ray tracing.

The subject of second part is 4D seismic time-lapse monitoring  $CO_2$  injection to subsurface struc-

ture. It is important to monitor the activity of  $CO_2$  distribution for many years after the injection process. Presently monitoring using seismic wave is considered as most superior method. We think it is better to use placed permanently Ocean-Bottom Cable (OBC) in the point of cost or quality of data. In this thesis, I use 3D ray tracing to two types of models and check the detectability of  $CO_2$  distribution when I change the number of receivers. The detectability is not only visibility but the quantitative indicators. Although If I can, I want to use FDTD and simulate full-wave, it is impossible to finish the calculating in practical time by ordinary computer.

In the third part, I develop the 3D elastic wave propagating simulator using FDTD on Graphic Processing Unit (GPU). GPU is getting usage for scientific calculation and I note GPU are an accelerator for calculation. Although the main stream of 3D modeling is ray tracing described in second part, it is much better to simulate by FDTD method because I can simulate similar condition with real world. We think that it is possible to finish the calculation in practical time if I use the whole potential of GPU. In this thesis, I improve the computation time of 3D elastic wave propagating simulation using FDTD method by GPU.

## **Part I**

# **Interferometry by cross-coherence**

# Chapter I.1

## Introduction

The main purpose of seismic interferometry is to construct a Green's function between two geophones, hydrophones, or accelerometers by using earthquakes, microtremors, or artificial seismic sources. Because one geophone behaves like a source and the other behaves like a receiver, I can obtain pseudo shot gather without an active source. The simplest algorithm in seismic interferometry is cross-correlation (Wapenaar, 2003). Some authors show various derivations to Green's function from cross-correlation using representation theorem (Wapenaar and Fokkema, 2006; Wapenaar, 2004), the principle of time reversal (Roux and Fink, 2003), and stationary points (Snieder et al., 2006). However, it is difficult to apply cross-correlation interferometry to data with complex source wavelets. If I plan to use cross-correlation, I should estimate the source wavelets by some other methods (Ikelle et al., 1997) and remove their effects. If I adopt the deconvolution method, I can remove the source wavelets by employing spectral division. The mathematical theory of the deconvolution method has been derived previously (Vasconcelos and Snieder, 2008a) and applied to field data (Snieder et al., 2006; Vasconcelos and Snieder, 2008b). However, in the deconvolution method, it is difficult to obtain a stable result from division; white noise must be added to avoid dividing by 0 or an extremely small number. The multidimensional deconvolution method is used for seismic interferometry (Wapenaar et al., 2008a; Wapenaar et al., 2008b).

Because all these methods have both advantages and disadvantages (Wapenaar et al., 2008c;

Snieder et al., 2009), I should choose the method that best suits the data. It is, however, difficult to apply these methods to data a lot of noise; interferometry methods can create additional noise owing to insufficient stacks or stationary noise.

Here I propose a new "cross-coherence" method, which can suppress the noise contribution. Cross-coherence is used to calculate the cross-correlation of traces normalized by their spectral amplitudes in the frequency domain. Studies usually employ velocity analysis for the equation of cross-coherence (Neidell and Taner, 1971) but some studies have used the new stacking method to obtain clear seismic sections (Fomel, 2009; Liu et al., 2009). Cross-coherence method focuses on the phase of each trace and suppresses the amplitude information. This part presents the mathematical theory of interferometry by cross-coherence, demonstrates the relationship between the equation and the simulation, and presents a case study using field data.

## Chapter I.2

# Theory of Interferometry

I describe the wavefield  $u(r, s, \omega)$  that is excited at  $s$  and received at  $r$  in the frequency domain. In this part, all equations are in the frequency domain unless otherwise noted; therefore I omit the  $\omega$  term in the equations. If I do not consider a noise term, the wavefield can be described by the multiplication of a source wavelet and a Green's function;

$$u(r, s) = W(s)[G_0(r, s) + G_s(r, s)], \quad (\text{I.2.1})$$

where  $W(s)$  is the source wavelet and  $G(r, s)$  is a Green's function. For simplicity, I do not consider the coupling between the receiver and the ground around the receiver point. I assume that the receiving motion corresponds perfectly to the motion of the ground. I can divide a Green's function into two waves. The subscripts 0 and  $s$  of Green's function  $G$  indicate unperturbed and scattered waves, respectively. These two waves are represented by direct and reflected waves, respectively. Therefore, these terms are related as

$$\frac{|G_s(r, s)|}{|G_0(r, s)|} \ll 1. \quad (\text{I.2.2})$$

From equations I.2.1 and I.2.2, I use a Taylor expansion to expand the basic equation of seismic interferometry.

## I.2.1 Cross-correlation and deconvolution

I review the cross-correlation and deconvolution methods in order to compare them with my proposed cross-coherence method (Snieder et al., 2006; Vasconcelos and Snieder, 2008a). The equation for retrieval of the Green's function from  $r_B$  to  $r_A$  by cross-correlation is

$$\begin{aligned} C_{AB} &= u(r_A, s)u^*(r_B, s) \\ &= |W(s)|^2 G(r_A, s)G^*(r_B, s). \end{aligned} \quad (\text{I.2.3})$$

The asterisk denotes complex conjugation. I integrate this equation over a closed surface  $\partial V$  that includes all sources, giving

$$\oint_{\partial V} C_{AB} ds = \langle |W(s)|^2 \rangle \oint_{\partial V} G(r_A, s)G^*(r_B, s) ds, \quad (\text{I.2.4})$$

where  $\langle |W(s)|^2 \rangle$  is the average of the power spectra of the source wavelets. Because the multiplication of  $\oint_{\partial V} G(r_A, s)G^*(r_B, s) ds$  in equation I.2.4 is equal to  $G(r_A, r_B)$  (Wapenaar et al., 2006), I obtain the Green's function between the two receivers. By considering the unperturbed and scattered waves, I obtain four terms from equations I.2.1 and I.2.3,

$$\begin{aligned} \oint_{\partial V} C_{AB} ds &\approx \underbrace{\oint_{\partial V} u_0(r_A, s)u_0^*(r_B, s) ds}_{C_{AB}^1} + \underbrace{\oint_{\partial V} u_s(r_A, s)u_0^*(r_B, s) ds}_{C_{AB}^2} \\ &+ \underbrace{\oint_{\partial V} u_0(r_A, s)u_s^*(r_B, s) ds}_{C_{AB}^3} + \underbrace{\oint_{\partial V} u_s(r_A, s)u_s^*(r_B, s) ds}_{C_{AB}^4}. \end{aligned} \quad (\text{I.2.5})$$

While in contrast, the basic equation of the deconvolution method is

$$D_{AB} = \frac{u(r_A, s)}{u(r_B, s)} = \frac{G(r_A, s)}{G(r_B, s)} = \frac{G(r_A, s)G^*(r_B, s)}{|G(r_B, s)|^2}. \quad (\text{I.2.6})$$

Deconvolution can remove the effects of the source wavelet  $W(s)$ . Because the absolute value in the

frequency domain does not contribute to phase, the the phase of  $D_{AB}$  is decided by the numerator of the last term  $G(r_A, s)G^*(r_B, s)$  in equation I.2.6. The term  $G(r_A, s)G^*(r_B, s)$  is the same as in the cross-correlation method (equation I.2.3). Therefore, the deconvolution method can also be used to construct  $G(r_A, r_B)$  when I integrate over a closed surface  $\partial V$  that includes the sources. Equation I.2.6 can be expanded as

$$\begin{aligned}
 D_{AB} &= \frac{G(r_A, s)}{G(r_B, s)} \\
 &= G(r_A, s) \left\{ \frac{1}{G_0(r_B, s) \left( 1 + \frac{G_s(r_B, s)}{G_0(r_B, s)} \right)} \right\} \\
 &= \frac{G(r_A, s)}{G_0(r_B, s)} + \frac{G(r_A, s)}{G_0(r_B, s)} \times \sum_{n=1}^{\infty} (-1)^n \left( \frac{G_s(r_B, s)}{G_0(r_B, s)} \right)^n.
 \end{aligned} \tag{I.2.7}$$

From the relationship of the expression I.2.2, I keep only the  $0^{th}$ - and first-order scattered terms and integrate them over the source position  $\partial V$  to obtain the following three terms:

$$\oint_{\partial V} D_{AB} ds = \underbrace{\oint_{\partial V} \frac{G_0(r_A, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} ds}_{D_{AB}^1} + \underbrace{\oint_{\partial V} \frac{G_s(r_A, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} ds}_{D_{AB}^2} - \underbrace{\oint_{\partial V} \frac{G_0(r_A, s)G_0^*(r_B, s)G_s(r_B, s)}{|G_0(r_B, s)|^2 G_0(r_B, s)} ds}_{D_{AB}^3}. \tag{I.2.8}$$

## I.2.2 Cross-coherence

The basic equation of cross-coherence  $H_{AB}$  is defined as.

$$H_{AB} = \frac{u(r_A, s)u^*(r_B, s)}{|u(r_A, s)||u(r_B, s)|} \tag{I.2.9}$$

in the frequency domain. The numerator of equation I.2.9 is the same as in the expression for cross-correlation (equation I.2.3), and the denominator contains the absolute values of each factor. In other words, in this method are multiplied then divided by their absolute values. I can treat each wavefield

as the product of amplitude and phase,

$$u(r_A, s) = A(\omega) \exp[i\phi(\omega)], \quad (\text{I.2.10})$$

where  $A(\omega)$  is the absolute amplitude and  $\exp[i\phi(\omega)]$  is the phase. Thus, when a wavefield is divided by its absolute value, only phase information remains. Because the amplitude is easily changed because of the sensitivity of the receivers, this equation should retrieve reliable information. I can rewrite equation I.2.9 as

$$H_{AB} = \frac{G(r_A, s)G^*(r_B, s)}{|G(r_A, s)||G(r_B, s)|}, \quad (\text{I.2.11})$$

because cross-coherence cancels the source wavelet term  $W(s)$  by division as in the case of deconvolution. I integrate equation I.2.11 over a closed surface  $\partial V$  containing all sources, giving

$$\oint_{\partial V} H_{AB} ds = \oint_{\partial V} \frac{G(r_A, s)G^*(r_B, s)}{|G(r_A, s)||G(r_B, s)|} ds. \quad (\text{I.2.12})$$

Since the right-hand side of equation I.2.12 is divided by absolute values, I can obtain only phase information. Accordingly, I can obtain the phase of the Green's function between two receivers.

To investigate the mathematical meaning of equation I.2.12, I expand it by inserting equation I.2.1 and using a Taylor expansion. At first, I expand only  $G$  divided by  $|G|$  and do not integrate them. I obtain

$$\begin{aligned} \frac{G}{|G|} &= \frac{G_0 + G_s}{|G_0 + G_s|} = \frac{G_0 + G_s}{\sqrt{(G_0 + G_s)(G_0^* + G_s^*)}} \\ &= (G_0 + G_s) \frac{1}{\sqrt{|G_0|^2 + G_0 G_s^* + G_s G_0^* + G_s G_s^*}} \\ &= (G_0 + G_s) \frac{1}{|G_0| \sqrt{1 + \frac{G_0 G_s^* + G_s G_0^* + G_s G_s^*}{|G_0|^2}}} \\ &= \frac{G_0 + G_s}{|G_0|} \left[ 1 + \sum_{n=1}^{\infty} \left\{ \frac{1}{n!} \prod_{i=1}^n \left( \frac{1}{2} - i \right) \left( \frac{G_0 G_s^* + G_s G_0^* + G_s G_s^*}{|G_0|^2} \right)^n \right\} \right]. \end{aligned} \quad (\text{I.2.13})$$

I calculate the expansion of equation I.2.12 by using equation I.2.13 and keeping track of only the  $0^{th}$ - and linear-terms of the scattered wavefields that are the same as in deconvolution. I show the second-order analysis in Appendix A. Here, linear means that these waves are reflected fewer than two times. Equation I.2.12 can be written as follows:

$$\begin{aligned}
 \oint_{\partial V} H_{AB} ds \approx & \underbrace{\oint_{\partial V} \frac{G_0(r_A, s) G_0^*(r_B, s)}{|G_0(r_A, s)| |G_0(r_B, s)|} ds}_{H_{AB}^1} \\
 & + \frac{1}{2} \underbrace{\oint_{\partial V} \frac{G_s(r_A, s) G_0^*(r_B, s)}{|G_0(r_A, s)| |G_0(r_B, s)|} ds}_{H_{AB}^2} + \frac{1}{2} \underbrace{\oint_{\partial V} \frac{G_0(r_A, s) G_s^*(r_B, s)}{|G_0(r_A, s)| |G_0(r_B, s)|} ds}_{H_{AB}^3} \\
 & - \frac{1}{2} \underbrace{\oint_{\partial V} \frac{G_0(r_A, s) G_0^*(r_B, s) G_0(r_A, s) G_s^*(r_A, s)}{|G_0(r_A, s)|^3 |G_0(r_B, s)|} ds}_{H_{AB}^4} \\
 & - \frac{1}{2} \underbrace{\oint_{\partial V} \frac{G_0(r_A, s) G_0^*(r_B, s) G_s(r_B, s) G_0^*(r_B, s)}{|G_0(r_A, s)| |G_0(r_B, s)|^3} ds}_{H_{AB}^5}. \tag{I.2.14}
 \end{aligned}$$

The five terms of equation I.2.14 have different phases, which means that these terms represent different raypaths and have different physical meanings. Because the denominators in all the terms are absolute values, they do not contribute to the phase. Thus, I concentrate on the numerators for understanding the phases.  $H_{AB}^1$  indicates causal and acausal unperturbed wavefields that are represented by direct waves between  $r_A$  and  $r_B$ .  $H_{AB}^2$  has the phase of the wave that is the cross-correlation between the scattered wavefield at  $r_A$  and the unperturbed wavefield at  $r_B$ . Thus, this term represents the scattered wavefield  $G_s(r_A, r_B)$  that is excited at  $r_B$  and propagates to  $r_A$ . This is a valuable wavefield for reflection analysis.  $H_{AB}^3$  is the acausal wavefield of  $H_{AB}^2$ . Both  $H_{AB}^4$  and  $H_{AB}^5$  have the "free-point boundary condition" as described by (Vasconcelos and Snieder, 2008a). From equation I.2.14, I can recognize that the wavefield of  $H_{AB}^5$  represents the convolution of two virtual wavefields. One wave,  $G_0(r_A, s) G_0^*(r_B, s)$ , is excited at  $r_B$  and propagates directly to  $r_A$ . The other wave,  $G_s(r_B, s) G_0^*(r_B, s)$ , is excited at  $r_B$  propagates to the scattering point  $x_s$ , and then reflects from this point to  $r_B$ . Because the convolutions of these two wavefields correspond to the summation of their phases, the wavefield

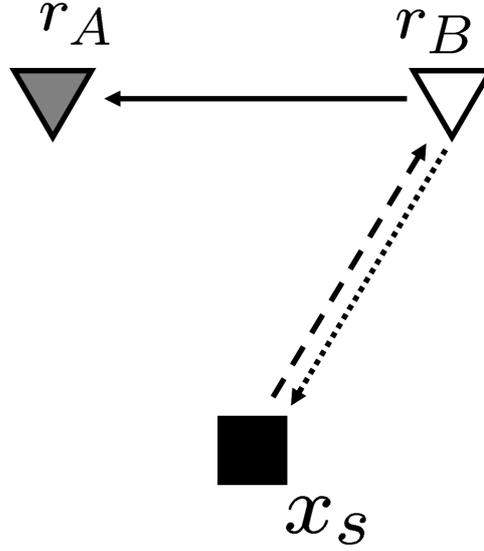


Figure I.2.1: The raypath of the term  $H_{AB}^5$ . The triangles show two receivers,  $r_A$  (gray) and  $r_B$  (white). The black square is the scattering point  $x_s$ .  $r_B$  is a pseudo point source. This wavefield is excited at  $r_B$  propagates to  $x_s$ , as shown by the dotted arrow, and is reflected and returns to  $r_B$  as shown by the dashed arrow. Then the wavefield is reflected at  $r_B$  toward  $r_A$  as shown by the solid arrow.

of  $H_{AB}^5$  is excited at  $r_B$  returns to  $r_B$  through  $x_s$ , and is reflected toward  $r_A$  from  $r_B$ , which is free-point boundary condition (Figure I.2.1).  $H_{AB}^4$  is the acausal part of  $H_{AB}^5$ . When  $r_A = r_B$ ,  $H_{AB}^2$ ,  $H_{AB}^3$ ,  $H_{AB}^4$ , and  $H_{AB}^5$  all vanish and only  $H_{AB}^1$  remains. In this case, the right-hand side of equation I.2.14 is equal to 1. Therefore, zero-offset cross-coherence interferometry gives

$$H_{BB}(t) = \delta(t) \quad (\text{I.2.15})$$

in the time domain. Deconvolution has the same property. If  $r_A = r_B$  in equation I.2.8, the right-hand side becomes equal to 1.  $D_{AB}^3$  in equation I.2.8 is induced by the free-point boundary condition, and  $r_B$  is a free-point boundary. This point is fixed. However, this phenomenon does not appear in an actual wavefield; thus, this boundary condition is nonphysical. In any case, wavefields with the free-point boundary condition can be treated as nonphysical noise created by interferometry.

Next, I compare the terms in equations I.2.5, I.2.8 and I.2.14. Note One can recognize that  $H_{AB}^1$  and  $C_{AB}^1$ ,  $H_{AB}^2$  and  $C_{AB}^2$ , and  $H_{AB}^3$  and  $C_{AB}^3$  have the same phases, respectively. Therefore, these indicate

identical wavefields with different amplitudes.  $C_{AB}^4$  consists of the convolution of the two scattered waves  $u_s(r_A, s)$  and  $u_s^*(r_B, s)$ . Because equation I.2.14 includes only the first-order scattered wave, it seems that cross-coherence does not include a term corresponding to  $C_{AB}^4$ . However, a wavefield of this phase also appears at higher orders of cross-coherence (Appendix A).  $C_{AB}^4$  is not useful because it, too, represents a nonphysical wavefield caused by interferometry methods. The same may be said of the deconvolution terms  $D_{AB}^1$ ,  $D_{AB}^2$ , and  $D_{AB}^3$ , which have the same phases as  $H_{AB}^1$ ,  $H_{AB}^2$ , and  $H_{AB}^5$ , respectively.

As I have shown, several terms within these three methods have common phases, while the other terms have different phases. The phases express characteristics of each method. For example, cross-correlation does not have the free-point boundary condition and deconvolution has only a causal scattered wavefield. Furthermore, one of the terms in cross-coherence, namely  $H_{AB}^4$ , is unique, and because a fraction (i.e.,  $\frac{1}{2}$ ) multiplies the scattered terms of equation I.2.14, the contribution of the scattered wavefields is smaller than in the other two methods.

## **Chapter I.3**

# **Advantages and Disadvantages of the cross-coherence method**

Like the other two methods, cross-coherence has both advantages and disadvantages, which I describe in this chapter.

### **I.3.1 Correction of the amplitude variation among traces**

Here I consider the management of data whose amplitudes vary by trace amounts owing to considerations such as special regional sources and differences in positioning or sensitivity between receivers. The sensitivity should be the same between receivers; however, achieving this is usually difficult because of reasons such as positioning or an incline. When receiver position  $\alpha$  records a larger amplitude

than others owing to a loud source near point  $\alpha$ , the equations of the three methods are

$$C_{A\alpha} = G(r_A, s)G^*(r_\alpha, s) \quad (I.3.1)$$

$$D_{A\alpha} = \frac{G(r_A, s)G^*(r_\alpha, s)}{|G(r_\alpha, s)|^2} \quad (I.3.2)$$

$$D_{\alpha A} = \frac{G(r_\alpha, s)G^*(r_A, s)}{|G(r_A, s)|^2} \quad (I.3.3)$$

$$H_{A\alpha} = \frac{G(r_A, s)G^*(r_\alpha, s)}{|G(r_A, s)||G(r_\alpha, s)|}, \quad (I.3.4)$$

where is two receivers that are  $r_A, r_\alpha$ .  $r_A$  has the average amplitude of all receivers and  $r_\alpha$  has a larger amplitude. Deconvolution interferometry is asymmetric, and the value changes when I exchange the pseudo source and receiver point. Thus, this method has two types of equations. The result of the other two methods do not change if you exchange the  $\alpha$  and  $A$ . I compare the amplitudes among equations I.3.1 - I.3.4.

In equation I.3.1, because the amplitude of  $G(r_A, s)$  is the average and that of  $G^*(r_\alpha, s)$  is large,  $C_{A\alpha}$  has a larger amplitude than the others. In equation I.3.2, since  $G(r_A, s)$  has the average amplitude and  $G(r_\alpha, s)$  has a large amplitude, the denominator has a larger amplitude than the numerator. As a result,  $D_{A\alpha}$  has a smaller amplitude.  $D_{\alpha A}$  in equation I.3.3, on the other hand, has a large amplitude that is caused by the large amplitude of the numerator and the average amplitude of the denominator. As mentioned above, the amplitudes of  $C_{A\alpha}$ ,  $D_{\alpha A}$ , and  $D_{A\alpha}$  are different from the average amplitude. Accordingly, when I use several receivers in an analysis that includes  $\alpha$  using cross-correlation or deconvolution, the amplitude of result is also unbalanced. Thus, I have the additional task of removing these variations. In equation I.3.4, there are same terms in both denominator and numerator. It is clear that the amplitudes of the denominator and numerator are the same, and the amplitude of  $H_{A\alpha}$  is the average. That is, cross-coherence can remove the influence of amplitude variation and achieve a stable amplitude without additional work. Nevertheless, when I use the cross-coherence method, I cannot compensate for the difference of phase between receivers, which for example, could be caused by an incline of the receivers. However, such an incline is typically due to human error and can be prevented.

Usually, an error caused by the receivers appears in the amplitude and not in the phase.

Furthermore, in deconvolution, white noise has to be added to prevent division by 0. If the data's trace balance is not good and I make the white-noise parameter large, the calculation is close to that of cross-correlation. However, if white noise is set to be small, the trace balance of the results is not flat. In cross-coherence, a term is divided by itself. Hence, if a term is close to 0, I can get a stable answer without a stability factor.

### I.3.2 Suppressing the influence of noise

In the previous section, I presented the basic equation of cross-coherence without noise. However, in the real world, data include noise caused by microtremors, observation equipment, and human activities. This noise can present obstacles to retrieval of a Green's function using seismic interferometry. Waves unusable for interferometry are considered to be noise in this study. I rewrite the wavefield  $u(r_A, s)$  to include a noise term as follows:

$$u(r_A, s) = G(r_A, s) + N(r_A). \quad (\text{I.3.5})$$

For simplicity, I do not separate  $u(r_A, s)$  into unperturbed and scattered waves. Furthermore, the source signature  $W(s)$  is set to be 1. By substituting equation I.3.5 into equations I.2.3, I.2.6, and I.2.9, I obtain the terms for cross-correlation, deconvolution, and cross-coherence using wavefields that include noise as follows:

$$C_{AB} = (G_A + N_A)(G_B^* + N_B^*) \quad (\text{I.3.6})$$

$$D_{AB} = \frac{G_A + N_A}{G_B + N_B} \quad (\text{I.3.7})$$

$$H_{AB} = \frac{(G_A + N_A)(G_B^* + N_B^*)}{|G_A + N_A||G_B + N_B|} \quad (\text{I.3.8})$$

Using these equations, I investigate the contribution of the noise term. I compare the signal-to-noise ratio among the three methods by gradually changing the noise amplitude.

In cross-correlation (equation I.3.6), the signal term is  $G_A G_B^*$  and the noise term is  $G_A N_B^* + N_A G_B^* + N_A N_B^*$ . First, I focus on the this signal term.  $G_A G_B^*$  does not include noise  $N(r)$ , so as the noise amplitude increases, the amplitude of the signal term remains stable. On the other hand, the amplitude of the noise term of cross-correlation increases as the noise amplitude increases. Turning now to deconvolution, the signal term of equation I.3.7 is  $\frac{G_A G_B^*}{|G_B + N_B|^2}$  and the noise terms are  $\frac{N_A G_B^* + G_A N_B^* + N_A N_B^*}{|G_B + N_B|^2}$ . As the noise amplitude increases, the amplitude of the signal decreases and the one for noise is stable. Next, I investigate cross-coherence in equation I.3.8 in which the signal term comes from  $\frac{G_A G_B^*}{|G_A + N_A||G_B + N_B|}$ . In the signal term, as the noise amplitude increases, the amplitude of the denominator also increases and that of the numerator is stable. On the other hand, the noise term of cross-coherence comes from  $\frac{N_A G_B^* + G_A N_B^* + N_A N_B^*}{|G_A + N_A||G_B + N_B|}$ , in which second-order noise terms are included in both the denominator and numerator. Thus, the noise amplitude is stable.

I assume  $|N|/|G| < 1$  and expand equations I.3.6 - I.3.8, ignoring noise terms higher than second-order. For the noise, sometimes  $|N|/|G| > 1$ , but at that point, it is difficult to create an image. Thus, I ignore points where the noise amplitude is greater than the signal's:

$$C_{AB} = G_A G_B^* + N_A G_B^* + G_A N_B^* + N_A N_B^* \quad (\text{I.3.9})$$

$$D_{AB} = \frac{G_A G_B^*}{|G_B|^2} + \frac{N_A G_B^*}{|G_B|^2} - \frac{G_A G_B^* N_B G_B^*}{|G_B|^4} - \frac{N_A G_B^* N_B G_B^*}{|G_B|^4} + \frac{G_A G_B^* N_B G_B^* N_B G_B^*}{|G_B|^6} \quad (\text{I.3.10})$$

$$\begin{aligned} H_{AB} = & \left( 1 - \frac{1}{4} \frac{|N_A|^2}{|G_A|^2} - \frac{1}{4} \frac{|N_B|^2}{|G_B|^2} \right) \frac{G_A G_B^*}{|G_A||G_B|} \\ & + \frac{1}{2} \frac{N_A G_B^*}{|G_A||G_B|} + \frac{1}{2} \frac{G_A N_B^*}{|G_A||G_B|} + \frac{1}{4} \frac{N_A N_B^*}{|G_A||G_B|} \\ & - \frac{1}{2} \frac{G_A G_B^* G_A N_A^*}{|G_A|^3 |G_B|} - \frac{1}{2} \frac{G_A G_B^* N_B G_B^*}{|G_A||G_B|^3} - \frac{1}{8} \frac{N_A G_A^* N_A G_B^*}{|G_A|^3 |G_B|} - \frac{1}{8} \frac{G_A N_B^* G_B N_B^*}{|G_A||G_B|^3} \\ & - \frac{1}{4} \frac{G_A N_A^* G_A N_B^*}{|G_A|^3 |G_B|} - \frac{1}{4} \frac{N_A G_B^* N_B G_B^*}{|G_A||G_B|^3} \\ & + \frac{3}{8} \frac{G_A G_B^* G_A N_A^* G_A N_A^*}{|G_A|^5 |G_B|} + \frac{3}{8} \frac{G_A G_B^* N_B G_B^* N_B G_B^*}{|G_A||G_B|^5} + \frac{1}{4} \frac{G_A G_B^* G_A N_A^* N_B G_B^*}{|G_A|^3 |G_B|^3} \end{aligned} \quad (\text{I.3.11})$$

Because these equations are complex and not useful, I consider the average and variance of equations I.3.9 - I.3.11 to investigate the influence of the noise term. In this study, I assume the noise is random. The average of random noise is 0 and the average of the convolution of  $G$  and  $N$  is also 0. The averages of equations I.3.9 - I.3.11 are

$$\langle C_{AB} \rangle = G_A G_B^* \quad (\text{I.3.12})$$

$$\langle D_{AB} \rangle = \frac{G_A G_B^*}{|G_B|^2}; \quad (\text{I.3.13})$$

$$\langle H_{AB} \rangle = \left( 1 - \frac{1}{4} \frac{|N_A|^2}{|G_A|^2} - \frac{1}{4} \frac{|N_B|^2}{|G_B|^2} \right) \frac{G_A G_B^*}{|G_A| |G_B|}, \quad (\text{I.3.14})$$

in which " $\langle \rangle$ " indicates an average. Their variances are

$$\sigma_C^2 = |G_B|^2 \sigma_{N_A}^2 + |G_A|^2 \sigma_{N_B}^2 \quad (\text{I.3.15})$$

$$\sigma_D^2 = \frac{1}{|G_B|^2} \sigma_{N_A}^2 + \frac{|G_A|^2}{|G_B|^4} \sigma_{N_B}^2; \quad (\text{I.3.16})$$

$$\sigma_H^2 = \frac{1}{2|G_A|^2} \sigma_{N_A}^2 + \frac{1}{2|G_B|^2} \sigma_{N_B}^2. \quad (\text{I.3.17})$$

Here  $\sigma$  denotes standard deviation. The average exhibits the effect of noise after stacking hundreds of so many times. Among equations I.3.12, I.3.13, and I.3.14, only cross-coherence (equation I.3.14) has noise terms  $-\frac{1}{4} \frac{|N_A|^2}{|G_A|^2} - \frac{1}{4} \frac{|N_B|^2}{|G_B|^2}$ . Therefore, when I stack many times, the influence of noise remains in cross-coherence as a bias. However, in the real world, it is impossible to get an average value because it is difficult to stack a sufficient number of times. Thus, I should recognize the effect of single operation as same as an average that is one of stacked data. The variance is shown a single realization. I should also consider this value because I cannot stack enough times. A small variance means the contribution of noise is also small. It is difficult to compare the variances in equations I.3.15, I.3.16, and I.3.17 because they are dependent on the absolute value of the basic equation. Therefore, I calculate the

relative errors, which are defined as the variance divided by the basic equation:

$$\frac{\sigma_C}{|C_{AB}|} = \sqrt{\frac{\sigma N_A^2}{|G_A|^2} + \frac{\sigma N_B^2}{|G_B|^2}} \quad (I.3.18)$$

$$\frac{\sigma_D}{|D_{AB}|} = \sqrt{\frac{\sigma N_A^2}{|G_A|^2} + \frac{\sigma N_B^2}{|G_B|^2}} \quad (I.3.19)$$

$$\frac{\sigma_H}{|H_{AB}|} = \sqrt{\frac{\sigma N_A^2}{2|G_A|^2} + \frac{\sigma N_B^2}{2|G_B|^2}}. \quad (I.3.20)$$

From equations I.3.18 - I.3.20, the relative errors are related to each other as follows:

$$\frac{\sigma_H}{|H_{AB}|} = \frac{1}{\sqrt{2}} \frac{\sigma_C}{|C_{AB}|} = \frac{1}{\sqrt{2}} \frac{\sigma_D}{|D_{AB}|} \quad (I.3.21)$$

from equation I.3.21, I see the value of cross-coherence is  $\sqrt{2}$  times smaller than for the other methods.

In summary, I compared the averages and relative errors (equations I.3.12-I.3.14 and I.3.18-I.3.20) and considered the influence of noise. The effect of bias in average of cross-coherence (equation I.3.14) is smaller than the effect of  $\sqrt{2}$  in variance (I.3.21). Thus, cross-coherence can suppress the influence of noise, which can help in obtaining stable results from noisy data.

### I.3.3 Removing the amplitude information

Cross-coherence only estimates phase information and the resulting amplitudes are not correct. Therefore, I cannot apply an amplitude analysis such as the reflection coefficient or AVO. However, I must understand that for the cross-correlation and deconvolution methods, it is also difficult to judge whether the amplitude of a result is correct because the correctness of the amplitude obtained is related to the source distribution (Wapenaar and Fokkema, 2006). I can state clearly that the wavefield retrieved by the cross-coherence method can be applied only to phase analysis.

## Chapter I.4

# Simulation

In this chapter, I describe the two-dimensional acoustic Finite-Difference Time-Domain (FDTD) simulation and its connection to the mathematical expressions in the previous chapters. The simulation model (Figure I.4.1) consists of two flat horizontal layers in a volume of  $5,000 \times 5,000m$ . The thickness of the first layer is  $2,500m$  and the velocities of the top and bottom layers are  $1,500m/s$  and  $2,000m/s$ , respectively. For simplicity, the model is surrounded by absorbing boundaries because reflection waves from free-surface boundary produces complex results. The purpose of the simulation is to understand the behavior of the cross-coherence method, so the use of absorbing boundaries is acceptable.. I set the coordinates  $x$  and  $z$  with the upper left point at  $(x,z) = (0,0)$ . The positions of the two receivers,  $r_A$  and  $r_B$ , are  $(1,500,900)$  and  $(3,500,900)$ , respectively. The sources are distributed from  $(500,500)$  to  $(4,500,500)$  at intervals of  $50m$ .

### I.4.1 Simulation of noise-free data

I first simulate the wavefield without noise to confirm the behavior of each term in equations I.2.5, I.2.8, and I.2.14. Figure I.4.2 shows the wavefields that are obtained by applying the three methods of interferometry to simulated receiver gathers. In this part, I use the term "interferometric gathers" to refer to the gathers shown in Figure I.4.2. By stacking interferometric gathers with respect to all source

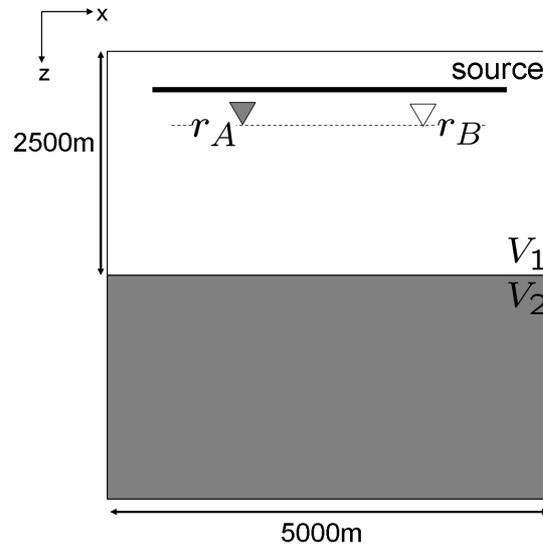


Figure I.4.1: Flat horizontal two-layer model for FDTD simulation. Volumes  $V_1$  and  $V_2$  have velocities  $1500m/s$  and  $2000m/s$ , respectively. There are two receivers,  $r_A$  (gray triangle) at  $(x, z) = (1500, 900)$  and  $r_B$  (white triangle) at  $(3500, 900)$ . The depth of the line of sources at  $50m$  increments is  $500m$ . The lateral edge positions are  $500$  and  $4500m$ . The simulated volume is  $5000 \times 5000m$  and the thickness of the first layer is  $2500m$ .

positions, I can obtain the wavefield that is excited at  $r_B$ , which is regarded as a pseudo source position, and received at  $r_A$ . Stacking means integration over a closed surface that encloses all sources. Figure I.4.2(a) shows the results of interferometry by using the cross-correlation method. Figure I.4.2(d) plots the traveltime of each term in equation I.2.5. Similarly, Figure I.4.2(e) is plotted with the travel time of Figure I.4.2(b) for deconvolution and Figure I.4.2(f) uses those of Figure I.4.2(c) for cross-coherence. There is a clear correspondence between the wavefields and each term in equation I.2.5. Also, weak waves can be seen in Figures I.4.2(a), I.4.2(b), and I.4.2(c) that do not appear among the traveltime curves in Figures I.4.2(d), I.4.2(e) and I.4.2(f). These waves come from higher-order scattered waves (see Appendix A) or they are simulation errors such as grid dispersions or waves that are reflected from the absorbing boundaries. The scattered waves in Figure I.4.2(c) are weaker than those in Figure I.4.2(b) because of the constants in equation I.2.14. In addition, cross-coherence also includes acausal wavefields. Waves of all phases for cross-correlation and deconvolution are included in the cross-coherence figure. Furthermore, deconvolution and cross-coherence have sharper wavelets than

those of cross-correlation because the first two methods remove source wavelets. These characteristics correspond to the properties obtained the equations. Note that the amplitudes are not the same among these three figures. The amplitudes of cross-correlation method (Figure I.4.2(a)) are much larger than the other two methods because of the power spectra of source wavelet. When I apply interferometry, I use only the causal part of the interferometric gathers and stack source position directions. Thus, I obtain the wavefield that is excited at  $r_B$  and received at  $r_A$ .

## I.4.2 Simulation of noise contaminated data

Next I simulate noisy data. I add random noise to the modeled wavefield and apply interferometry to retrieve the wavefield between  $r_A$  and  $r_B$ . The results are shown in Figure I.4.3. For simplicity, I retrieve wavefield from 0.9 to 4.0s to distinguish between direct and reflected waves. The time of the vertical axes corresponds to that of Figure I.4.2. I retrieve wavefield  $u(r_A, r_B)$  using data with various signal-to-noise ratios. The leftmost trace in each plot is of noise-free data so the signal-to-noise ratio is infinite. The signal-to-noise ratio of the second trace from the left is 20, which means that I added 5% noise, while that of the third trace is 19 and that of the fourth trace is 18. The amount of noise gradually increases until the rightmost trace's signal-to-noise ratio is 1, which means that I have added a noise volume equal to that of the signal. Figures I.4.3(a), I.4.3(b), and I.4.3(c) show the wavefields retrieved from cross-correlation, deconvolution, and cross-coherence, respectively. The wave at about 1.3s is a direct wave between  $r_A$  and  $r_B$ , and the wave at about 2.5s is a reflected wave from the boundary whose at a depth is 2,500m. When the signal-to-noise ratio is low, toward the right side of the figures, cross-correlation (Figure I.4.3(a)) shows a lot of noise. For the other two methods, however, the noise is not as apparent. On the other hand, I focus on signals that are direct and reflected waves. The amplitude in cross-correlation remains the same in each trace, but in the others it decreases. These properties correspond to features that are obtained from equations I.3.6 - I.3.8. Comparison of deconvolution with cross-coherence shows that cross-coherence is less noisy

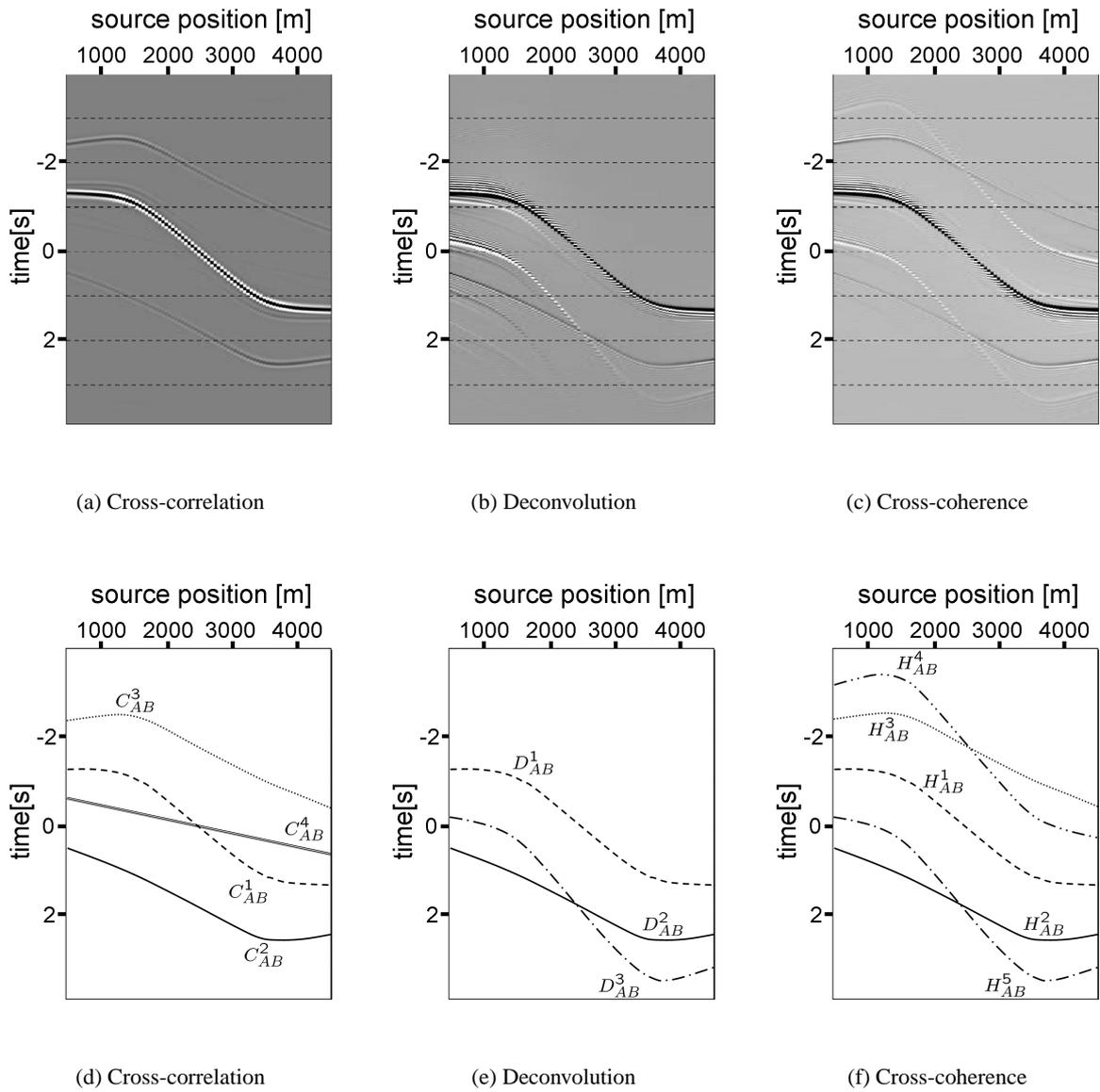
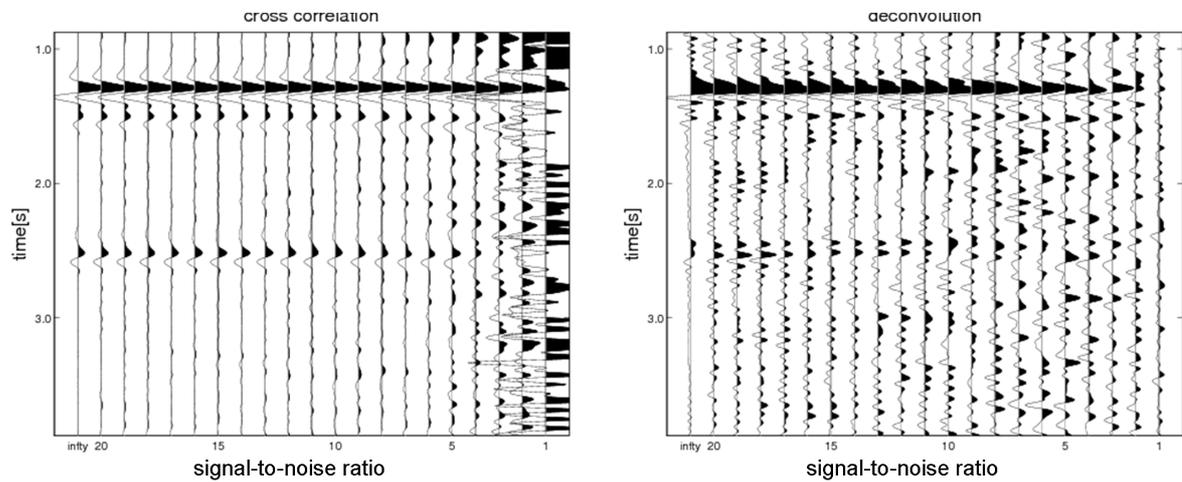


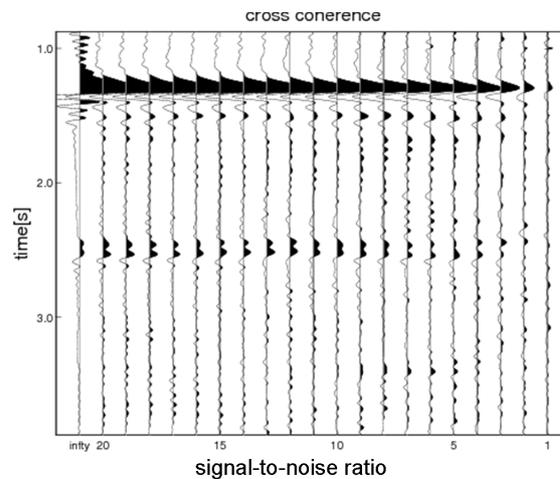
Figure I.4.2: Interferometry gathers constructed by interferometry using the of (a) cross-correlation, (b) deconvolution, and (c) cross-coherence methods between  $r_A$  and  $r_B$  in simulated data and the arrival times of the wavefields for (d) cross-correlation, (e) deconvolution, and (f) cross-coherence. The labels "C", "D", and "H" in these figures correspond to equations I.2.5, I.2.8 and I.2.14, respectively.

than deconvolution, as predicted from equation I.3.21. Moreover, the signal amplitude decreases in cross-coherence faster than in deconvolution because of the bias in equation I.3.14. Thus, when I apply cross-coherence interferometry to noisy data, I get the clearest result. Note that the absolute value of the amplitude in Figure I.4.3(a) is different from those in Figures I.4.3(b) and I.4.3(c), which exhibit the same amplitudes.



(a) Cross-correlation

(b) Deconvolution



(c) Cross-coherence

Figure I.4.3: The effect of noise added to the simulation data before applying (a) cross-correlation, (b) deconvolution, and (c) cross-coherence interferometry. One trace of these figures is made from the stack to source position of Figure I.4.2. The time range is from 0.9 to 4.0s in Figure I.4.2. In each figure, the signal-to-noise ratio is different between traces. No noise is added to the leftmost trace. The second trace from the left has a signal-to-noise ratio of 20. From this trace, the signal-to-noise ratio is reduce by one for each successive trace. The rightmost trace's signal-to-noise ratio is 1g

## Chapter I.5

### Case Study

I applied the cross-coherence method to real traffic-noise data acquired in Gunma Prefecture, Japan. An aerial photograph of the observation site is shown in Figure I.5.1(a). The survey line (blue line in Figure I.5.1(a)) was nearly linear, paralleling the river, and several roads and railways are crossed or paralleled the line that are shown by solid arrows. The length of the survey line was about  $2km$  with geophones at  $10m$  intervals. I made 1,200 records, each with a data length of  $30s$ , at a  $250Hz$  sampling frequency. Figure I.5.1(b) shows an example of a noise record along the entire line. Higher levels of traffic noises came from crossing roads and railways at receiver numbers 20, 50, 180 and 200. Since the source wavelets of the traffic noises had wide-ranging and complex frequency spectra, it was often difficult to estimate them.

If  $D(\omega)$  is the original trace in the frequency domain and its real and imaginary parts are  $\alpha(\omega)$  and  $\beta(\omega)$ , respectively, I get

$$\begin{aligned} D(\omega) &= \alpha(\omega) + i\beta(\omega) \\ &= \sqrt{\alpha^2(\omega) + \beta^2(\omega)} \left( \frac{\alpha(\omega) + i\beta(\omega)}{\sqrt{\alpha^2(\omega) + \beta^2(\omega)}} \right). \end{aligned} \quad (I.5.1)$$

$\sqrt{\alpha^2(\omega) + \beta^2(\omega)}$  is amplitude of  $D(\omega)$  and is equal to  $|D(\omega)|$ . Figure I.5.2 shows the observed and normalized data in the time and frequency domains. The frequency component of the observed data

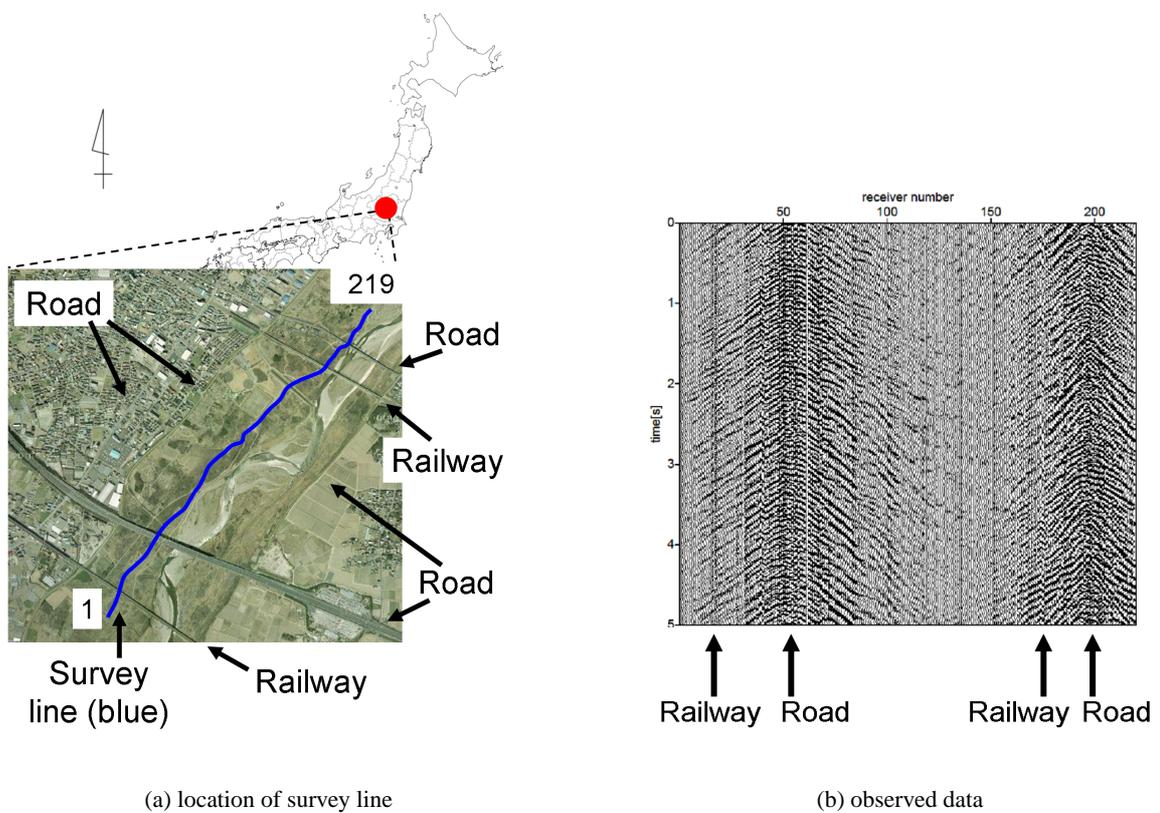


Figure I.5.1: (a)The location of survey line. I obtained the traffic noise data at Gunma, Japan. There are some crossing and parallel roads and trains. The survey line is almost straight along the river. (b)Observed gather. I set the receiver number from south to north.

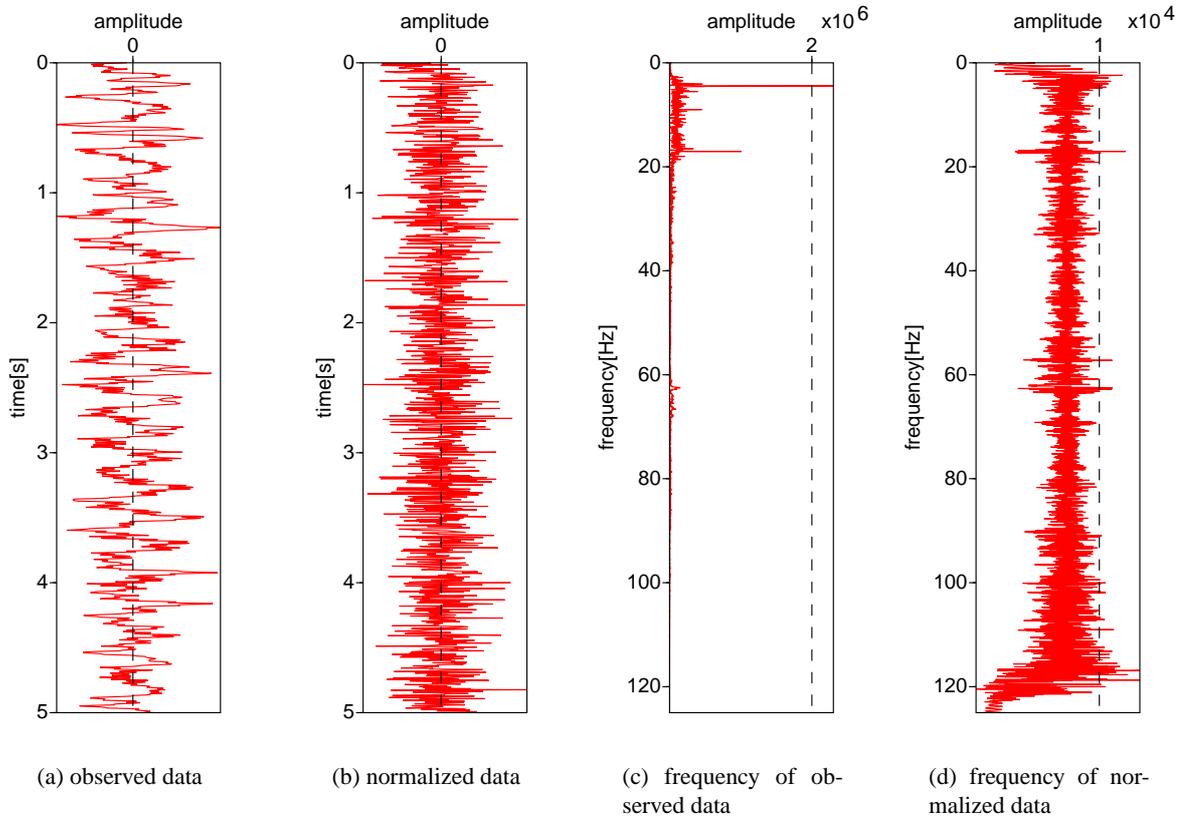


Figure I.5.2: (a), (c) Observed data in time and frequency domain, respectively. (b), (d) normalized data in time and frequency domain, respectively. (b) shows  $D(\omega)$  and (d) shows  $\frac{D(\omega)}{|D(\omega)|}$ .

(Figure I.5.2(c)) has some peaks, whereas that of the normalized data (Figure I.5.2(d)) is smooth and covers almost the full range of the spectrum up to the Nyquist frequency. The cross-coherence method convolves two traces after applying frequency normalization.

Figure I.5.3 show the pseudo shot gathers derived through the procedures of cross-correlation, deconvolution, and cross-coherence. Trace data at shot point 60 are used as the reference, and all data from the 1,200 records are stacked. No other filter is not applied to the records to construct these interferometry profiles. In cross-correlation (Figure I.5.3(a)), ringing noises are dominant due to the cyclic characteristics of the source wavelets of the traffic noise and amplitude levels are higher at the positions of noise sources (roads and railways). While the ringing noises are suppressed in the case of deconvolution as shown in Figure I.5.3(b), the signal-to-noise ratio is not very high and larger local amplitude levels still remain. Among these methods, cross-coherence (Figure I.5.3(c)) gives the best

results from the point of view of signal-to-noise ratio and trace balance. I can clearly see hyperbolic-like events that suggest reflections at around 0.4 and 1.3s.

I performed seismic reflection data processing using the CMP stack method and applying migration of each data sets from each of these three methods. I compared the results among the three interferometry methods and active shot gather (Figure I.5.4). After making pseudo shot gathers by interferometry, I applied identical processes of bandpass filter and NMO correction by same velocity to shot gathers. Figures I.5.4 (a) - (c) show the migration sections using pseudo shot records derived through cross-correlation, deconvolution, and cross-coherence, respectively. For cross-correlation, because of source wavelets from the crossing traffic, I obtained locally strong waves that obstructed the imaging of the structure. For deconvolution, although I could cancel out the source wavelet terms, the section was also noisy. I could recognize a faint similar structure in the shallow area of the CMP range, but I observed a lot of noise that hid most of the structure. It is obvious that cross-coherence interferometry gave the best result among the three methods. A reference seismic section using active seismic sources along the same line. The change of the dip of the reflections seen in cross-coherence section around 800m Horizon Distance is similar to that seen in active source section.

It is difficult to estimate the deep structure because the traffic noise level is not so strong enough, but I can increase the investigation depth of the interferometry down to about 300m by using the cross-coherence method. Although the investigation depth is limited, delineation of shallow structures down to 300m is useful for ground motion prediction by seismic monitoring for earthquake disaster prevention and basement surveys for civil engineering. Relative amplitudes are not preserved in the case of cross-coherence, but it is advantageous for the delineation of underground structures when the coherent noise level is low.

I now apply this improved interferometry technique to shear wave velocity estimation. I often observe clear surface waves in records of experiments, such as can be seen in Figure I.5.5(a), which is not surprising because because the theory of interferometry does not focus only on body waves. Figures I.5.5(b) and I.5.5(c) show the phase velocity and dispersion curve from the pseudo shot record

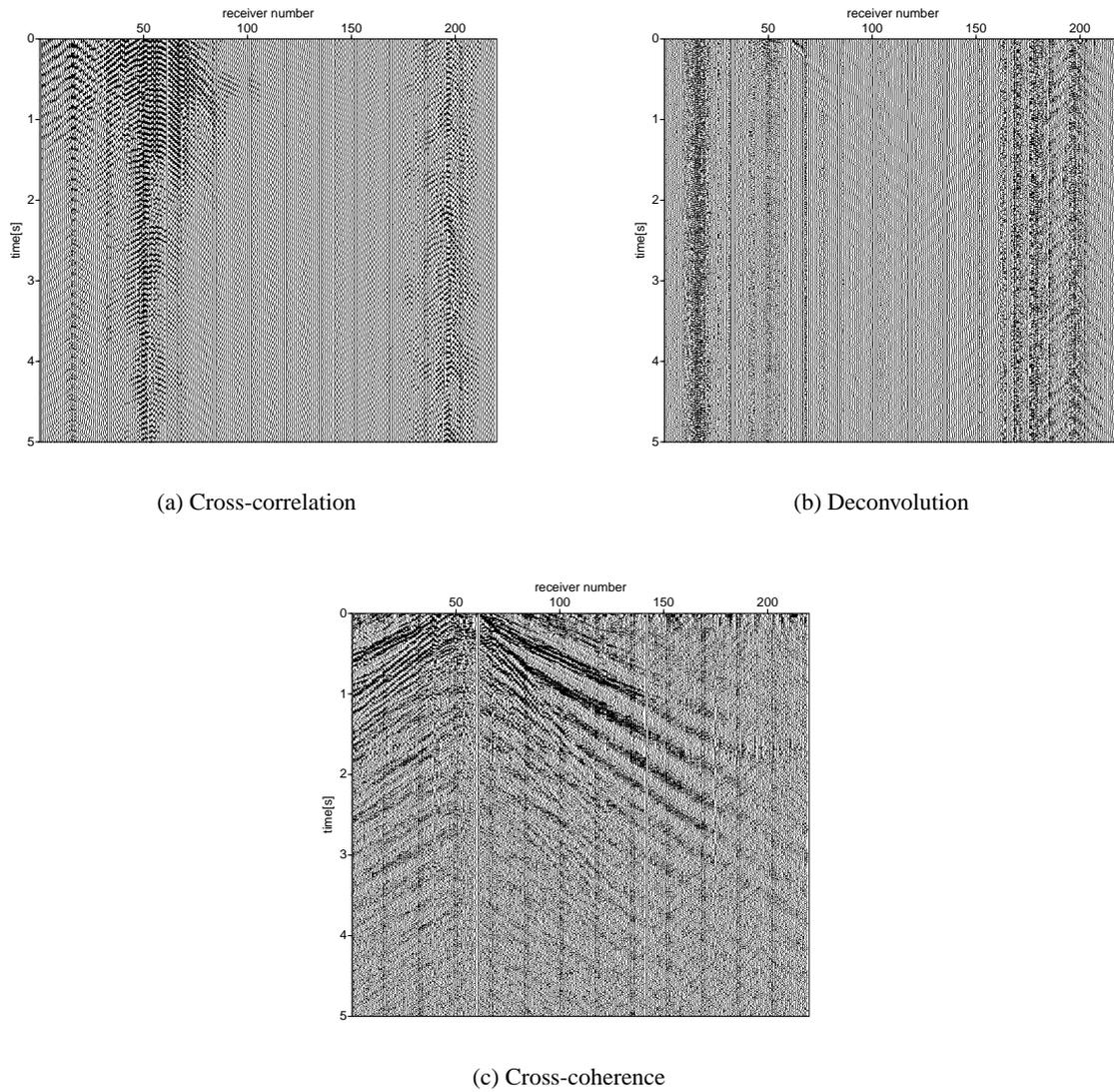


Figure I.5.3: Pseudo shot gathers made by interferometry. (a)cross-correlation, (b)deconvolution, and (c)cross-coherence. The source point of these gathers is receiver number 60. I do not apply some filters to these displayed data.

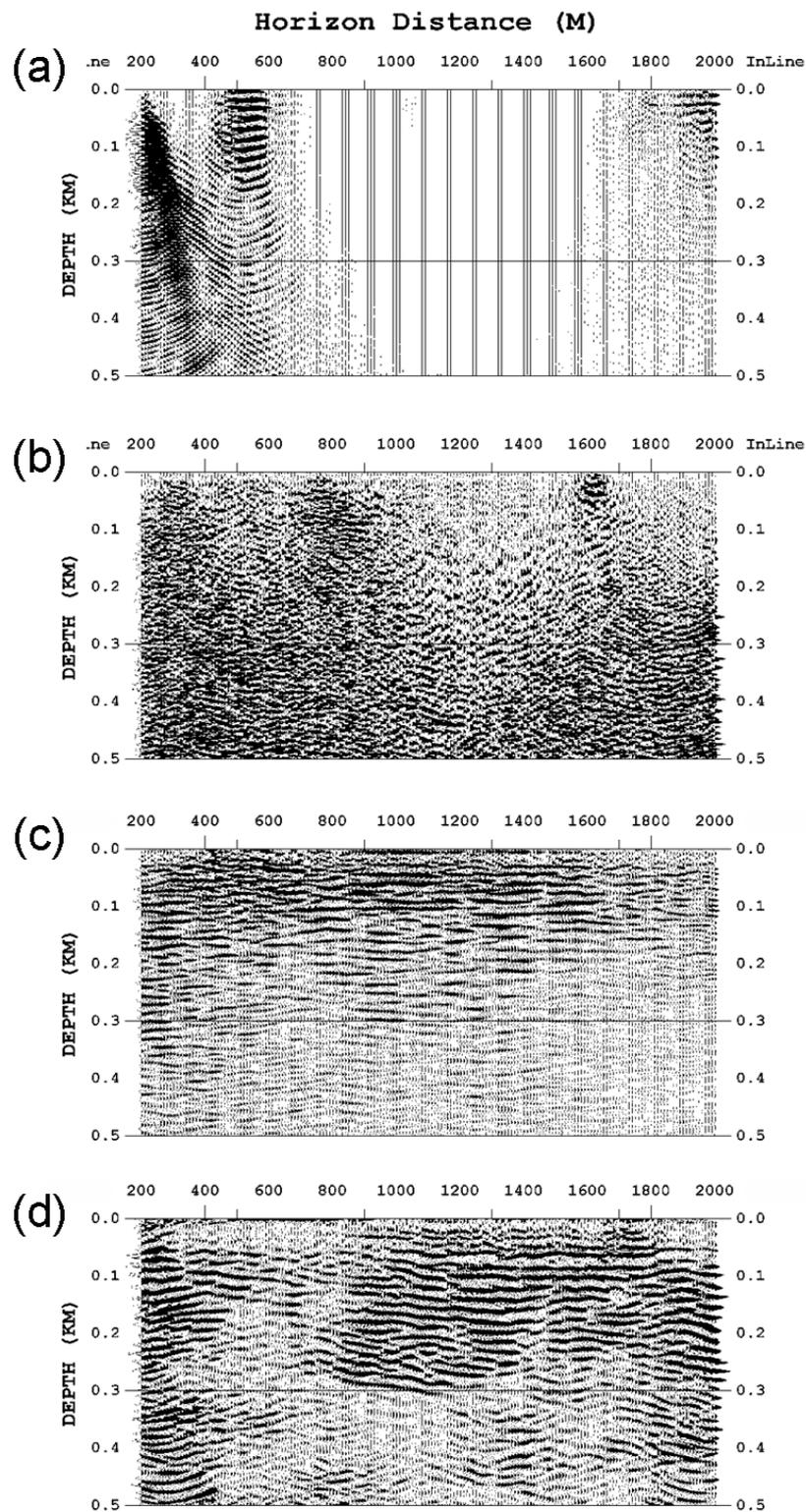
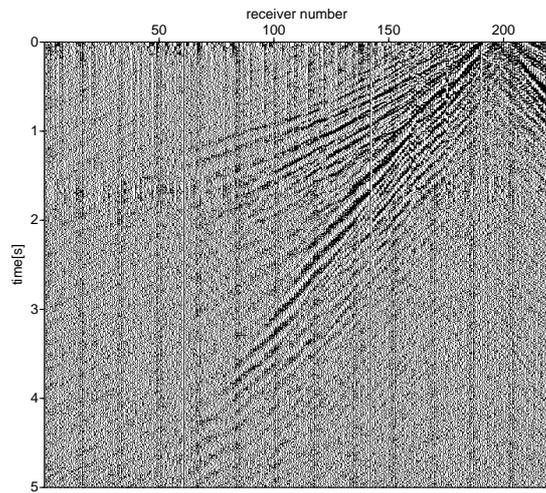


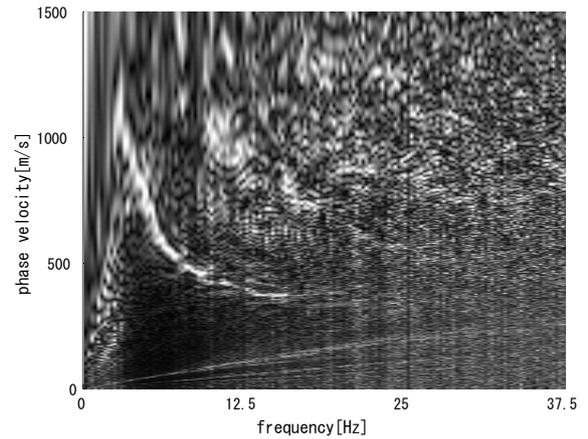
Figure I.5.4: Subsurface structure made by CMP stack method and migration using reflection wave. (a)cross-correlation, (b)deconvolution (c)cross-coherence interferometry and (d) is made by using active source. These figures are applied 3-5-35-40 Hz band pass filter.

at receiver number 190. The dispersion curve is estimated with sufficient resolution for applying Generic Algorithm (GA) inversion. GA inversion by (Yamanaka and Ishida, 1995) was used to pick out the phase velocity distribution from this dispersion curve. The blue crosses in Figure I.5.5(c) are the points of the fundamental mode of the surface waves chosen by an automatic picking system that determines the picking points from previous picking points, and the red line is the result of the GA inversion. The shear wave interval velocity distribution down to around 200m is estimated (I.5.5(d)). Because Figure I.5.5(a) includes a clear surface wave, I can obtain a clear phase velocity section.

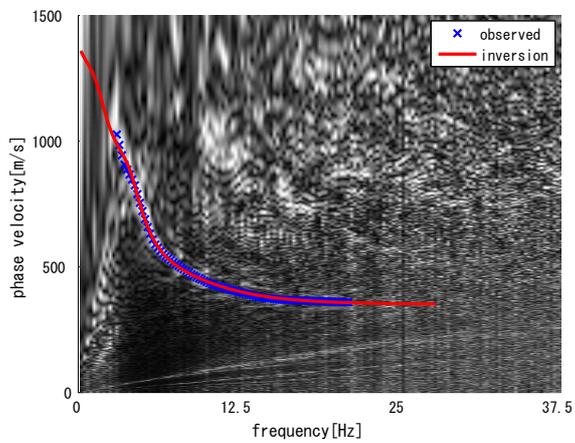
Figure I.5.6 shows a short-time Fourier transform section, which helps me understand why cross-coherence is suitable for surface wave analysis. For shear wave velocity analysis, I do not depend on the amplitude and using only the phase of each wave and a wide range of frequency response is necessary to analyze shallow to deep portions of the subsurface. From Figure I.5.6, it is obvious that cross-coherence gives the widest frequency range. Because random noise has a wide frequency range, the values from deconvolution (Figure I.5.6(b)) seems high. Not only the wide frequency range by random noise in deconvolution method is, however, not useful for analyzing the dispersion curve but it disturbs the analysis because random noise adds random values to the phase velocity section and moves the maximum value point. Thus, I might pick the wrong point in the phase velocity for calculating the dispersion curve. The mainly wide frequency range of cross-coherence is not dependent on random noise, and I can easily pick the dispersion curve by using the automatic picking system. Figure I.5.7 shows the maximum values of each frequency in the phase velocity section. The frequency range is 0 – 15Hz, which is narrower than that shown in Figure I.5.5(b). This figure compares values obtained by the three methods. Some values on the right side move at a faster velocity because of higher mode influence. The important point to note is that the cross-coherence method picks the lowest frequency. I can recognize that this is because of the wide frequency range, as shown in Figure I.5.6(a).



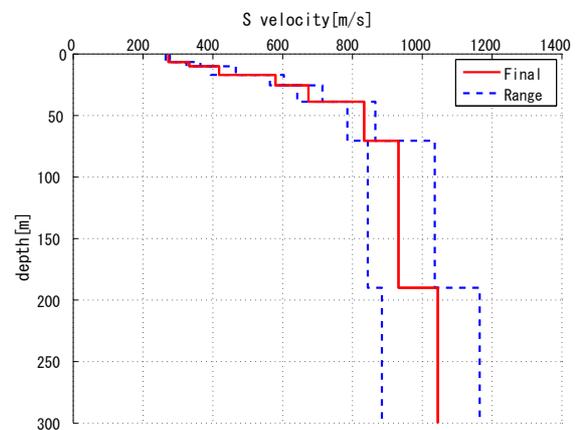
(a) pseudo shot gather



(b) phase velocity



(c) dispersion curve



(d) interval velocity

Figure I.5.5: (a)pseudo shot gather that are made by cross-coherence interferometry and that source point is receiver number 190. (b)phase velocity, (c)dispersion curve and (c)interval velocity that are made by (a). In (c), the blue  $\times$  is picking point of fundamental mode of surface wave and the red line is the dispersion curve from inversion. In (d), this interval velocity is estimated by inversion. The red thickness line is the estimated velocity and the blue dashed lines are the range of error.

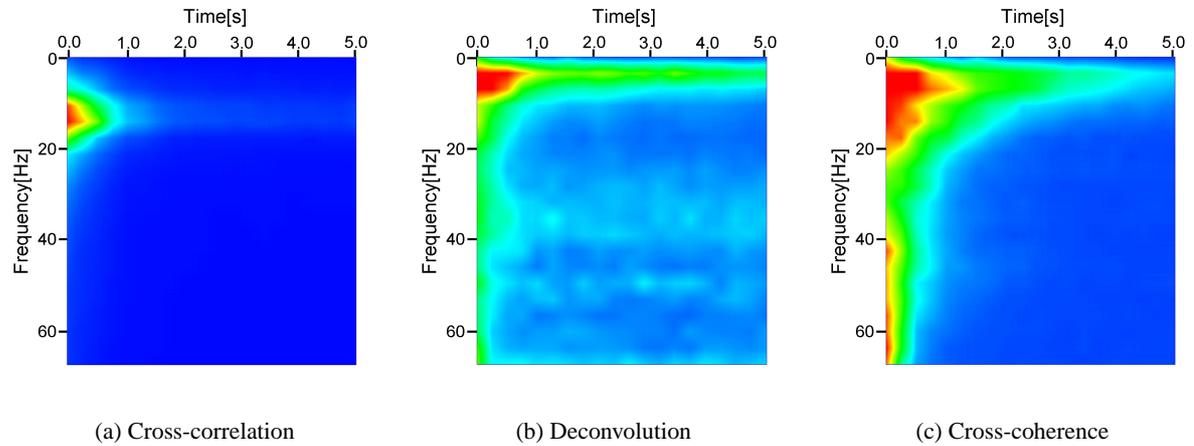


Figure I.5.6: Short time window FFT section from pseudo shot gathers by (a)cross-correlation, (b)deconvolution, and (c)cross-coherence interferometry. Transverse axis is in time domain, and vertical axis is in frequency domain. These values are normalized by maximum value of these section.

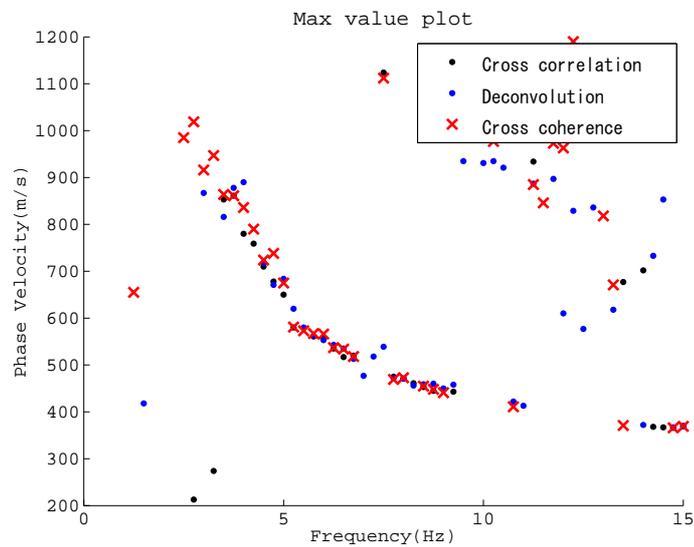


Figure I.5.7: Phase velocity section. I compare max amplitude values between three methods in each frequency range. Black, blue and red points are picked from cross-correlation, deconvolution and cross-coherence sections, respectively.

## Chapter I.6

# Conclusion

By using a Taylor expansion and separating the unperturbed and scattered waves, I can understand the wavefields included in the cross-coherence method and their difference in the cross-correlation and deconvolution methods. From a noise-added equation, cross-coherence can retrieve the phase of the Green's function that reduces the effect of noise. The properties of the equations were also shown in a 2D FDTD simulation. Furthermore, cross-coherence did not have to be added white noise to obtain stable results. In an analysis of real data, cross-coherence was shown to be the best method: its pseudo shot gather is clearer than those in the other methods. Cross-coherence uses only phase information in the frequency domain. However, the amplitude can change easily, for example, because of the sensitivity of the receivers or due to a surface structure. Because of releasing from consideration and applying additional works to reducing these effects, retrieving only the phase is sometimes better. Because of the normalization employed, the method repairs amplitude variations among traces. However, because amplitude information is lost, the method is not appropriate for some types of analysis, for example, of the reflection coefficient. The best method should be selected according to the data. I should retain as much information as I can. Thus, if the source wavelet is simple and the data have a high signal-to-noise ratio, I should choose cross-correlation. This method is easy and it preserves the amplitude in ordinary source and receiver locations. Cross-coherence is suitable for data that are noisy, that vary in amplitude among traces, or have long and complex source wavelets. The ultimate choice of the al-

gorithm to employ depends on the complexity of the excitation function, the amount of additive noise, and the amplitude variation among traces.

## **Appendix**

# **Higher order terms in the cross-coherence method**

Usually, the relationship between the unperturbed wave and the scattered wave satisfies equation I.2.2. If this is the case, I can ignore terms of the scattered wave that are higher than first order. However, higher orders are useful for better understanding of the cross-coherence method. If I include

all terms on the right-hand side of equation I.2.13 and inserting them into equation I.2.9, then I obtain

$$\begin{aligned}
 H_{AB} = & \frac{G_0(r_A, s)G_0^*(r_B, s) + G_s(r_A, s)G_0^*(r_B, s) + G_0(r_A, s)G_s^*(r_B, s) + G_s(r_A, s)G_s^*(r_B, s)}{|G_0(r_A, s)||G_0(r_B, s)|} \\
 & \times \left[ 1 + \sum_{n=1}^{\infty} \left\{ \frac{1}{n!} \prod_{i=1}^n \left( \frac{1}{2} - i \right) \right. \right. \\
 & \quad \times \sum_{a=0}^n \sum_{b=0}^a \frac{n!}{(n-a)!(a-b)!b!} \underbrace{\left( \frac{G_0(r_A, s)G_s^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{n-a} \left( \frac{G_s(r_A, s)G_0^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{a-b} \left( \frac{|G_s(r_A, s)|^2}{|G_0(r_A, s)|^2} \right)^b}_{(1)} \left. \right\} \\
 & + \sum_{n=1}^{\infty} \left\{ \frac{1}{n!} \prod_{i=1}^n \left( \frac{1}{2} - i \right) \right. \\
 & \quad \times \sum_{a=0}^n \sum_{b=0}^a \frac{n!}{(n-a)!(a-b)!b!} \underbrace{\left( \frac{G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{n-a} \left( \frac{G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{a-b} \left( \frac{|G_s(r_B, s)|^2}{|G_0(r_B, s)|^2} \right)^b}_{(2)} \left. \right\} \\
 & + \sum_{n=1}^{\infty} \sum_{m=1}^{\infty} \left\{ \frac{1}{n!} \frac{1}{m!} \prod_{i=1}^n \left( \frac{1}{2} - i \right) \prod_{j=1}^m \left( \frac{1}{2} - j \right) \sum_{a=0}^n \sum_{b=0}^a \sum_{c=0}^m \sum_{d=0}^c \frac{n!}{(n-a)!(a-b)!b!} \frac{m!}{(m-c)!(c-d)!d!} \right. \\
 & \quad \underbrace{\left( \frac{G_0(r_A, s)G_s^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{n-a} \left( \frac{G_s(r_A, s)G_0^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{a-b} \left( \frac{|G_s(r_A, s)|^2}{|G_0(r_A, s)|^2} \right)^b}_{(3)} \\
 & \quad \left. \underbrace{\left( \frac{G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{m-c} \left( \frac{G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{c-d} \left( \frac{|G_s(r_B, s)|^2}{|G_0(r_B, s)|^2} \right)^d}_{(3)} \right\}. \tag{A.1}
 \end{aligned}$$

Equation A.1 is complex. First I extract expressions (1), (2), and (3) as labeled in this equation and expand them separately. These parts and the first line of equation A.1 determine the phase of the wavefield of  $H_{AB}$ . In expression (1), I consider the two situations of  $n - a \geq a - b$  and  $n - a < a - b$ .

Then

$$n - a \geq a - b$$

$$\left( \frac{G_0(r_A, s)G_s^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{n-2a+b} \left( \frac{|G_s(r_A, s)|^2}{|G_0(r_A, s)|^2} \right)^{a-b} \quad (\text{A.2a})$$

$$n - a < a - b$$

$$\left( \frac{G_s(r_A, s)G_0^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{2a-b-n} \left( \frac{|G_s(r_A, s)|^2}{|G_0(r_A, s)|^2} \right)^{n-a}. \quad (\text{A.2b})$$

The phase of expression A.2b is  $(G_s(r_A, s)G_0^*(r_A, s))^{2a-b-n}$ . This term means that waves bounce between  $r_A$  and  $x_s$  for  $2a - b - n$  times (Figure I.A.1(a)) and  $r_A$  behaves as a free-point boundary. The wave propagation of expression A.2a is the acausal part of expression A.2b. Expression (2) is similar to expression (1) with just a changes from  $r_A$  to  $r_B$ . Thus, these waves bounce between  $r_B$  and  $x_s$ .

Next, to analyze expression (3) I have to consider eight situations. For simplification, I do not show the absolute value terms in the numerator because these terms do not contribute to the phase of the wavefield. These terms are  $\left( \frac{|G_s(r_A, s)|^2}{|G_0(r_A, s)|^2} \right)^M \left( \frac{|G_s(r_B, s)|^2}{|G_0(r_B, s)|^2} \right)^N$ , in which  $M$  and  $N$  are replaced by appropriate

values. Thus, I obtain

$$n - a < a - b, m - c < c - d \text{ and } 2a - b - n > 2c - d - m$$

$$\left( \frac{G_s(r_A, s)G_0^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{m-2c+d} \left( \frac{G_s(r_A, s)G_0^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{2a-b-n-(2c-d-m)} \quad (\text{A.3a})$$

$$n - a > a - b, m - c < c - d \text{ and } n - 2a + b > 2c - d - m$$

$$\left( \frac{G_0(r_A, s)G_s^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{2c-d-m} \left( \frac{G_0(r_A, s)G_s^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{n-2a+b-(2c-d-m)} \quad (\text{A.3b})$$

$$n - a < a - b, m - c > c - d \text{ and } 2a - b - n > m - 2c + d$$

$$\left( \frac{G_s(r_A, s)G_0^*(r_A, s)G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{m-2c+d} \left( \frac{G_s(r_A, s)G_0^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{2a-b-n-(m-2c+d)} \quad (\text{A.3c})$$

$$n - a > a - b, m - c > c - d \text{ and } n - 2a + b > m - 2c + d$$

$$\left( \frac{G_0(r_A, s)G_s^*(r_A, s)G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{m-2c+d} \left( \frac{G_0(r_A, s)G_s^*(r_A, s)}{|G_0(r_A, s)|^2} \right)^{n-2a+b-(m-2c+d)} \quad (\text{A.3d})$$

$$n - a < a - b, m - c < c - d \text{ and } 2a - b - n < 2c - d - m$$

$$\left( \frac{G_s(r_A, s)G_0^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{2a-b-n} \left( \frac{G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{2c-d-m-(2a-b-n)} \quad (\text{A.3e})$$

$$n - a > a - b, m - c < c - d \text{ and } n - 2a + b < 2c - d - m$$

$$\left( \frac{G_0(r_A, s)G_s^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{n-2a+b} \left( \frac{G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{2c-d-m-(n-2a+b)} \quad (\text{A.3f})$$

$$n - a < a - b, m - c > c - d \text{ and } 2a - b - n < m - 2c + d$$

$$\left( \frac{G_s(r_A, s)G_0^*(r_A, s)G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{2a-b-n} \left( \frac{G_s(r_B, s)G_0^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{m-2c+d-(2a-b-n)} \quad (\text{A.3g})$$

$$n - a > a - b, m - c > c - d \text{ and } n - 2a + b < m - 2c + d$$

$$\left( \frac{G_0(r_A, s)G_s^*(r_A, s)G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_A, s)|^2|G_0(r_B, s)|^2} \right)^{n-2a+b} \left( \frac{G_0(r_B, s)G_s^*(r_B, s)}{|G_0(r_B, s)|^2} \right)^{m-2c+d-(n-2a+b)} \quad (\text{A.3h})$$

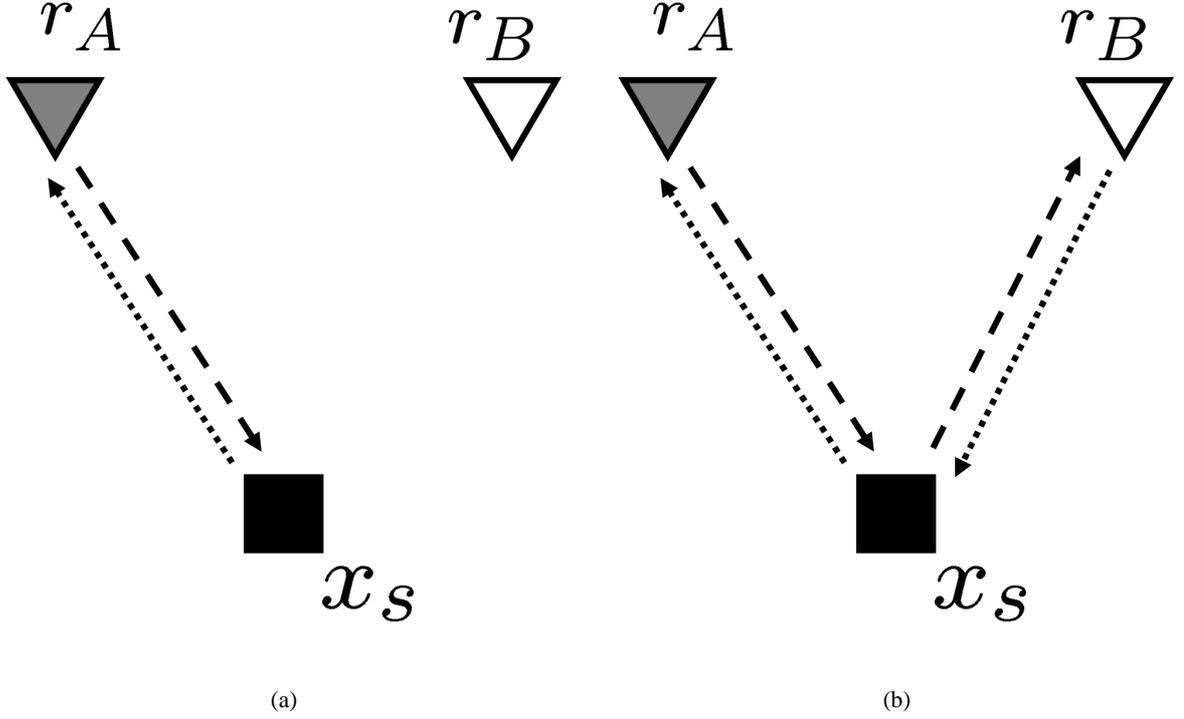


Figure I.A.1: (a)  $(G_s(r_A, s)G_0^*(r_A, s))^{2a-b-n}$ . This wavefield is excited at  $r_A$  and propagates to  $x_s$  (dashed arrow) and reflects at  $x_s$  and returns to  $r_A$  (dotted arrow). (b)  $\{G_s(r_A, s)G_0^*(r_A, s)G_s(r_B, s)G_0^*(r_B, s)\}^{m-2c+d}$ . This wavefield is excited at  $r_A$  and propagates to  $r_B$  through  $x_s$  (dashed arrow). And this wavefield reflects at  $r_B$  and returns to  $r_A$  through  $x_s$  again (dashed arrow).  $r_B$  behaves free-point boundary.

I transform the numerator of the left term of expression A.3a and A.3e to

$G_s(r_A, s)G_0^*(r_B, s)\{G_s^*(r_B, s)G_0(r_A, s)\}^*$ . Thus, this term shows the convolution between a wave that is excited at  $r_A$  and propagates to  $r_B$  via  $x_s$  and a wave that is excited at  $r_B$  and propagates to  $r_A$  via  $x_s$ . From this examination, the wavefield of this term is seen to shuttle between  $r_A$  and  $r_B$  through  $x_s$  for  $m - 2c + d$  times (Figure I.A.1). The terms of expression A.3d and A.3h are the acausal parta of this term. The other four terms describe the convolutions between the unperturbed wave and second-order scattered wave. The difference between expressions A.3b and A.3c is the source point of each wavefield. All these considerations combine to clarify the phases of the left term of each expression. The right term has the same phase as expression (1) or (2). Therefore, for example, expression A.3a describes the wavefield that shuttles between  $r_A$  and  $r_B$  via  $x_s$  for  $m - 2c + d$  times and shuttles between  $r_A$  and  $x_s$  for  $2a - b - n - (2c - d - m)$  times.

Thus, I have clarified the phases of expressions (1), (2), and (3) in equation A.1. All wavefields are made by the first line of equation A.1 or multiplication between this line and (1), (2), and (3).

## **Part II**

# **Indicators for survey designing for seismic time-lapse monitoring of $CO_2$ injection to subsurface aquifer**

## Chapter II.1

# Introduction

$CO_2$  injection to underground reservoir is one of hot topics as Carbon Dioxide Capture and Storage (CCS) all over the world. Although I certainly do reservoir simulation before injection in fact, simulation is not completely same as real world. Thus, when I want to inject  $CO_2$ , it is important to grasp the activity of  $CO_2$  in the reservoir after injection. I have some methods that could detect the  $CO_2$  activity; using seismic, electromagnetic, gravity, remote sensing and so on. Although I can choose or combine these methods, I think the most superiority method of monitoring is seismic exploration that has some achievements of deep structure in oil-production. Time-lapse monitoring is a powerful tools for detecting the changes in subsurface structure (Koster et al., 2000; Lumley et al., 2001). The purpose of monitoring is to identify the edge of  $CO_2$  and detecting the leak. I can get the good result of 4D seismic survey in foreign country (Arts et al., 2004). In Japan, however, it is not completely same as foreign country's case because Japanese geological features are usually very complex. Thus, I establish two models that are synthetic and realistic, and simulate the seismic wave response in 3D. In this part, I assume the reservoir is an aquifer and the depth of this is about 1,000 m from the bottom of the sea. The method of simulation is 3D Ray Tracing (Cerveny, 2001) because it takes shorter time than Finite-Difference method. I also change the properties of a small portion of model that is assumed a  $CO_2$  injection portion and simulate both before and after models for time-lapse seismic survey. The noise of P wave 4D seismic survey may be white noise, coherent noise that are some multiples, P-S

converted wave and S wave from reflectors that exist upper reservoir, and the difference of positions of sources and receivers whose noise is specific of time-lapse survey. In this part, I do not focus the specific noise of time-lapse survey because I think it is smaller contribution than white or coherent noise. I compare the differential between before and after data that are made by CMP stack or post-stack time migration that are changing the number of receivers or sources or the amount of noise. I want to reduce the number of receivers as possible as I can because I placed Ocean-Bottom Cable (OBC) semi-permanently. The semi-permanently placed OBC is suitable for monitoring by to reduce the cost and to reduce the noise. If I can reduce the number of receivers, this OBC is cheaper than ordinary OBC that puts down and pulls up every time. If I put semi-permanently OBC, I can reduce the one of specific noise of time-lapse survey that are difference of receiver position between two explorations.

## Chapter II.2

# Simulated data processing

I simulate two models that are synthetic and realistic. Synthetic model is symmetric and realistic model is built from three-dimensional seismic volume. I analyze these simulated data and compare the difference of result when I change the parameters of simulation.

### II.2.1 Ray tracing simulation using synthetic model

This model is idealistic anticline structure (Figure II.2.1) and size of this model is  $10 \times 10 \times 3km$  including 9 layers. The properties that are P and S wave velocity and density of each layer are constant (Table II.2.1). Figure II.2.1(b) shows the P wave velocity section at  $Y = 5km$ . I set  $CO_2$  injection point which is round shape between layer 5 and 6 (Figure II.2.2). I calculate the properties of  $CO_2$  injected portion using Gassmann theory in two cases that are 50% saturated (high saturated) and 2.4% saturated (low saturated).

I set sources that are  $5 \times 5km$  and receivers that are  $3 \times 3km$  in the center of model at  $z = 0km$ . The imaging range of CMP stack method is  $4 \times 4km$  at the center of model. I show the densest case of both sources and receivers (Pattern A) (Figure II.2.3). In the densest case, the interval of source and receiver points is  $25m$  and the interval of source and receiver lines is  $100m$ . Because this survey is really high density, I set other two cases. In Pattern B, sources are same position as Pattern A and the interval of

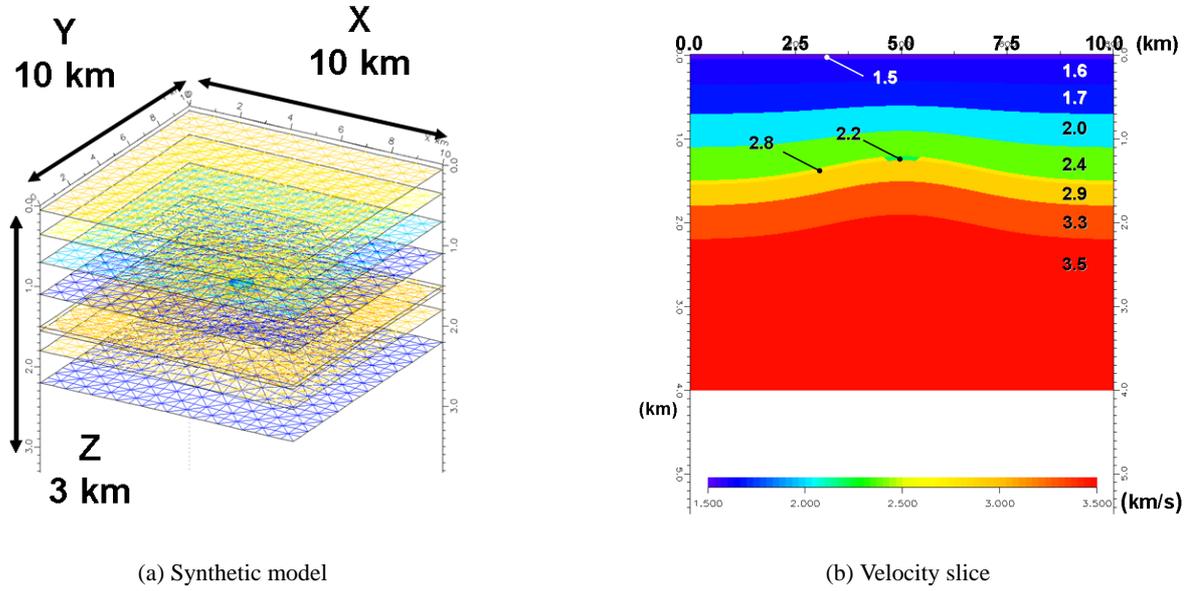


Figure II.2.1: (a) Synthetic model for ray tracing. The size of this model is  $10 \times 10 \times 3 \text{ km}$  and this has 9 layers. (b) P wave velocity slice at  $Y = 5 \text{ km}$ .

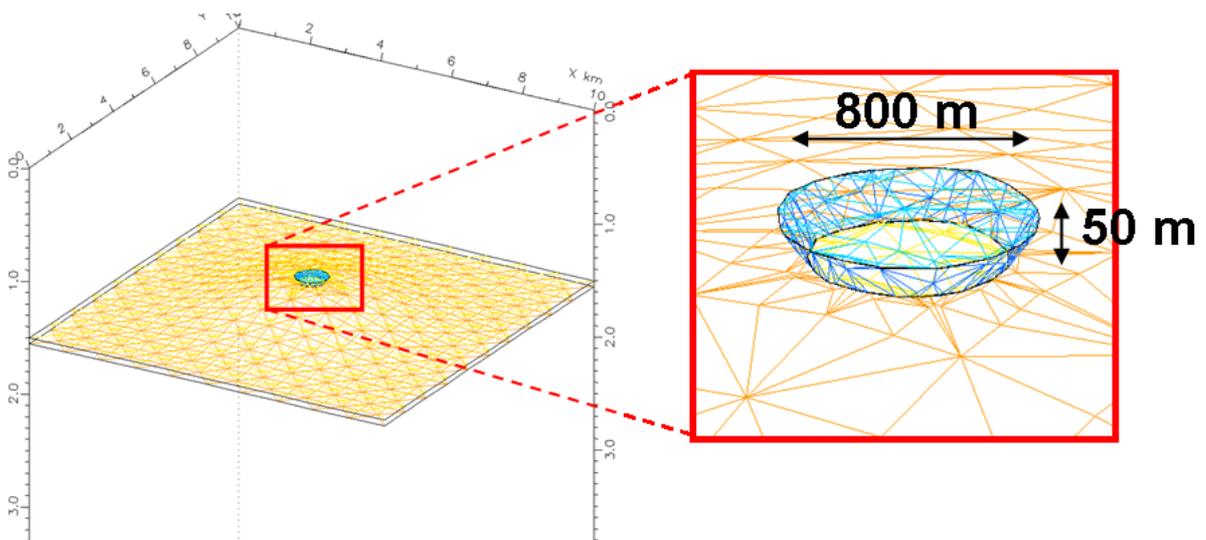


Figure II.2.2:  $\text{CO}_2$  injection portion. This is round shape and size is  $800 \times 50 \text{ m}$ . This portion is put between two layers.

Table II.2.1: The properties of synthetic model

Layer ID	depth (m)		velocity (m/s)		Density (kg/m <sup>3</sup> )	note
	col	top	P wave	S wave		
1	50	50	1500	0	1030	the bottom of the sea
2	350	300	1600	730	1800	
3	700	600	1700	810	2100	
4	1100	900	2000	960	2150	
5	1500	1200	2400	1200	2200	the top of the aquifer
6	1550	1250	2800	1455	2240	the bottom of the aquifer
7	1800	1500	2900	1600	2300	
8	2200	1900	3100	1750	2400	
9			3200	1820	2450	
			2200	1480	2180	CO <sub>2</sub> (the rate of saturation is 50%)
			2450	1456	2241	CO <sub>2</sub> (the rate of saturation is 2.4%)

receiver points is 100m and one of receiver lines is 300m. In Pattern C, receivers are same position as Pattern B and the interval of source points is 25m and one of source lines is 300m. I show the number of folds of each pattern in Figure (II.2.4).

Table II.2.2: The number of sources and receivers in synthetic model

	source	receiver
Pattern A	10,000	3,600
Pattern B	10,000	300
Pattern C	3,300	300

I simulate the seismic wave using Wavefront Reconstruction Method of Ray Tracing (2D; (Vinje et al., 1993), 3D; (Vinje et al., 1996a; Vinje et al., 1996b)). Although I calculate primary P waves only in this simulation, the amplitude of simulated wave is received the effect of S wave velocity and density like Amplitude-Variation-with-Offset (AVO). I show the shot gather that are received by all receivers in Pattern A (Figure II.2.5). I add some noises that are scattering and white noise to simulated gathers and compare 3 cases that are noise free, adding scattering noise and adding both scattering and white noises. Scattering noise means the variation of amplitude or time shift since the directly under bottom of the sea because this portion has strong inhomogeneity. I add twice amplitude white noise ( $S/N = 0.5$ ) from the amplitude of center of reflection wave from Layer 4. I assume this noise amplitude is maximum grade in actual seismic exploration. I show the CMP gathers in Pattern A after NMO correction whose

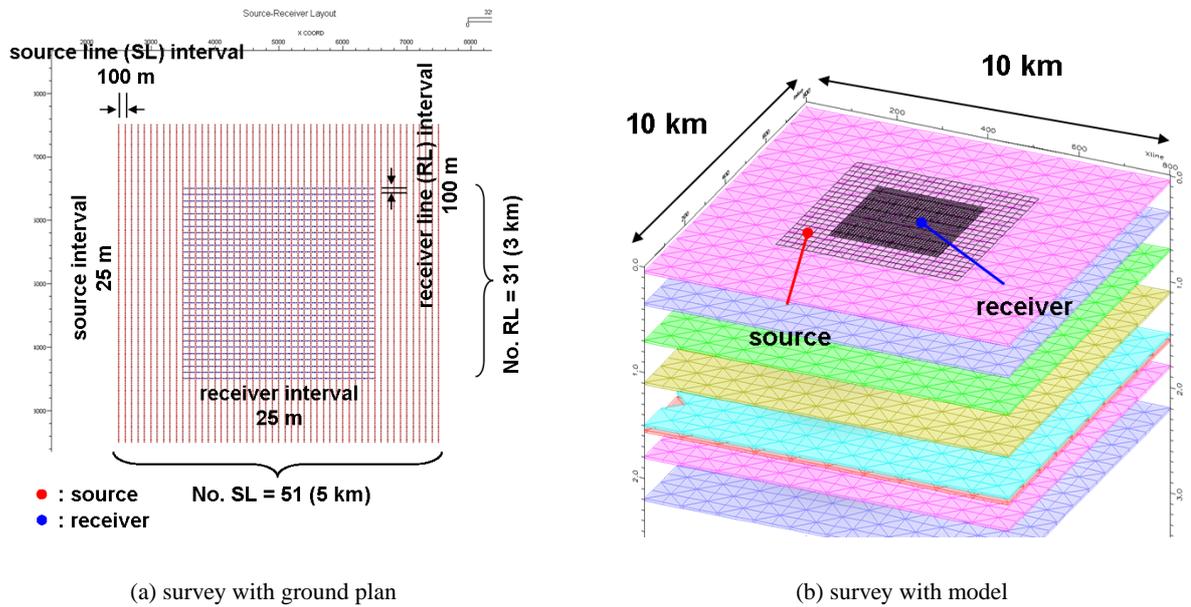


Figure II.2.3: Sources and Receivers. I show the densest pattern. (a)ground plan of sources and receivers. Red point is source and blue point is receiver. This survey takes orthogonal geometry and I can get a variety of offsets without near. (b)Survey and layers.

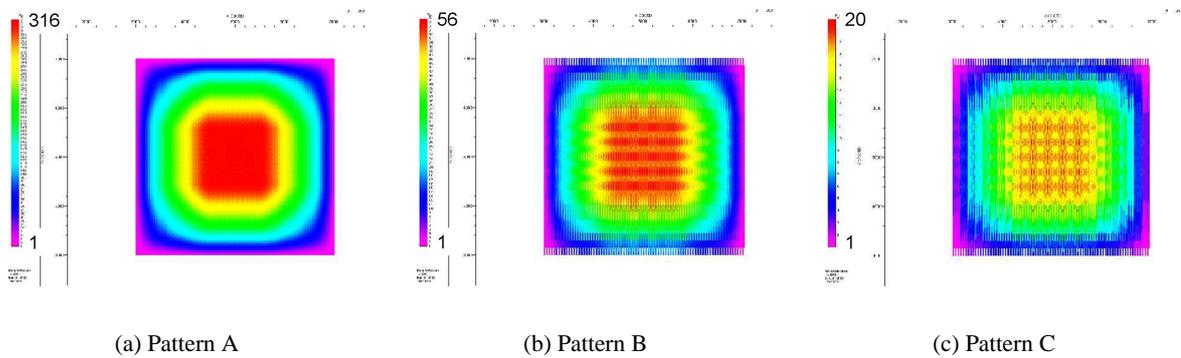


Figure II.2.4: The number of folds. The range of this square is  $4 \times 4 \text{ km}$  at the center of the model.

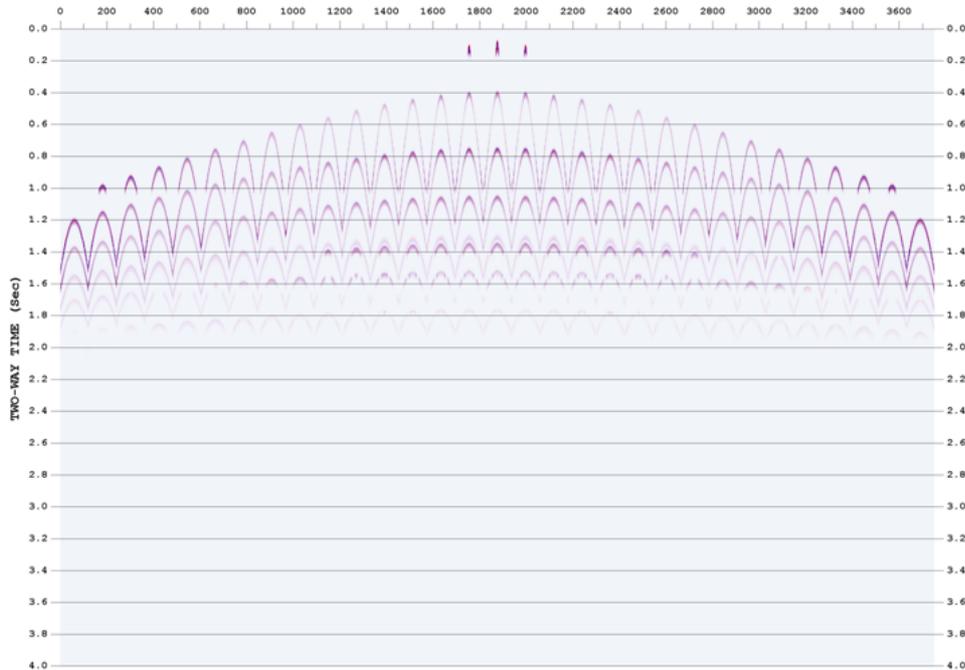


Figure II.2.5: Shot gather. The shot point is  $(x, y, z) = (5, 5, 0\text{km})$  and I show all receivers gather. Because I simulate only primary wave from each layer, I have no data after about  $2.0\text{sec}$ . This shot gather is made by ray tracing, this do not simulate surface wave or diffraction wave. The label of horizontal axis is receiver number and there are 121 receivers by each receiver line.

noise are difference (Figure II.2.6).

The analyzing process is CMP sort, amplitude correction of geometric attenuation, NMO (normal moveout) correction, DMO (dip moveout) (Hale, 1984), CMP stack and migration. CDP bin size is  $12.5 \times 12.5\text{m}$  in Pattern A and  $25 \times 12.5\text{m}$  in Pattern B and C. We show DMO section (Figure II.2.7) that compare the difference of variety of noise before migration and migration section (Figure II.2.8) that compare the effect of the number of sources and receivers, and one of noise. We can see that the reflection wave from  $\text{CO}_2$  injected portion is getting unclearly from Pattern A to C and from high saturated to low. We also show the RMS amplitude of  $32\text{ms}$  whose center is target horizon that are the top aquifer (Figure II.2.9). Because it is difficult to detect the range of the anomaly on the slice sections in 3D data, this horizon section is helpful. We can know the time of target horizon from the before injection section and calculate the RMS amplitude on the horizon section. It is really difficult to recognize the injection point in Pattern C, low saturated.

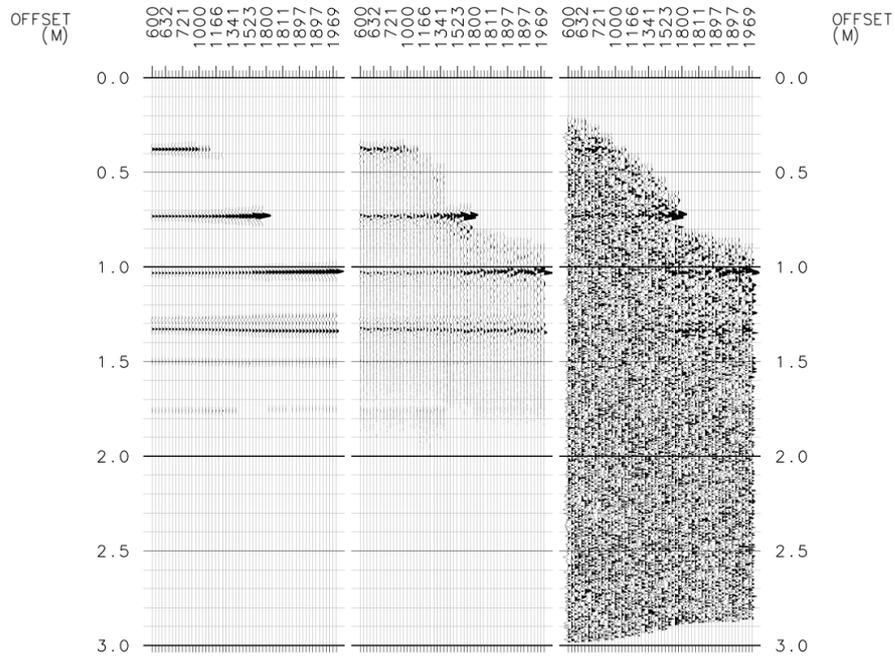


Figure II.2.6: CMP gather after NMO correction. Left figure is noise free, middle figure is added scattering noise and right figure is added both scattering and white noises.

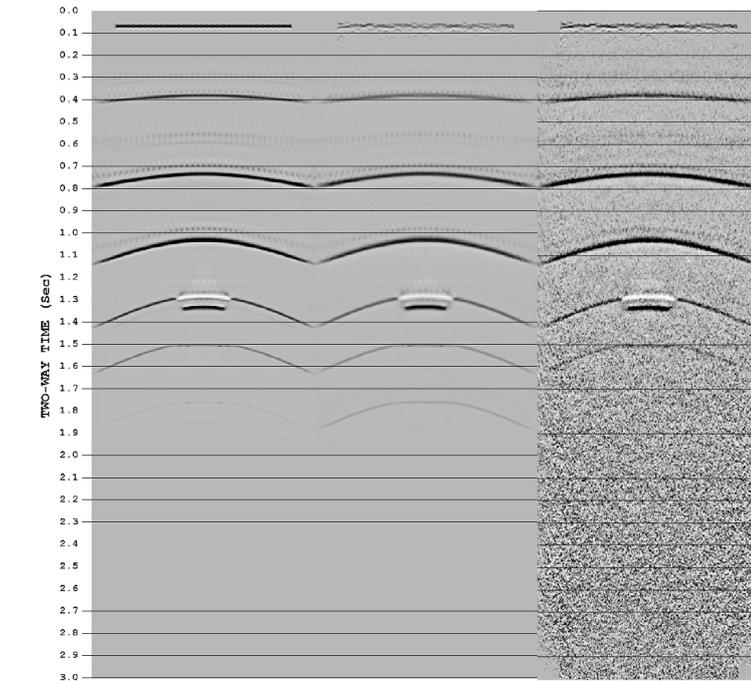


Figure II.2.7: DMO stack section of after high saturated  $CO_2$  injection. Left figure is noise free, middle figure is added scattering noise and right figure is added both scattering and white noises.

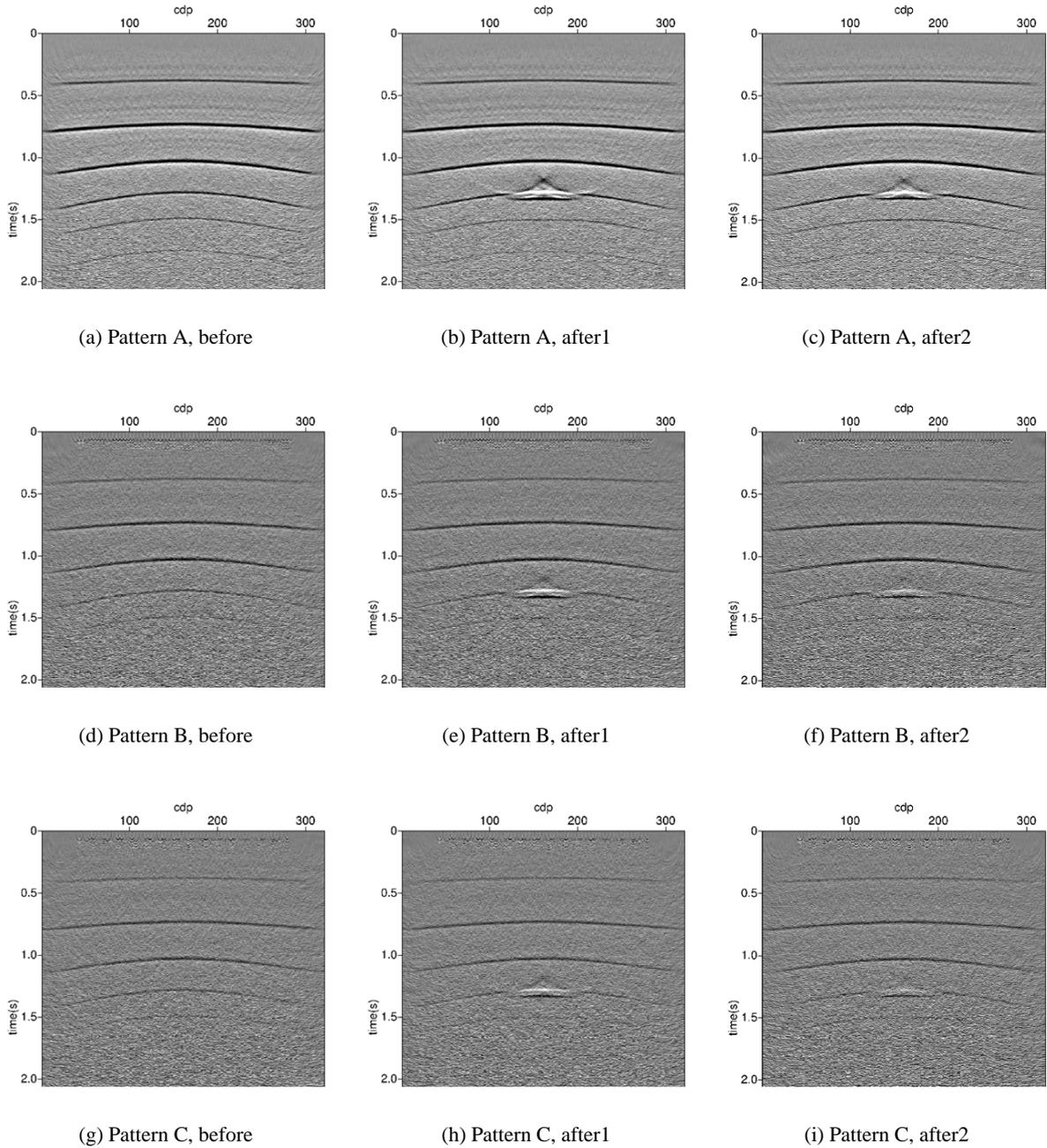


Figure II.2.8: Migration sections. before, after1 and after2 is before injection, after injection that are high saturated  $CO_2$  and after injection that are low saturated  $CO_2$ , respectively. (a), (b) and (c) densest survey (Pattern A). (d), (e) and (f) dense sources and sparse receivers (Pattern B). (g), (h) and (i) sparsest survey (Pattern C).

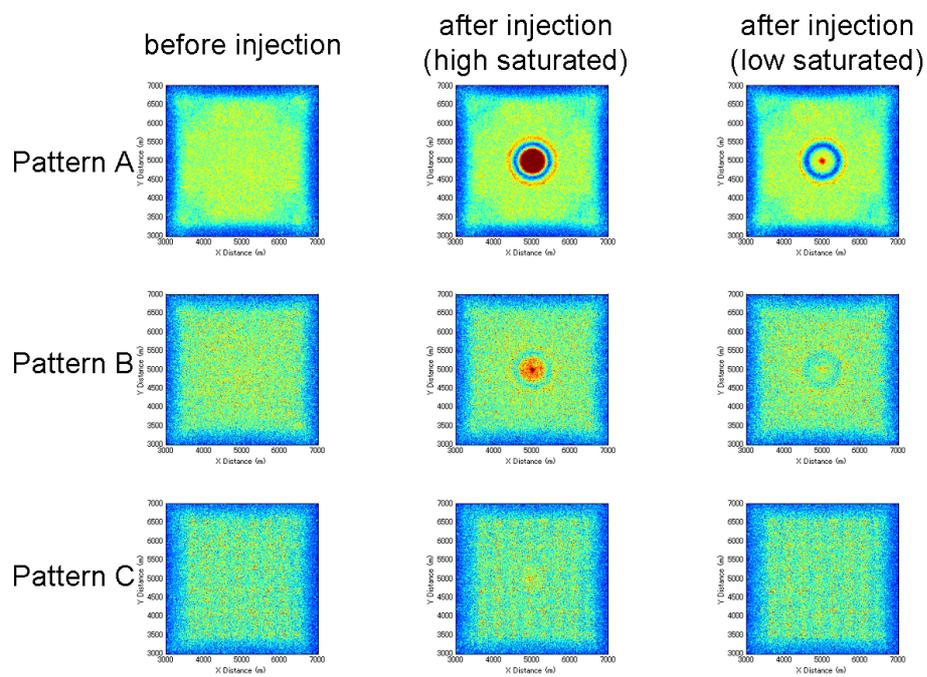


Figure II.2.9: The RMS amplitude that time gate is 32ms whose center is the top of aquifer. Top, middle and bottom line figures are Pattern A, B and C, respectively. Left, middle and right row figures are before  $CO_2$  injection, after high saturated  $CO_2$  injection, and after low saturated  $CO_2$  injection, respectively.

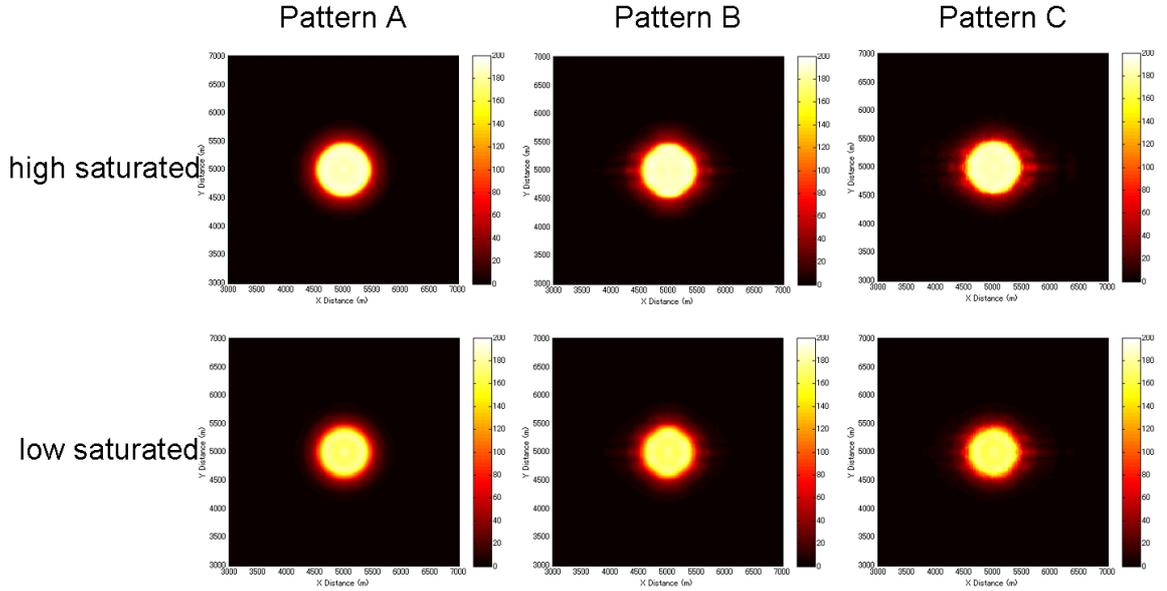


Figure II.2.10: NRMS of noise free data. White color shows 200 and black color shows 0. Top and bottom line figures are high and low  $CO_2$  saturated. Left, middle and right row figures are Pattern A, B and C.

We calculate the normalized rms difference (NRMS) (Kragh and Christie, 2002) between before and after injection section. NRMS is calculated

$$NRMS = \frac{200 \times RMS(a_i - b_i)}{RMS(a_i) + RMS(b_i)}. \quad (II.2.1)$$

$$RMS(x_i) = \sqrt{\sum_{t_1}^{t_2} (x_i)^2 / N} \quad (II.2.2)$$

$N$  is the number of samples from  $t_1$  to  $t_2$ . NRMS measures the difference of everything that include amplitude, phase, and time shift. The maximum and minimum value of NRMS is 200 and 0;  $NRMS = 200$  means two traces has opposite phase and  $NRMS = 0$  means completely same traces. We show the NRMS of noise free (Figure II.2.10) and added both noises (Figure II.2.11). Because NRMS is 141 in random noise traces, my result approaches to 141 when I add both noises. We also see it is difficult to recognize injection portion in sparse receivers' pattern.

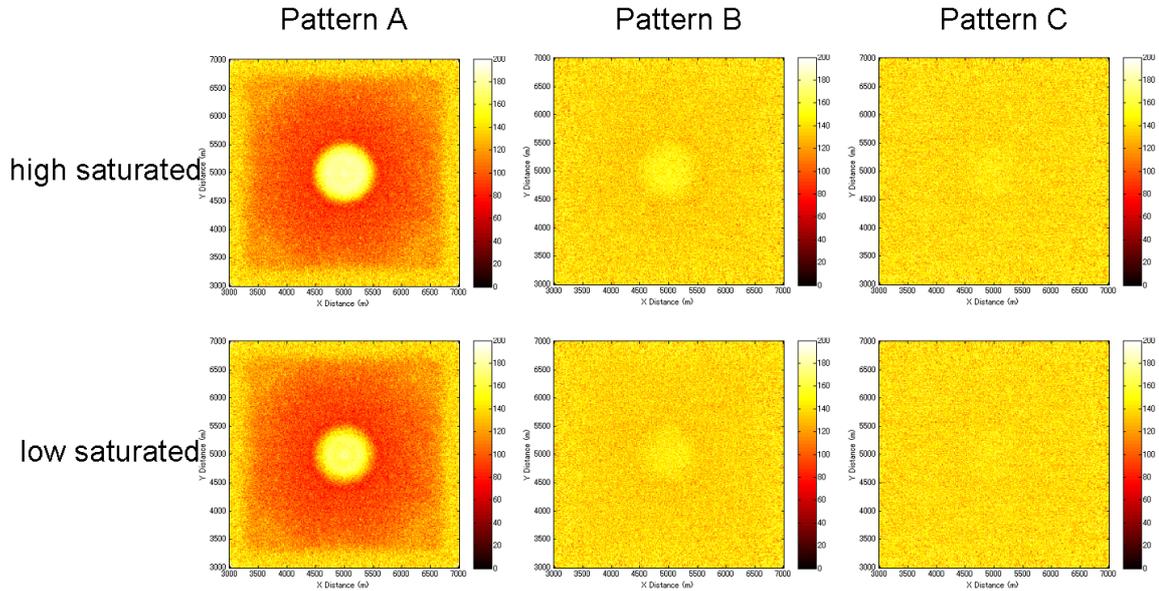
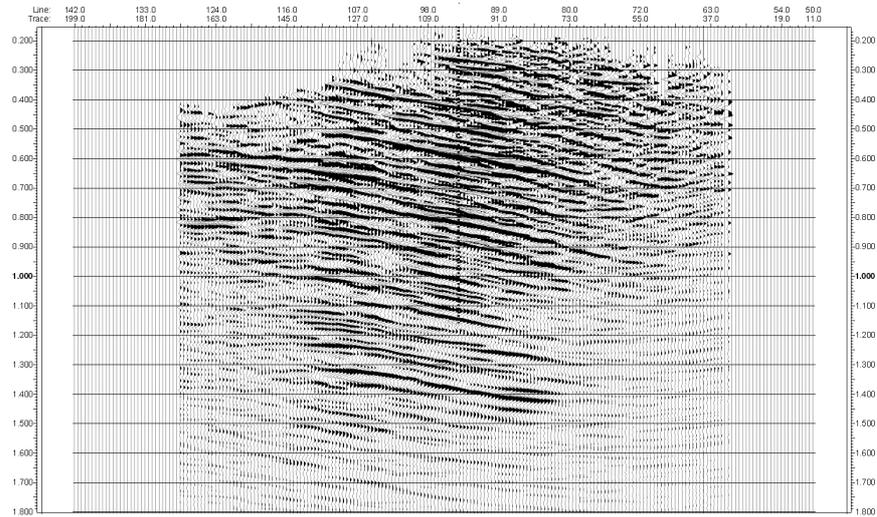


Figure II.2.11: NRMS of data that are added scattering and white noise. The arrangement of figure is same as II.2.10.

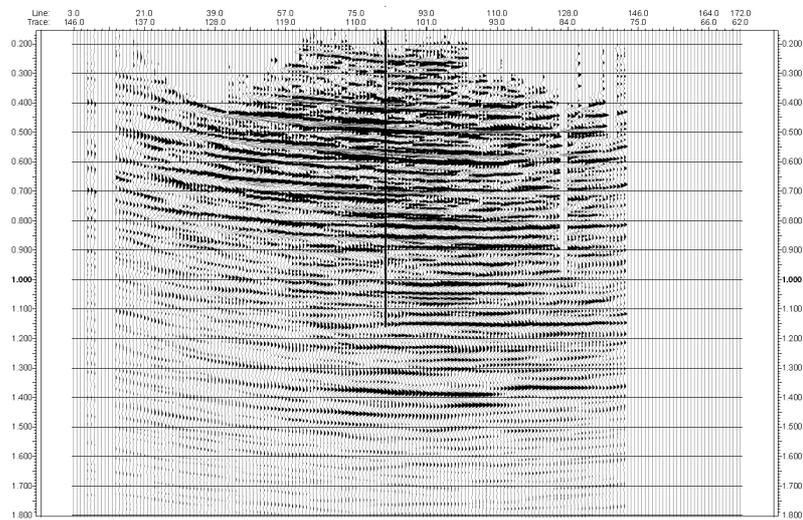
## II.2.2 Ray tracing simulation using realistic model

We make realistic model from the real 3D seismic volume where has logging data, too. After making model, I take the same way with one of synthetic model. The seismic volume is about  $2 \times 2 \text{ km}$  and I want to make  $5 \times 5 \text{ km}$  model. We show 2D seismic slice section that are East to West line (Figure II.2.12(a)) and North to South line (Figure II.2.12(b)) that are post-stack time migration section data. This region has single inclination from East to West and North to South is approximately horizontal. We also make synthetic seismogram from sonic and density data of well logging (Figure II.2.13). From synthetic seismogram, I can understand the position of target horizon and the basic wavelet extracted from seismic volume is similar to 35 or higher  $H_z$  ricker wavelet. We can pick some horizons that lead to strong reflection wave and have strong reflection coefficient in synthetic seismogram from 3D seismic volume. From picked horizons, I make grid data whose interval is  $100 \text{ m}$  and size is  $4.7 \times 4.5 \text{ km}$  (Figure II.2.14). We take blocking the logging data for setting the constant P and S wave velocities and density in each layer from well logging data and make 11 layers model (Table II.2.3).

We set the three  $\text{CO}_2$  injection zones that are  $100 \text{ m}$  radius between layer 7 and 8 ( $5 \text{ m}$ ) (injection



(a) East - West



(b) North - South

Figure II.2.12: Migration sections of real 3D seismic data. (a)East-West line. (b)North-South line. The broken line (a) and solid line (b) show the well that observe some properties.

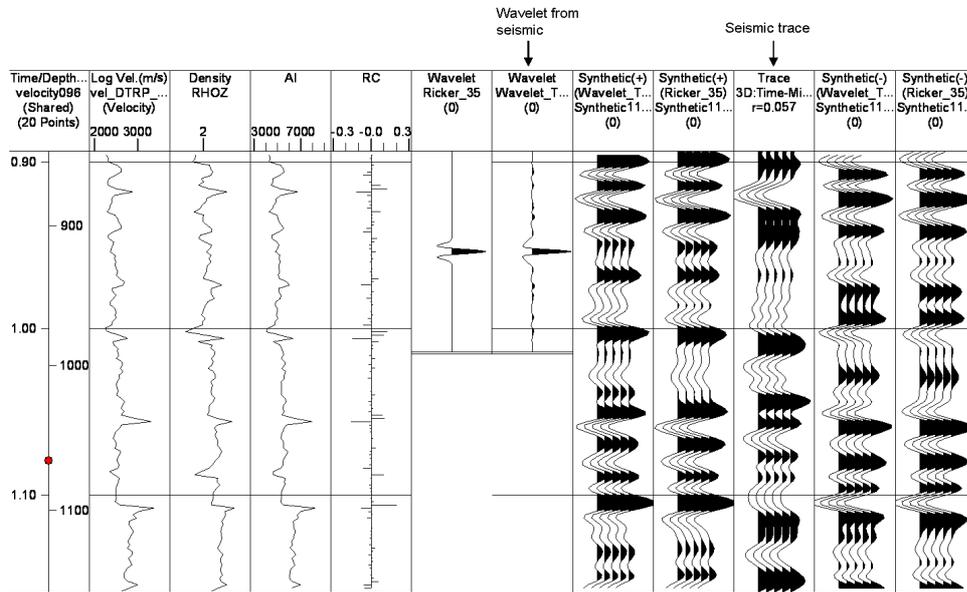


Figure II.2.13: Synthetic seismogram of logging data. The type of data from left are time and depth index, P wave velocity, density, acoustic impedance, reflection coefficient,  $35\text{Hz}$  ricker wavelet, wavelet that are extracted from seismic data around well, synthetic data that are convoluted extracted wavelet, synthetic data of ricker wavelet, seismic trace around well, opposite phase of synthetic data of extracted wavelet, and opposite phase of synthetic data of ricker wavelet.

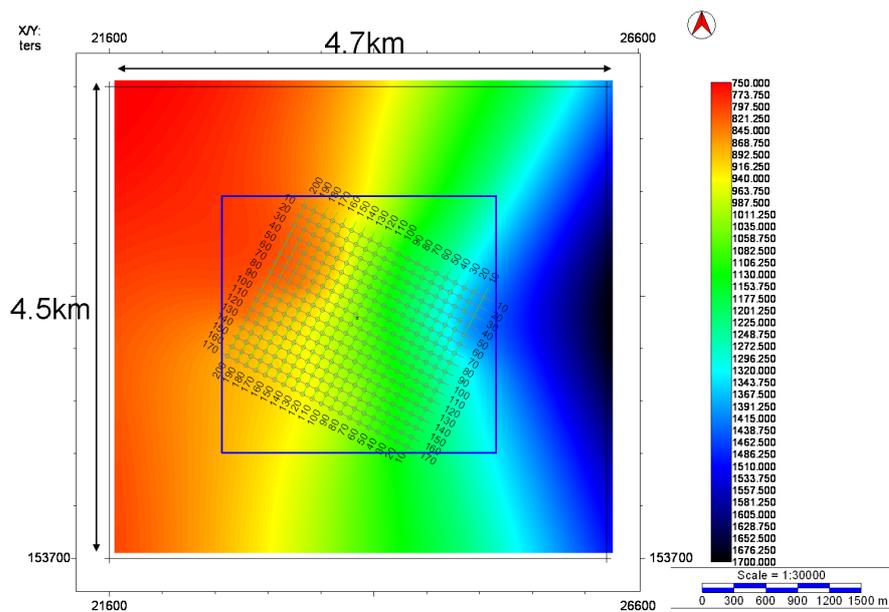


Figure II.2.14: Depth of grid of one horizon. After picking horizon, I take grids whole range of model. The unit of right colorbar is *m*. The grid in the center of figure is real 3D seismic survey range.

Table II.2.3: The properties of realistic model

Layer ID	depth (m)	velocity (m/s)		Density	note
	col	P wave	S wave	(kg/m <sup>3</sup> )	
1	50	1500	0	1000	the bottom of the sea
2	295	2000	800	2030	
3	320	2500	1220	2150	
4	900	2100	875	2080	
5	1140	2500	1220	2150	the top of the aquifer
6	1145	3500	1890	2500	
7	1150	2800	1455	2240	
8	1155	2800	1455	2240	
9	1175	2800	1455	2240	
10	1200	2800	1455	2240	the bottom of the aquifer
11		3000	1580	2300	
		2200	1480	2180	CO <sub>2</sub>

1), 500m radius between layer 7 and 8 (injection 2), and 225m radius between layer 7 and 9 (injection 3). We confirm the righteousness of blocking the logging data using FDTD simulation and making shotgathers. We assume the horizontally multi-layered structure and the properties are using 11 layers blocked model or using observed logging data directly (Figure II.2.15). Although I can image easily that the shotgather of observed logging data model has much more number of reflection waves, it is important that I can make the reflected waves that are come from target horizon or are observed close time from reflection wave from target horizon. The wave that exist about 1.1sec at 1,000m in Figure II.2.15(a) is reflected wave from target horizon. In comparison between Figure II.2.15(a) and II.2.15(b), it is very closely, especially near the target wave and I confirm that this blocking is right for using the ray tracing model. Because I also understand which wave overlap with target wave, I can calculate not only P primary wave but multiples or P-S converted wave that are distinguished imaging. We identify some waves came from which horizon in Figure II.2.15(b) (Figure II.2.16).

We simulate 3D seismic wave by ray tracing using this model (Figure II.2.17). We set the CO<sub>2</sub> injection portion in the center of model. This model is asymmetry and the velocity slice of X and Y is quite difference (Figure II.2.18). In Figure II.2.19, I show the CO<sub>2</sub> injection portion that is injection 1. This portion is between two horizons and horizontal distance is 200m in injection 1. We set sources

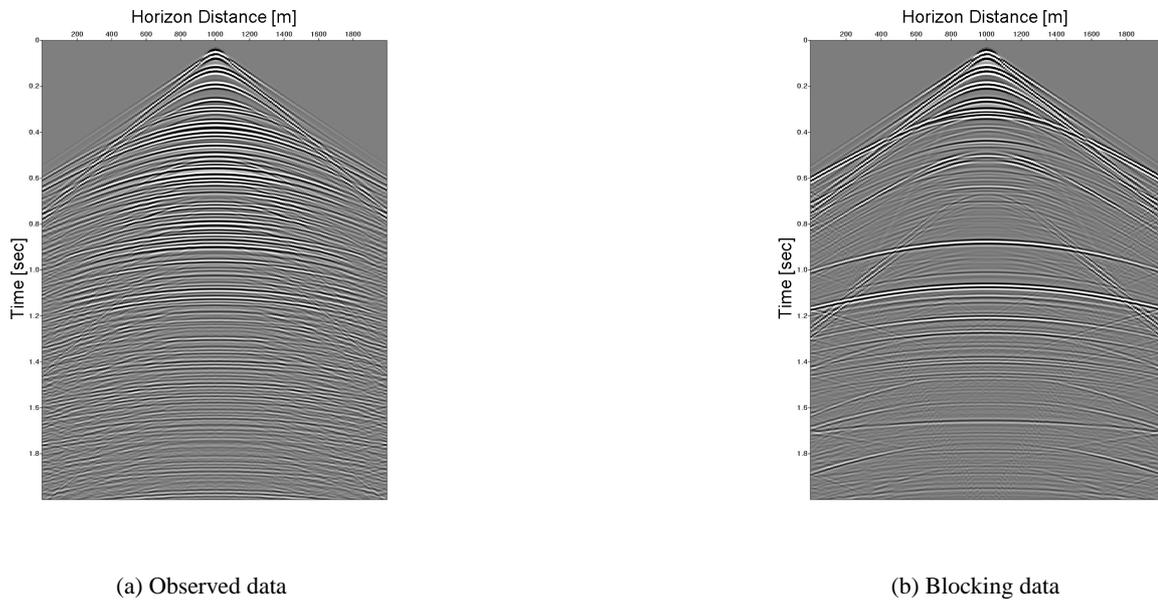


Figure II.2.15: Simulated sections of 2D FDTD method. The models are (a)observed logging data and (b)blocking data from observed data (Table II.2.3), and this is assumed horizontally multi-layered structure. The sources are top of the sea and horizon distance is 1,000m, and receivers are bottom of the sea.

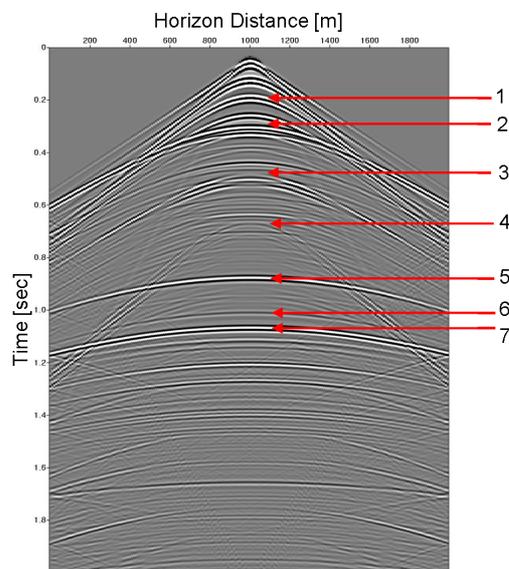


Figure II.2.16: Identification of wave that is existed in Figure II.2.15(b). No. 1 wave that are attached arrow is multiple between top and bottom of the sea. No. 2, 3 and 4 waves are P wave primary, P-S converted and S wave primary reflection wave from layer 2, reflectively. No. 5 and 6 waves are primary and multiple P reflected wave from layer 4 and No. 7 is the primary P reflected wave from layer 5 that are target horizon in this research.

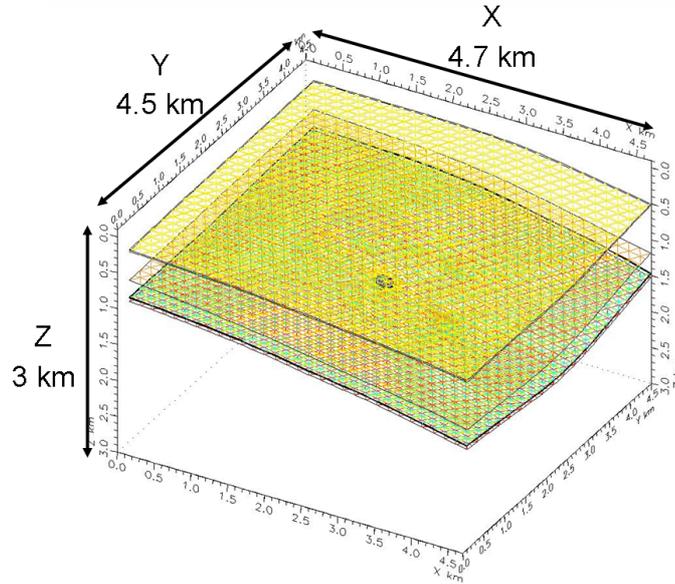


Figure II.2.17: Realistic model. The size of this model is  $4.7 \times 4.5 \times 3km$  and this has 11 layers.

at top of the sea and receivers at bottom of the sea because I want to simulate in realistic case that are air guns from a ship and OBC at the sea bed (Figure II.2.20). Because the source range is  $3 \times 3km$  and receiver range is  $2 \times 2km$ , the range of CMP is  $2.5 \times 2.5km$ . We also set 3 pattern receiver geometries but the source position is common by all patterns. In Pattern A, the interval of both source and receiver points is  $25m$ , and one of source and receiver lines are  $250m$  that are orthogonal geometry as same as synthetic model simulation. The interval of receiver points is  $100m$  and one of receiver lines are  $250m$  in Pattern B, and the interval of receiver points is also  $100m$  and one of receiver lines are  $500m$  in Pattern C. The number of sources and receivers shows in Table II.2.4. We show the number of folds

Table II.2.4: The number of sources and receivers in realistic model

	source	receiver
Pattern A	1573	729
Pattern B	1573	189
Pattern C	1573	105

(Figure II.2.21); it can fold under 30 in Pattern C.

We also simulate by Wavefront Reconstruction Method of ray tracing. The analyzing process is as same as one of synthetic model. In this simulation, I add coherent noise, scattering noise and

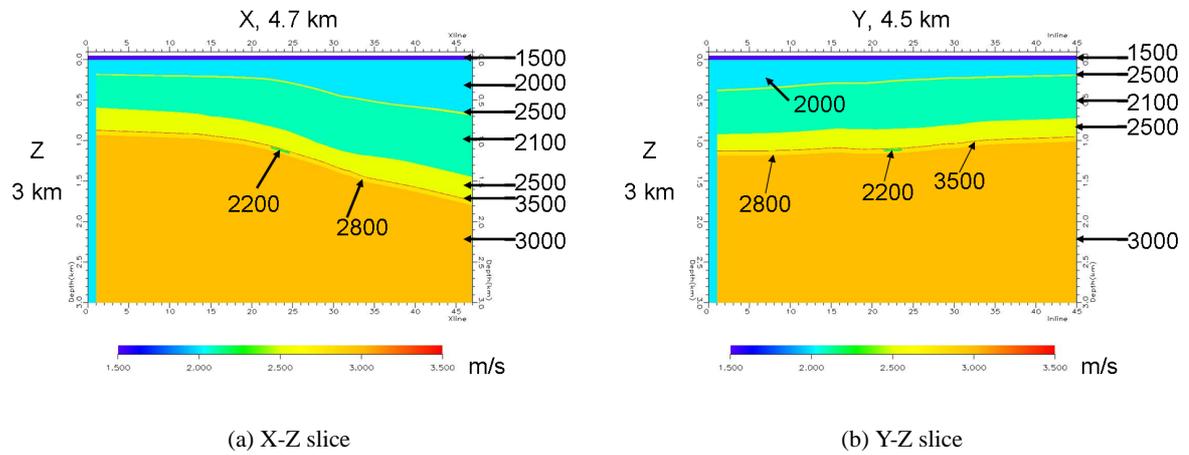


Figure II.2.18: P wave velocity slice. (a)X-Z slice at  $Y = 2.25\text{km}$  and (b)Y-Z slice at  $X = 2.35\text{km}$ .

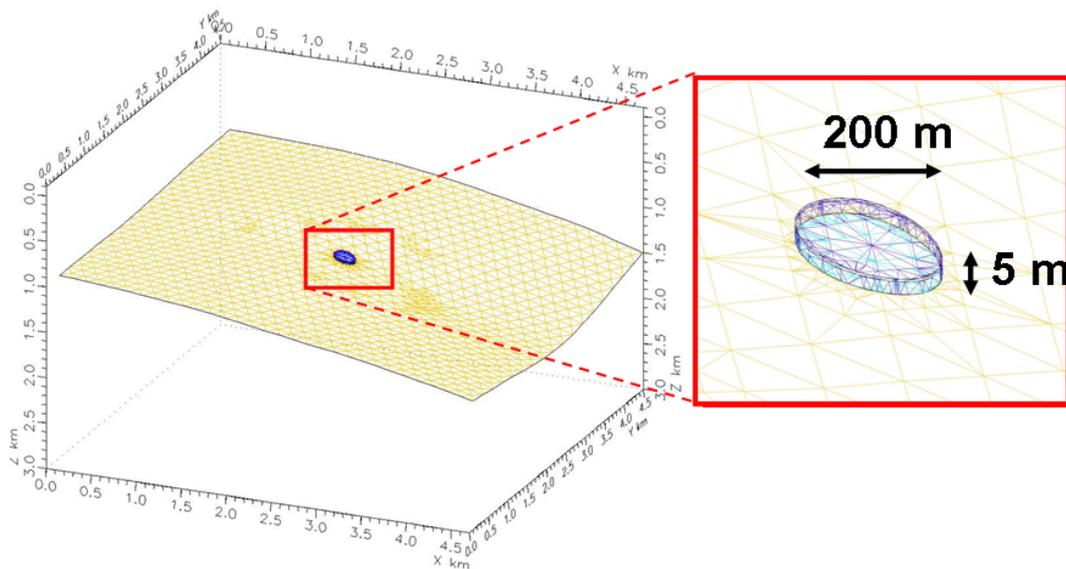


Figure II.2.19: The injection portion of injection 1 that size is  $200 \times 5\text{m}$ . This portion is center of  $X - Y$  plane and put between two layers.

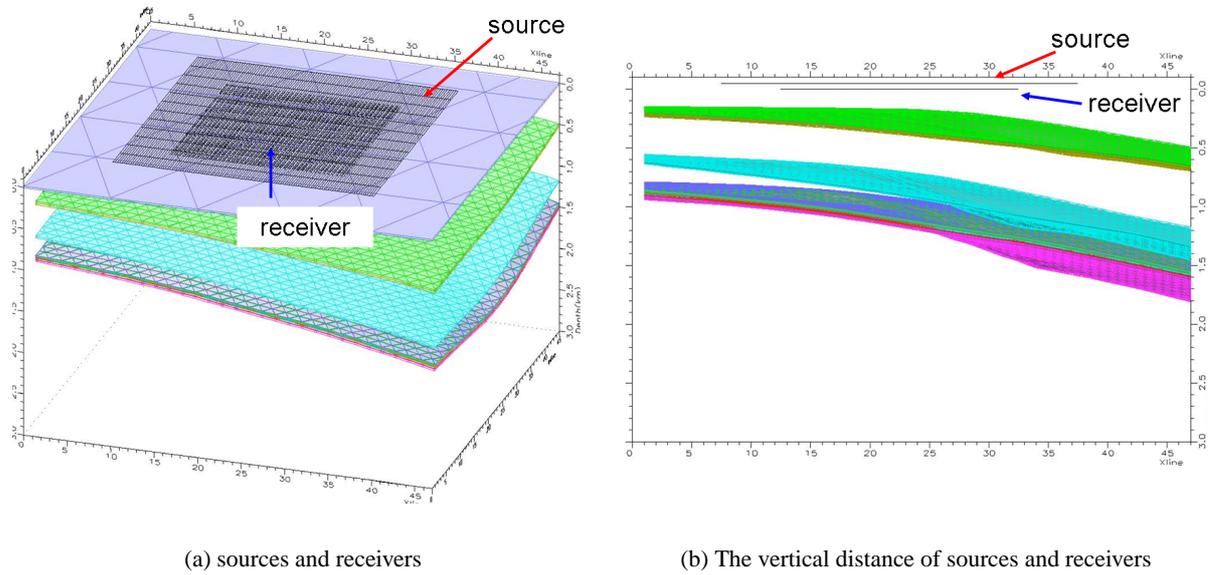


Figure II.2.20: The densest pattern of sources and receivers in realistic model. (a)survey with layers. The vertical distance between source and receivers is 50m(b).

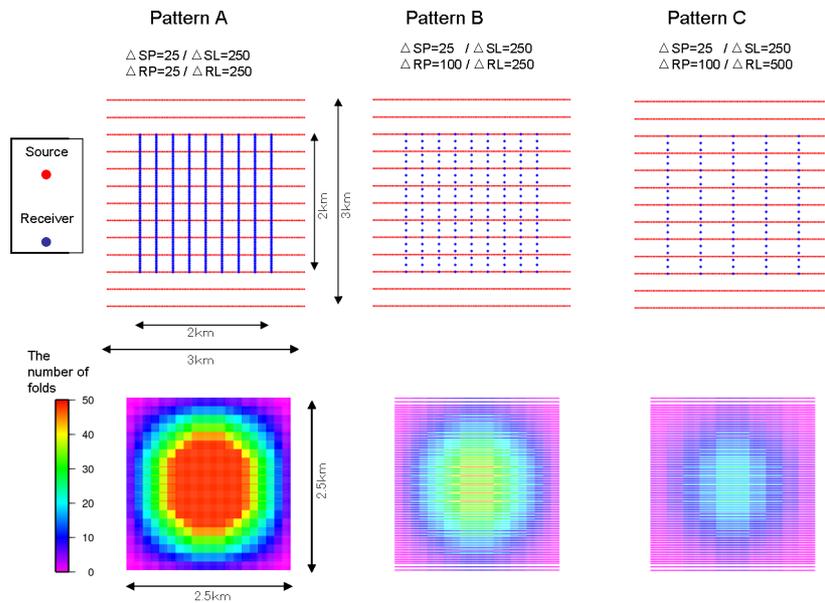


Figure II.2.21: The number of folds. Top figures are survey geometries. Red point is source and blue point is receiver. Bottom figures show the number of folds at all CMP. The size of CMP is  $2.5 \times 2.5km$ .

white noise that are 3 noise to signal ratios. Because I define some multiples and P-S or S-S waves for coherent noise, I received over 40 events that are the primary and coherent waves by one source and receiver. After ray tracing, I convolute 40Hz minimum phase ricker wavelet for constructing the seismic shot gather. We show the CMP gathers that are difference the variety of noises or the analyzing process (Figure II.2.22). Although this CMP gathers show the de-multi processing, this processing is not effectiveness and I do not apply de-multi in the processing flow. We change the injection zone that are injection 1, 2 and 3, and check the detectability. The wave length around target horizon is 70m and I can not recognize the reflection wave from target horizon separately when injection 1 and 2 (Figure II.2.23). Although it is hard, I can see the difference of amplitude that is depended on  $CO_2$  in noise free data. When I apply the migration, I can move the structure almost correct position that CNP number of the center of injection portion is 17,790. It is, however, more difficult to detect the anomaly since the  $CO_2$  injection because of smile noise. Thus, after now, I compare the before and after injection data at DMO stack and I adopt injection 3 because it is most realistic model when I inject 250,000 tons of  $CO_2$  in this reservoir. In Figure II.2.24, I add noise sequent. It is depended on the software limit that the multiple waves between the top and bottom of the sea is cut off suddenly at about 0.9sec. We simulate three noise patterns that are  $S/N = 4$  (Figure II.2.25),  $S/N = 2$  (Figure II.2.26) and  $S/N = 0.5$  (Figure II.2.27), respectively. The calculating method is same as one of synthetic model. We should choose the optimizing survey design that are possibly depended on noise level because I do not fix the injection region at now. When I analyze field data, it is impossible to know the correct velocity. Although I usually decide the velocity that constructs clearest section, this velocity should different from the correct velocity. Because in this simulation, however, I know the correct velocity, I should change the velocity explicitly. Concretely, I set the wrong velocity that is 5% slower than correct velocity (Figure II.2.25(d) - II.2.25(f), II.2.26(d) - II.2.26(f), II.2.27(d) - II.2.27(f)).

Although I know the effect of noise in 2D slice, the imaging point of anomaly move by incline in this model. Thus, I should calculate RMS amplitude on the target horizon that are picked from simulated 3D seismic volume because this data are not applied migration and this structure does not go

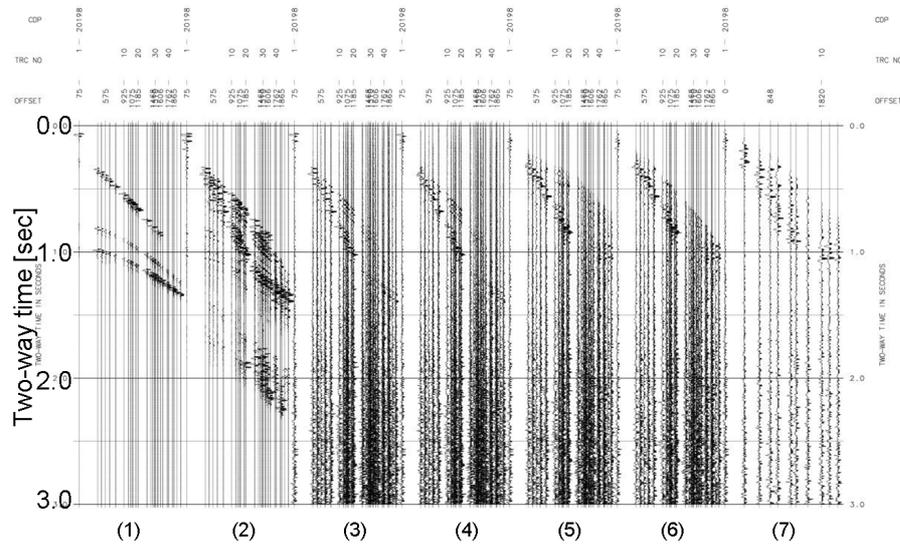


Figure II.2.22: CMP gathers. This gathers line keeping the distance of offset. (1)P wave primary only, (2)P wave primary and coherent noise that are multiple, P-S converted and S wave, (3)adding white noise from (2), (4)adding scattering noise from (3), (5)applying NMO correction to (4), (6)applying the handling that are remove the multiples to (5), (7)applying DMO correction to (6).

back to the initial model position. In this time, I use 50ms time range for calculating RMS because this model is little complex. We also calculate NRMS in same time range. At first, I show the difference of NRMS between injection 1 through 3 and primary only or adding coherent noise (Figure II.2.28). We can expect easily the difference between injection 1 to 3 that are all easy to detect because not injected place has no wave and injection 3 has highest value. Because NRMS value is not intuitive and the variation is not linear, the difference between injection 2 and 3 is important. It is wondering that the change when I add coherent noise because I simulate same coherent noise between before and after injection. Eventually, NRMS is depended on the absolute amplitude of each data that means when the both data have big amplitude certainly same phase and adding white noise, NRMS value is getting lower.

We show both NRMS and difference of RMS amplitude in each case (Figure II.2.29 - II.2.34). Because the NRMS and the margin of RMS amplitude have difference properties, I should use both methods for judgement. When I consider the velocity error, it is approximately impossible to detect the

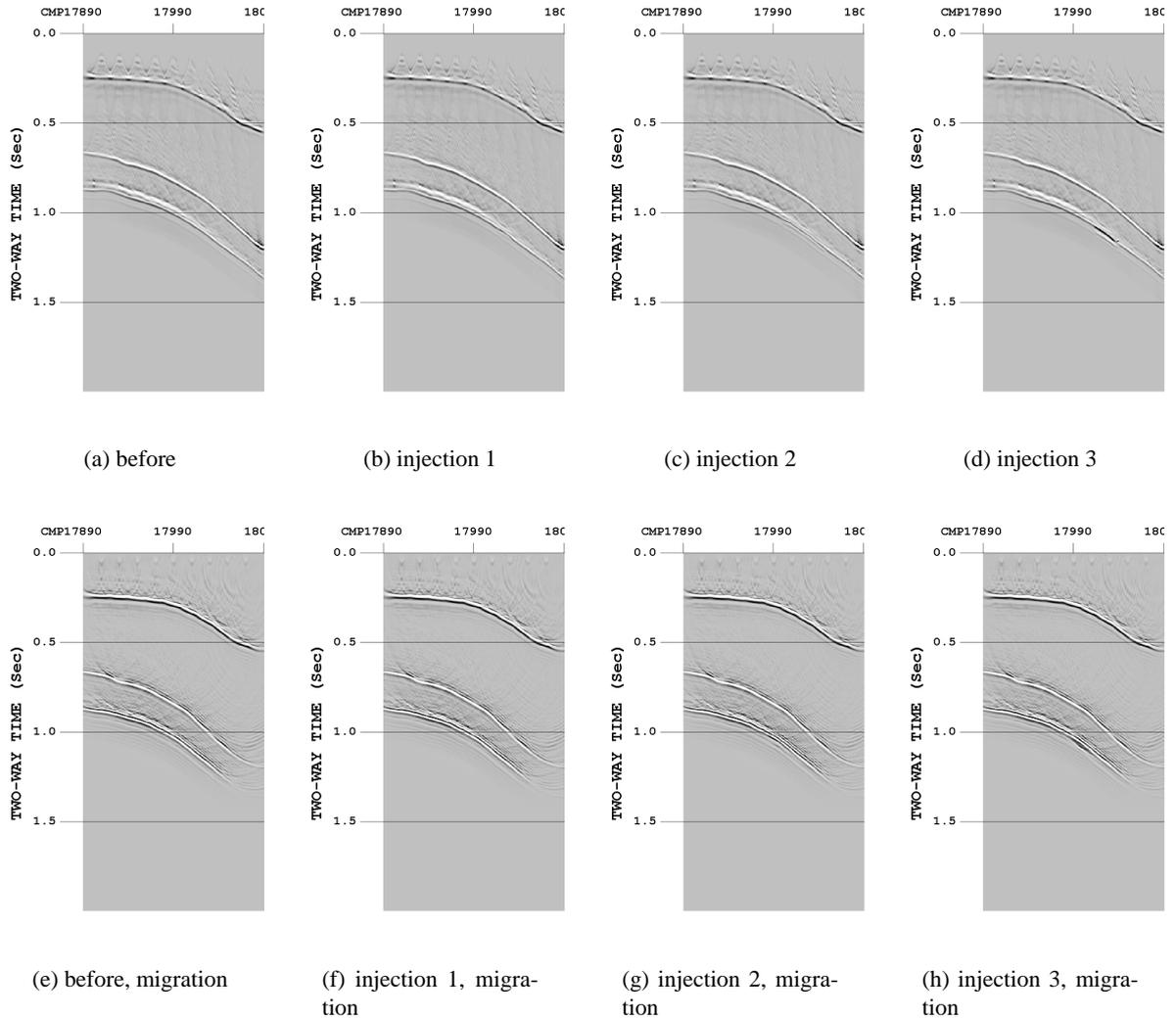


Figure II.2.23: DMO stack sections and Migration sections that are sliced at  $CMPY = 90$ . Top and bottom line figures are DMO stack and Migration section, respectively. (a) and (e) are before injection, (b) and (f) are smallest injection area, (c) and (g) are wide but thin injection area, and (d) and (h) are middle width and thickness.

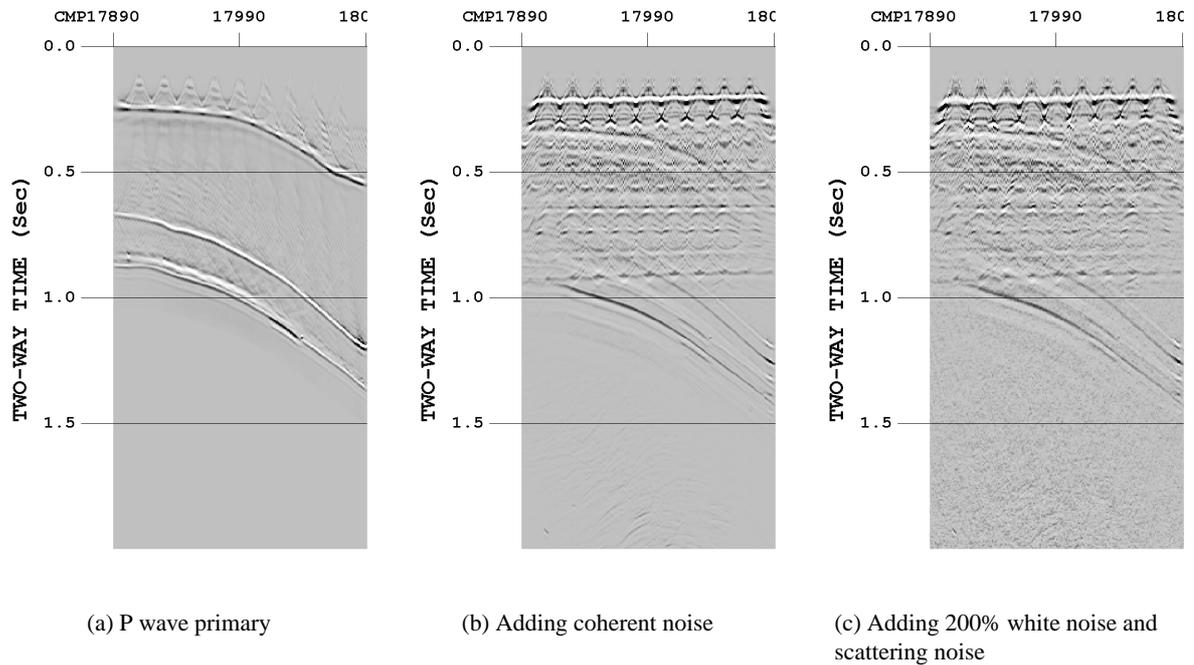


Figure II.2.24: DMO stack sections. (a)P wave primary only, (b)Primary wave and coherent noise, and (c)Adding white noise that are  $S/N = 0.5$  and scattering noise to (b). The slice position is same as Figure II.2.23.

anomaly in all Patterns of receivers of  $S/N = 0.5$ . Thus, I should set receivers denser in this noise level.

In  $S/N = 2$ , I may be able to detect when I use Pattern B or A. Because this discussion is completely just visibility, it is difference the result by analyzer. I discuss this subject in next chapter.

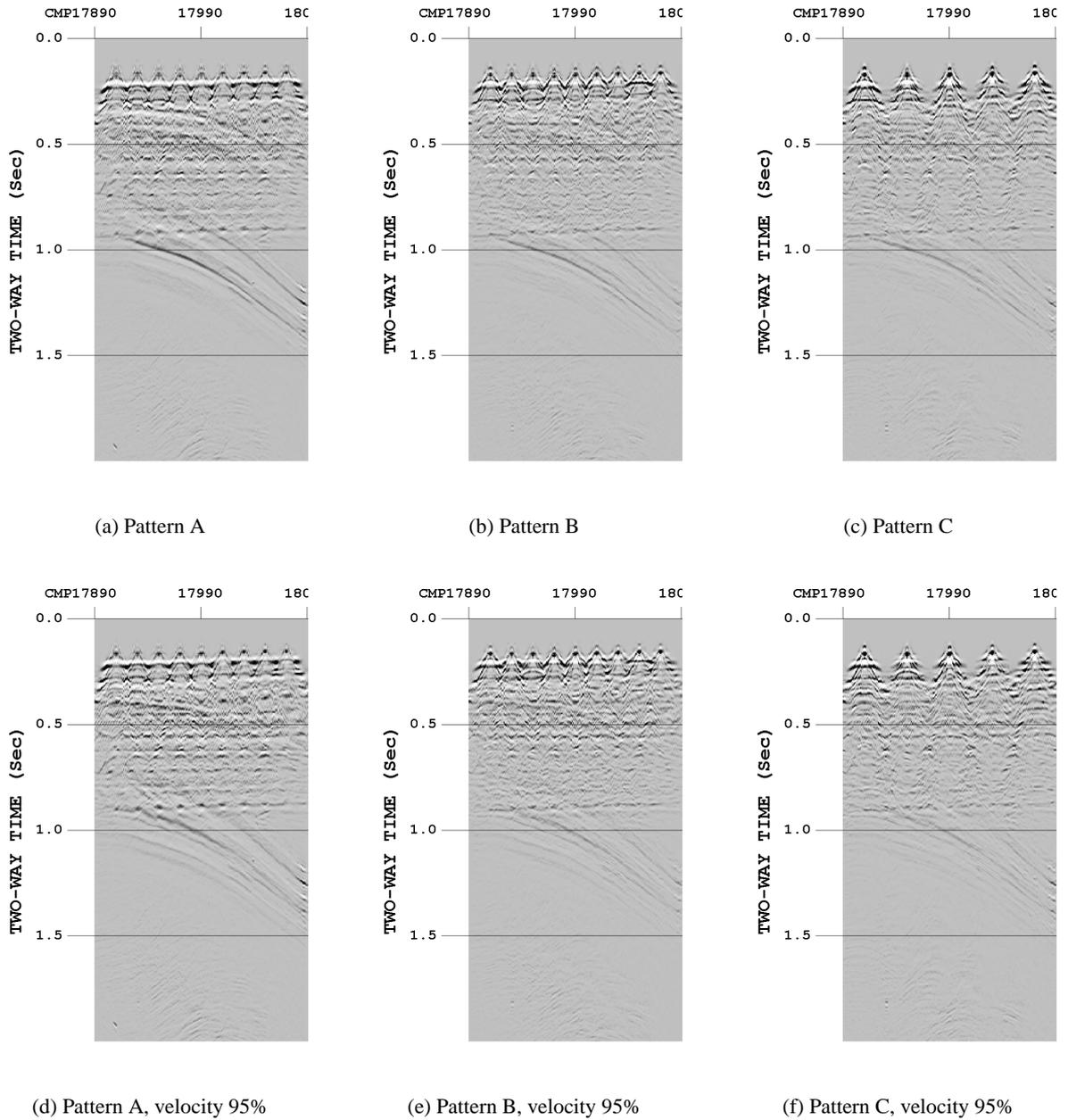
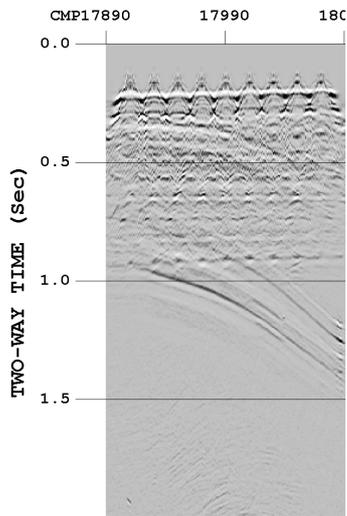
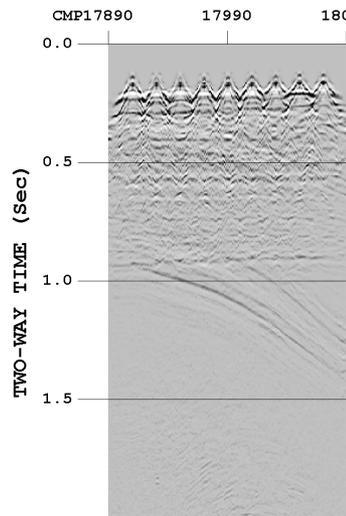


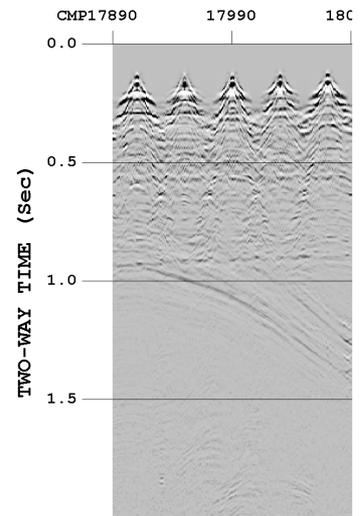
Figure II.2.25: DMO section of  $S/N = 4$ . Top and bottom line figures are using exact velocity and 95% velocity, respectively. The receiver geometry of (a) and (d) are Pattern A, (b) and (e) are Pattern B, and (c) and (f) are Pattern C. The slice position is same as Figure II.2.23.



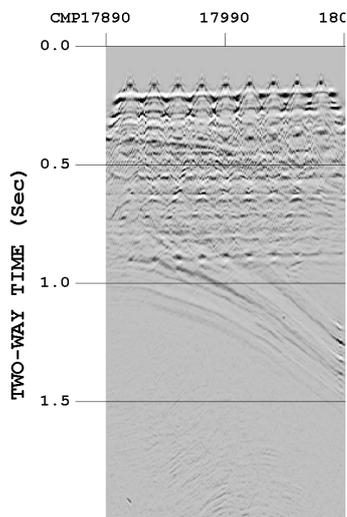
(a) Pattern A



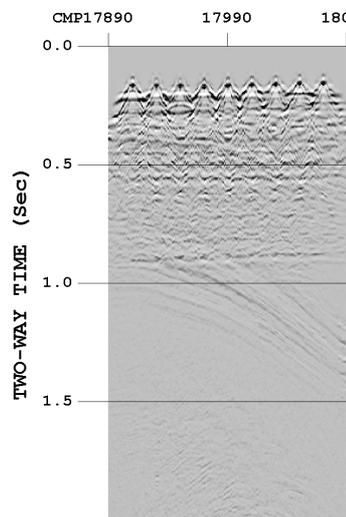
(b) Pattern B



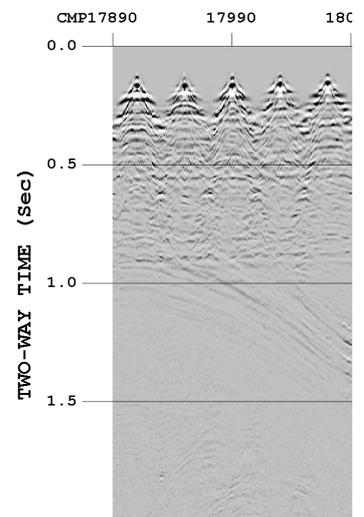
(c) Pattern C



(d) Pattern A, velocity 95%

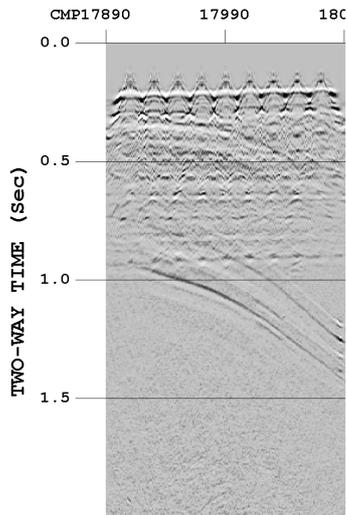


(e) Pattern B, velocity 95%

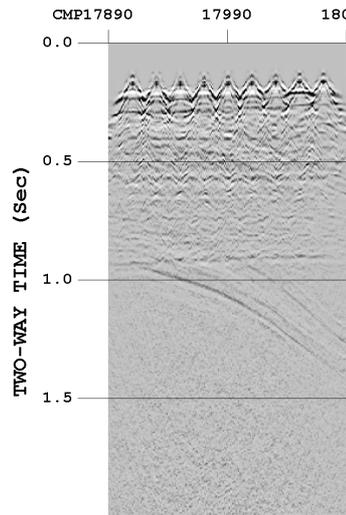


(f) Pattern C, velocity 95%

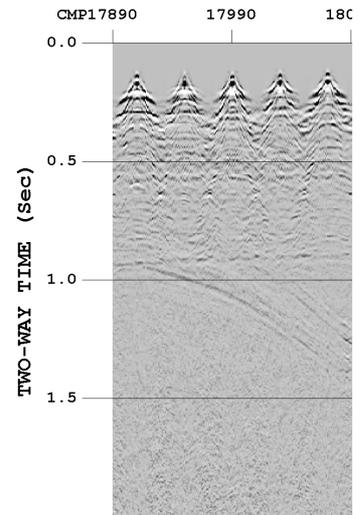
Figure II.2.26: DMO section of  $S/N = 2$ . The geometry and slice position of figures are same as Figure II.2.25.



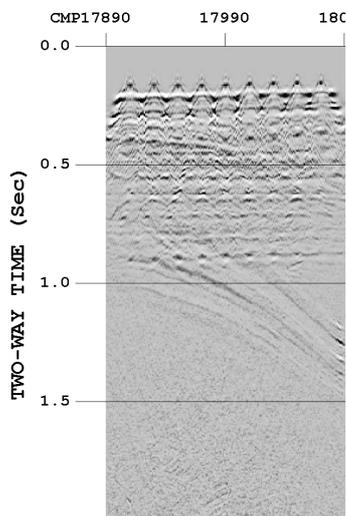
(a) Pattern A



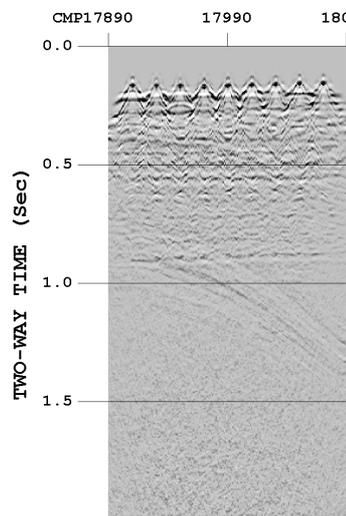
(b) Pattern B



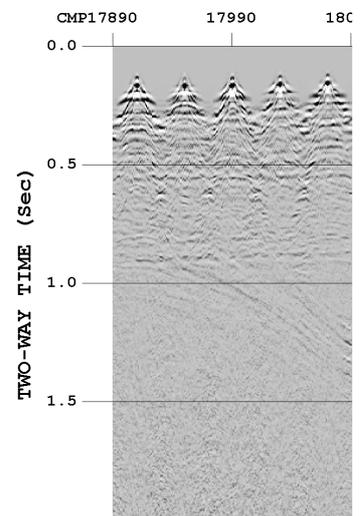
(c) Pattern C



(d) Pattern A, velocity 95%



(e) Pattern B, velocity 95%



(f) Pattern C, velocity 95%

Figure II.2.27: DMO section of  $S/N = 0.5$ . The geometry and slice position of figures are same as Figure II.2.25.

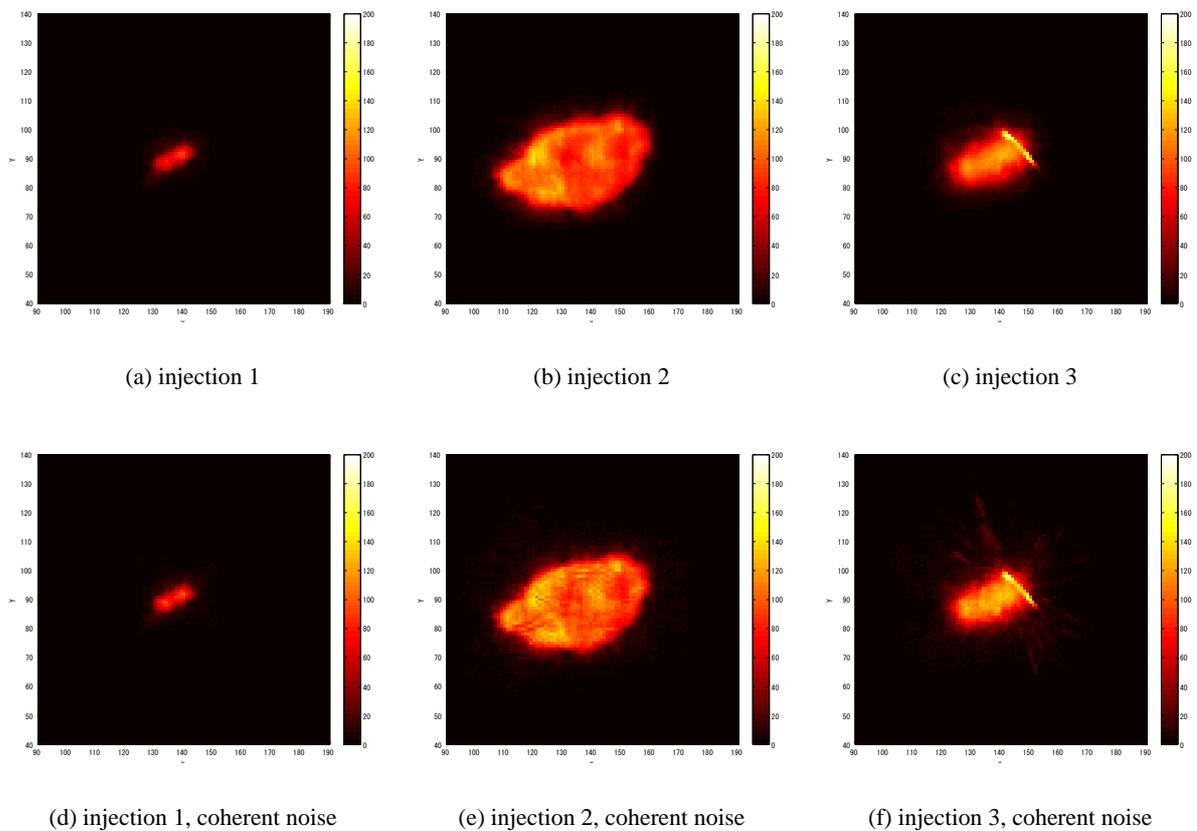


Figure II.2.28: NRMS at of DMO stack. The time gate of NRMS is  $50ms$  and the center of this gate is the top of aquifer. Top and bottom line figures are P wave primary only and adding coherent noise, respectively. The size of injection is (a) and (d) are smallest (injection 1), (b) and (e) are wide but thin (injection 2), and (c) and (f) are middle width and thickness (injection 3).

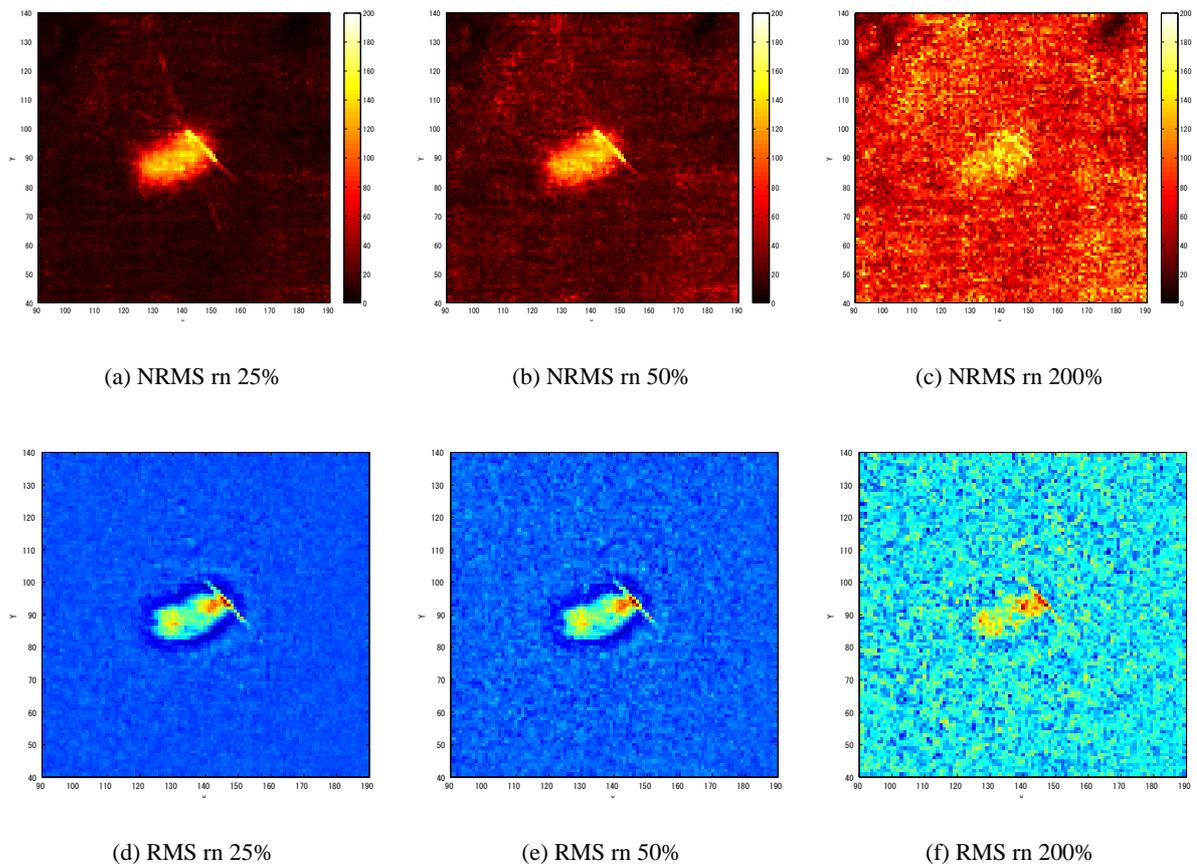


Figure II.2.29: NRMS and RMS of receiver geometry Pattern A. Top and bottom line figures are NRMS and RMS, respectively. The signal to noise ratio of (a) and (d) are 4, (b) and (e) are 2, and (c) and (f) are 0.5.

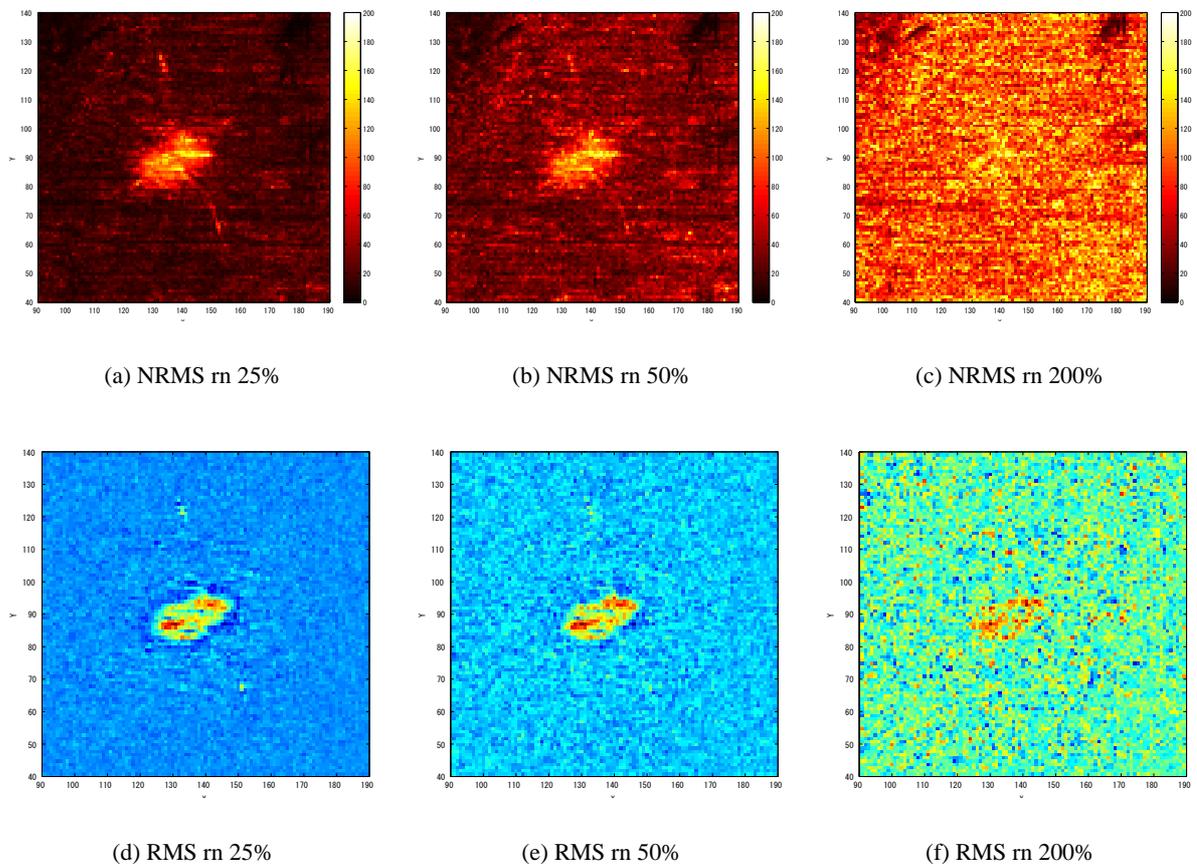


Figure II.2.30: NRMS and RMS of receiver geometry Pattern B. The figure geometries are same as Figure II.2.29.

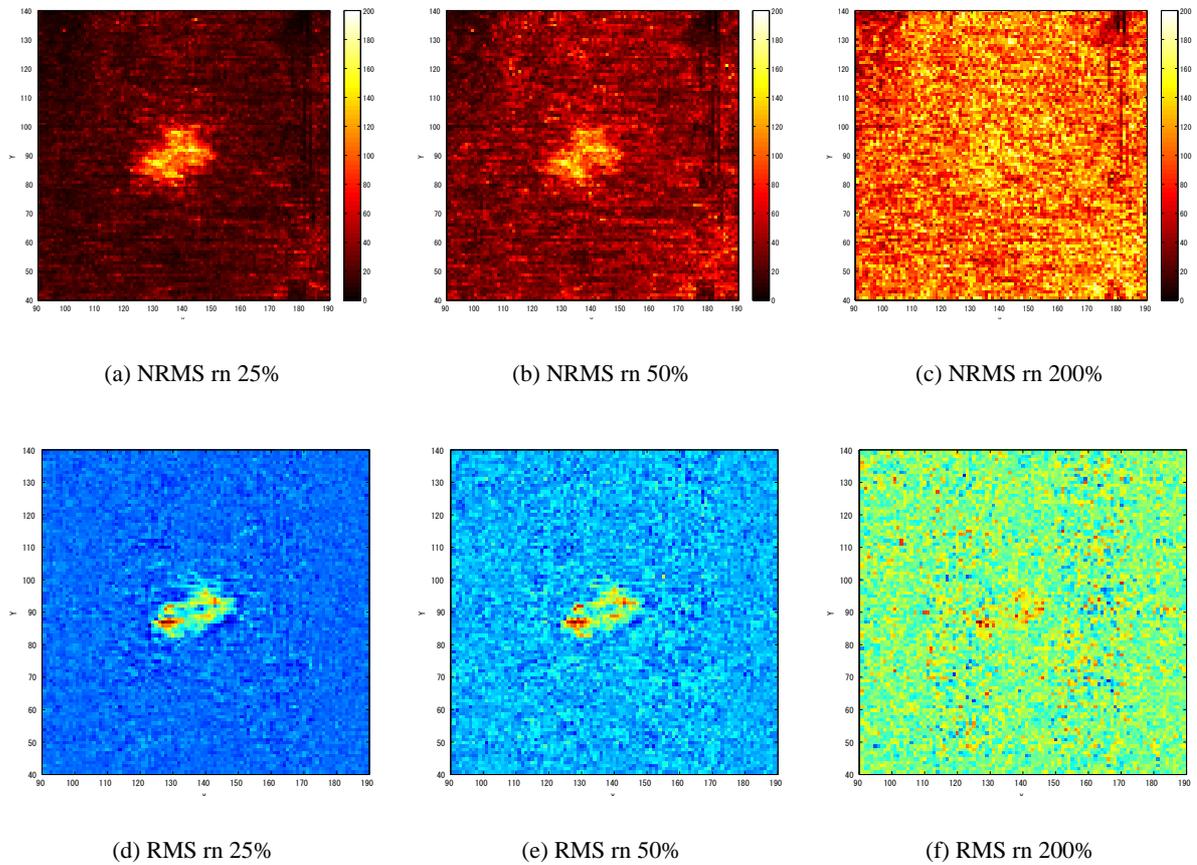


Figure II.2.31: NRMS and RMS of receiver geometry Pattern C. The figure geometries are same as Figure II.2.29.

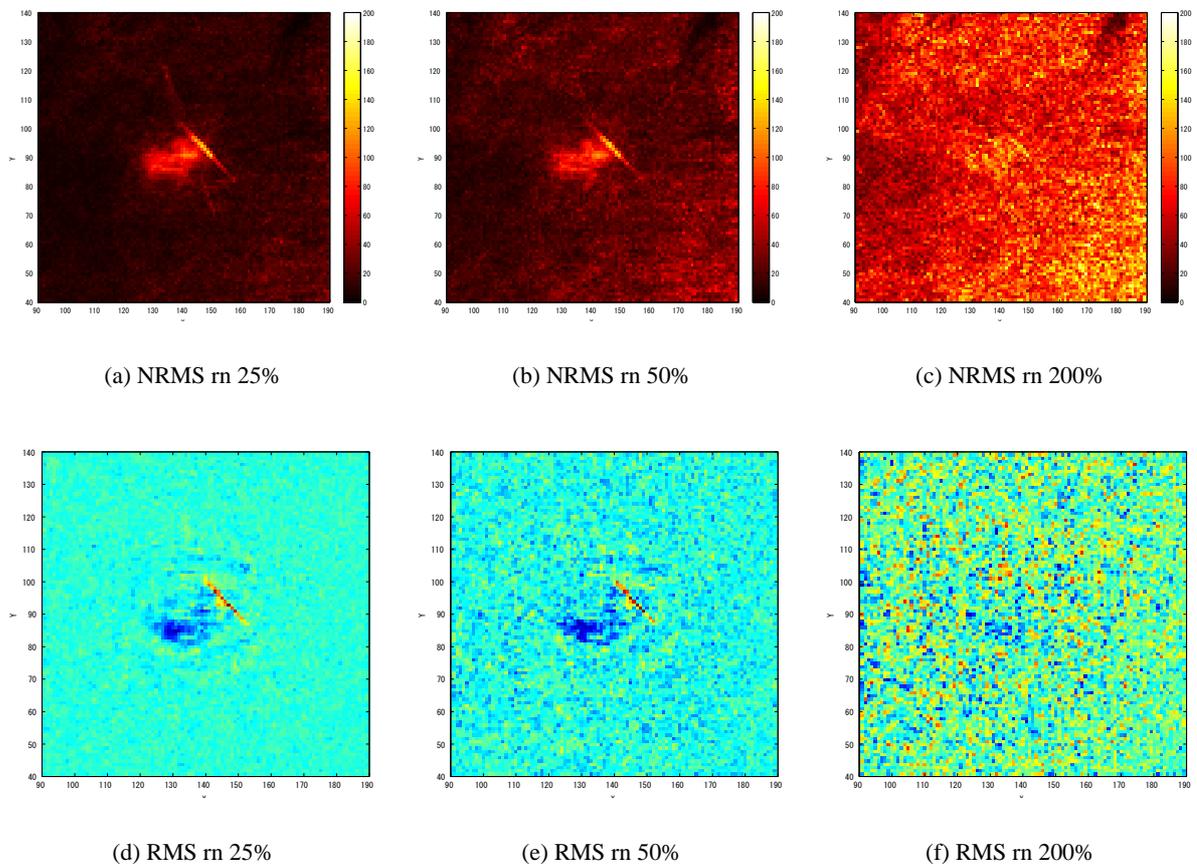


Figure II.2.32: NRMS and RMS of receiver geometry Pattern A and using 95% velocity. The figure geometries are same as Figure II.2.29.

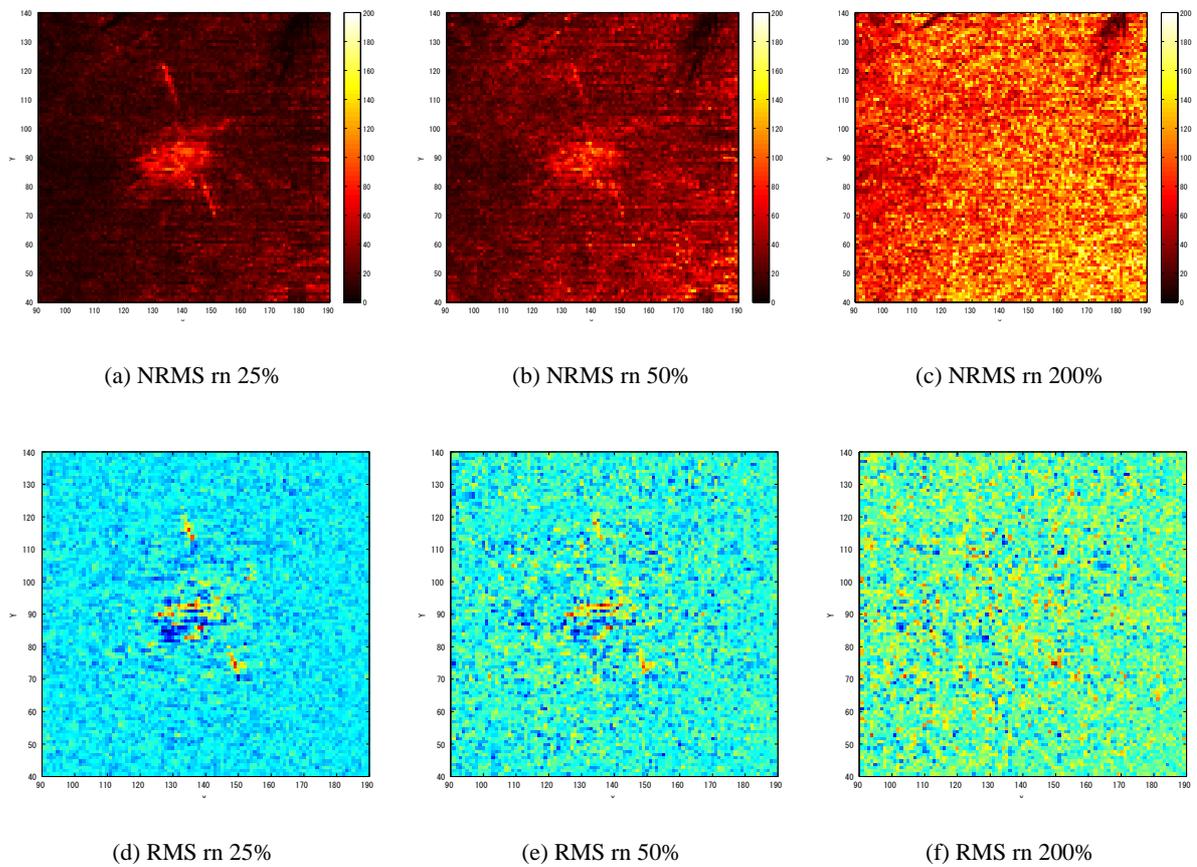


Figure II.2.33: NRMS and RMS of receiver geometry Pattern B and using 95% velocity. The figure geometries are same as Figure II.2.29.

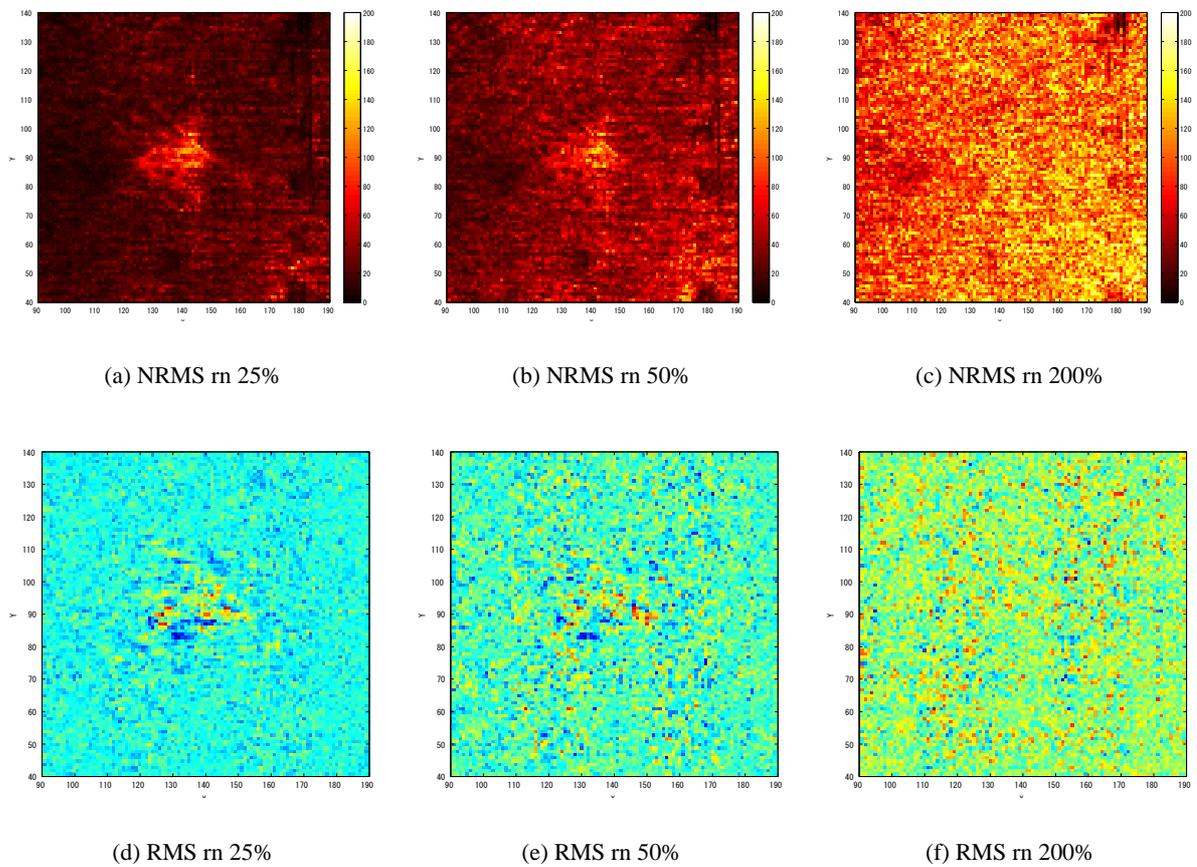


Figure II.2.34: NRMS and RMS of receiver geometry Pattern C and using 95% velocity. The figure geometries are same as Figure II.2.29.

## **Chapter II.3**

### **Discussion**

#### **II.3.1 Importance of migration in ray tracing data**

In this part, I estimate the difference at migration section in synthetic data processing and at DMO stack section in realistic data. Because ray tracing can not simulate diffraction wave, it is difficult to discuss the importance of migration. This problem is specifying the ray tracing and I do not have to care in field data processing or using full wave simulator like FDTD. In the usual wave that has full wave, when I apply migration, I can move the structure to correct position and the waves like diffraction waves are gathered. If I apply migration to the stack section made by ray tracing, although I can move the structure to correct position, I diffuse the wave since the smile noise by migration rather than focusing. When I only consider the focusing rate of anomaly, it is suitable to compare in DMO stack section whose focusing rate is possibly similar the migration stack of field data.

#### **II.3.2 Indicators of survey design using NRMS and RMS**

I use some values for recognizing the sensitivity of anomaly. Although visibility is one of important parameters, it is depended on the analyzer's skill. I want to propose the quantitative value. I have two ideas for estimation; one is averaging one dimension and other is averaging two dimensions and

compares the values that are injected and non-injected portion. The assumptions of these ideas are I know the injection point and injection point has maximum density of  $CO_2$ . I think this assumptions are reasonable.

In one-dimensional averaging method, I average the value in x or y direction (Figure II.3.1). The red point is an injection point, not the maximum value of red line. I calculate the maximum value that is mean of injection point and the values of 10 grids by back and forth from injection point, divided by other all values of red line. The number of 21 is usually I choose the estimating  $CO_2$  anomaly size. In this part, the size of  $CO_2$  is different because I use DMO stack data that are not applied migration. Although I get two rating values that are x direction and y direction, the difference of direction is not so meaningful when the media is isotropy and I take average of these two values. I call this value to one-dimensional averaging value. I set the ID number for tidiness of the difference of changing the kind of noises, receiver patters and velocity error (Table II.3.1). I calculate the one-dimensional averaging value (Figure II.3.2, Table II.3.2). Although the visibility in Table II.3.1 is just my subjectivity, I think it is important that the indicator like one-dimensional averaging value either follow or not my subjectivity. Criterion of visibility is I can or can not distinguish the injection position without foresight information. It is important to distinct the area and the edge of  $CO_2$ . When I take the index of visibility, I also compare the size of anomaly. It is difficult to check the size. I decide the size changing the color or maximum value of plot.

In visibility of Table II.3.1, NRMS and RMS is little different; NRMS is vulnerable to white noise, on the other hand, RMS is vulnerable to velocity error and difficult to measure the anomaly size. I compare the visibility and one-dimensional averaging value. I can view and distinguish the edge over 1.2 in NRMS and over 2.4 in RMS. I can not view under 1.08 in NRMS and 1.123 in RMS but 1.123 is also distributed to visible but indistinguishable the edge class. Although this value is not meaningful because this is depended on the size of analyzing range that is  $101 \times 101$  CMPs in this analysis, one-dimensional averaging value likely agree the visibility.

I propose other method that is two-dimensional averaging method. This method is not depended

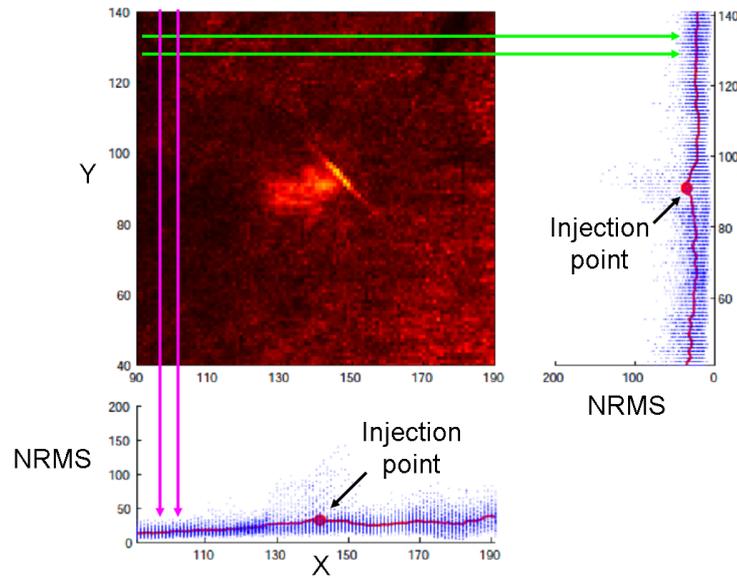


Figure II.3.1: The computation method of one-dimensional averaging. I take average by green arrow direction and pink arrow direction. The blue points of each graph are NRMS value of each point that are not averaging. The red line of each graph is average of blue points and red circles are injection point.

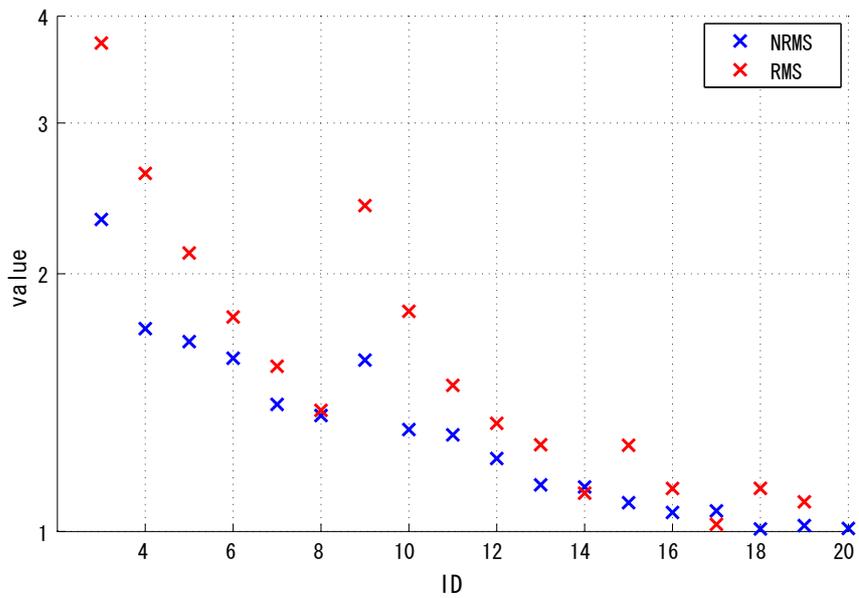


Figure II.3.2: One-dimensional averaging value. ID numbers correspond to Table II.3.1. Blue crosses are NRMS and Red crosses are RMS.

Table II.3.1: The relationship of ID number and the variety of data

ID	primary	coherent	noise scattering	white	receiver	velocity error	visibility NRMS	RMS
1	○				250 × 25m		○	○
2		○			250 × 25m		○	○
3		○	○	$S/N = 4$	250 × 25m		○	○
4		○	○	$S/N = 4$	250 × 100m		○	○
5		○	○	$S/N = 4$	500 × 100m		○	△
6		○	○	$S/N = 4$	250 × 25m	○	○	△
7		○	○	$S/N = 4$	250 × 100m	○	○	△
8		○	○	$S/N = 4$	500 × 100m	○	○	△
9		○	○	$S/N = 2$	250 × 25m		○	○
10		○	○	$S/N = 2$	250 × 100m		○	△
11		○	○	$S/N = 2$	500 × 100m		○	△
12		○	○	$S/N = 2$	250 × 25m	○	○	△
13		○	○	$S/N = 2$	250 × 100m	○	△	△
14		○	○	$S/N = 2$	500 × 100m	○	△	×
15		○	○	$S/N = 0.5$	250 × 25m		△	△
16		○	○	$S/N = 0.5$	250 × 100m		×	△
17		○	○	$S/N = 0.5$	500 × 100m		×	×
18		○	○	$S/N = 0.5$	250 × 25m	○	×	×
19		○	○	$S/N = 0.5$	250 × 100m	○	×	×
20		○	○	$S/N = 0.5$	500 × 100m	○	×	×

○, △ and × in visibility are visible and distinguishable the edge, visible but indistinguishable the edge, and invisible, respectively.

on the size of analyzing range (Figure II.3.3). In this method, at first, I do gridding by some intervals of CMP that are possibly depended on the complexity of structure or volume of  $CO_2$  and I use 10 CMPs intervals in this part. Then, I take average by each grid and compare the values that are injection point (blue circle in Figure II.3.3) and the others. I call this value to two-dimensional averaging value. I calculate the two-dimensional averaging value as same ID as one-dimensional averaging value (Figure II.3.4, Table II.3.2). I also compare the visibility and two-dimensional averaging value. I can view and distinguish the edge over 2.0 in NRMS and over 7.0 in RMS. I can not view under about 1.4 in NRMS and about 1.5 in RMS. Because I do not define the injection area in detail, invisible indicator like 1.4 and 1.5 is not clearly, especially RMS. I think, however, it is enough that I can distinguish the edge of injection area by 125m. Some points are improved by two-dimensional averaging method. In NRMS, the relative relationship between "visible and distinct the edge point" and "visible but indistinct the

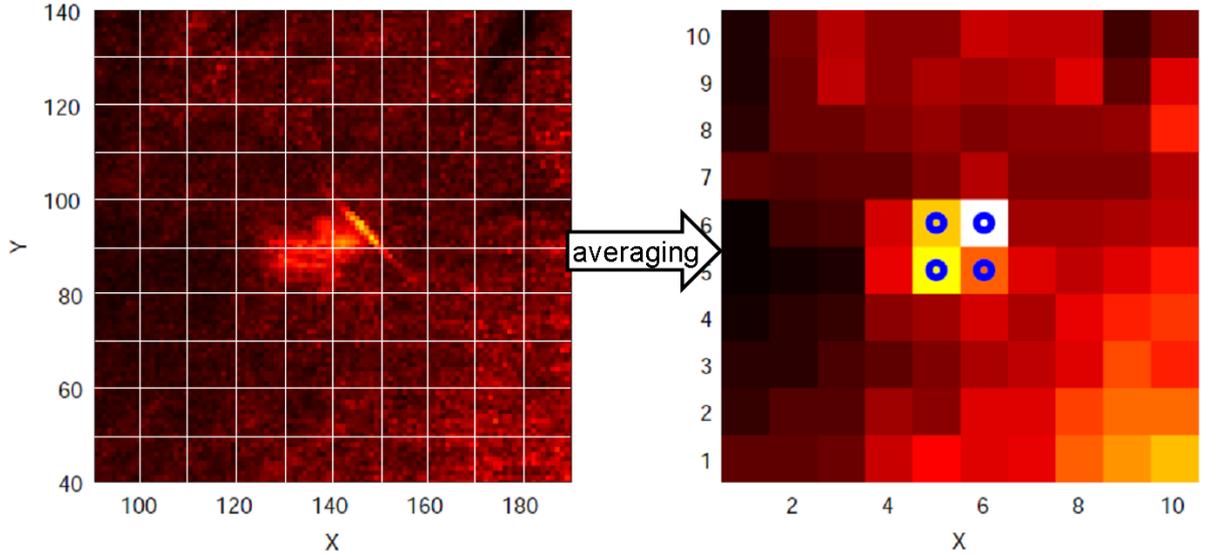


Figure II.3.3: The computation method of two-dimensional averaging. I take grids on the NRMS or RMS horizontal section and take average in each grid. Blue circles on the averaging figure show injection portion. The color between before and after averaging does not indicate same value.

edge” is extend like between ID 12 and 13. I also confirm the size of injection portion by changing the number and position of blue circle in Figure II.3.4. The relative relationship between ID 15 and 16 is also extending. In RMS, relative relationship of ID 18 and 19 is low. I think two-dimensional averaging method is easier to distinguish.

Next, I improve the calculation of NRMS. I indicate at previous chapter that NRMS is depended on the absolute amplitude of each data. For repairing this thing, I would calculate NRMS divided by regulated RMS amplitude of before injection (DNRMS). DNRMS is getting from

$$DNRMS = \frac{NRMS}{\frac{RMS(b_i)}{average(RMS(b_i))}} = \frac{200 \times RMS(a_i - b_i) \times average(RMS(b_i))}{\frac{RMS(a_i)}{RMS(b_i)} + 1}, \quad (II.3.1)$$

$b_i$  is before injection. Using DNRMS, I can remove the effect of amplitude that is same value between before and after (Figure II.3.5). In Figure II.3.5(a), upper-left and lower-right are different colors from around. This anomaly is just bigger or smaller amplitude relatively and not important different for time-lapse monitoring. I apply DNRMS for this data and I get Figure II.3.5(b). I can almost remove

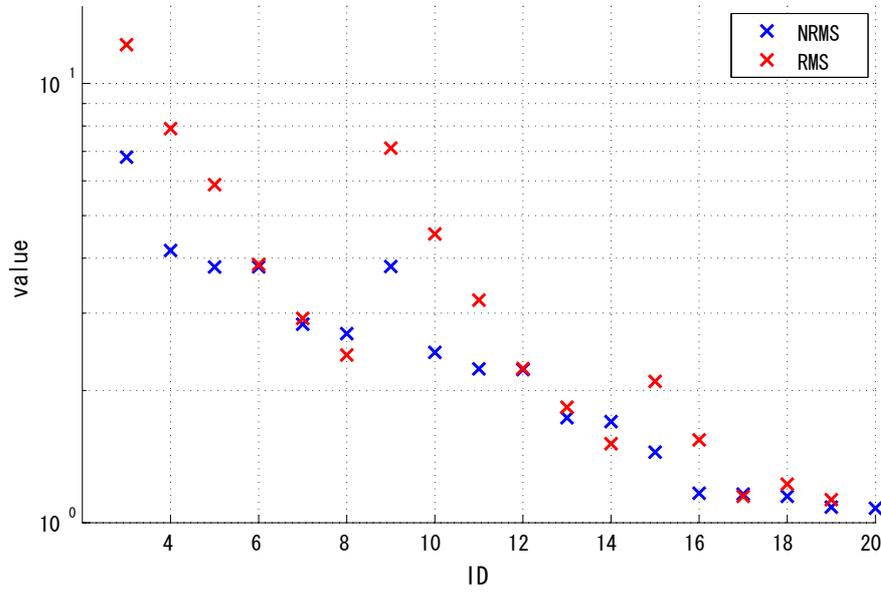


Figure II.3.4: The rating value of two-dimensional averaging. Blue crosses are NRMS and Red crosses are RMS.

the color difference and get clear horizontal section. I apply DNRMS to all IDs and calculate one or two-dimensional averaging value (Figure II.3.6, II.3.7, Table II.3.2). Visibility and averaging values in receiver pattern A and B are improved by DNRMS. The reason of averaging values change to the worse in pattern C is that this pattern is too sparse receivers to imaging by  $12.5 \times 12.5m$  CMP interval and the number of fold is too small. The suitable interval is  $50 \times 50m$ . Thus, I think it will be also improve when I adjust the interval of CMP in pattern C.

In the fact, the monitoring may be disturbed by the depth of sea or the difference of instruments. Thus, I have time shift or phase shift between two time-lapse data. Because NRMS is extremely sensitive to the smallest of changes in the data, it is possibly difficult to get the change from injection. Predictability is another indicator of time-lapse monitoring and it is possibly useful. Because predictability use crosscorrelation, this is not sensitive to overall static, phase, or amplitude differences. It is sensitive to noise and to changes in the earth reflectivity. Predictability is calculated

$$PRED = \frac{\sum \Phi_{ab}(\tau) \times \Phi_{ab}(\tau)}{\sum \Phi_{aa}(\tau) \times \Phi_{bb}(\tau)}, \quad (\text{II.3.2})$$

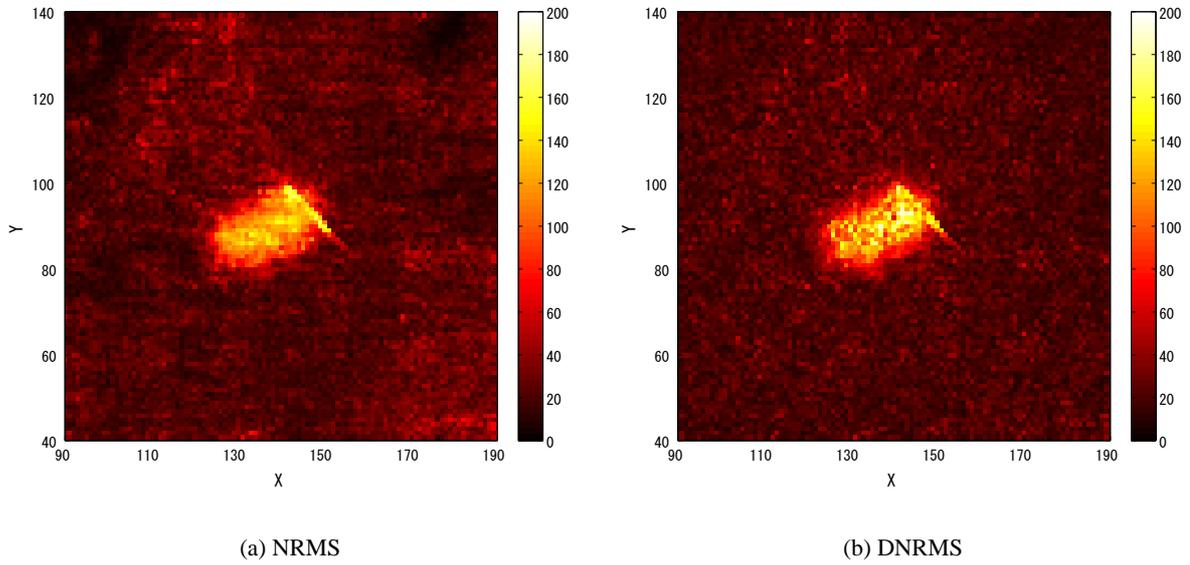


Figure II.3.5: (a)NRMS and (b)DNRMS.

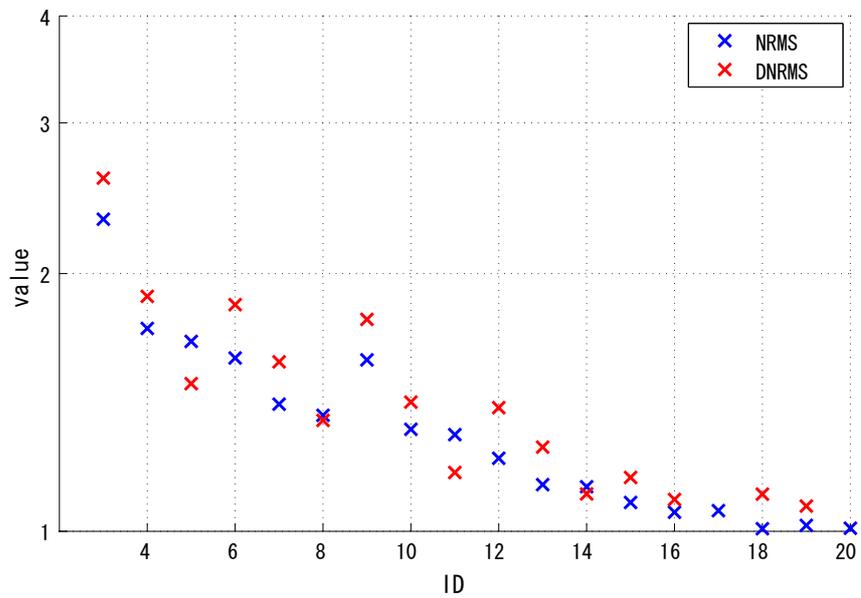


Figure II.3.6: The rating value of one-dimensional averaging of NRMS (blue crosses) and DNRMS (red crosses).

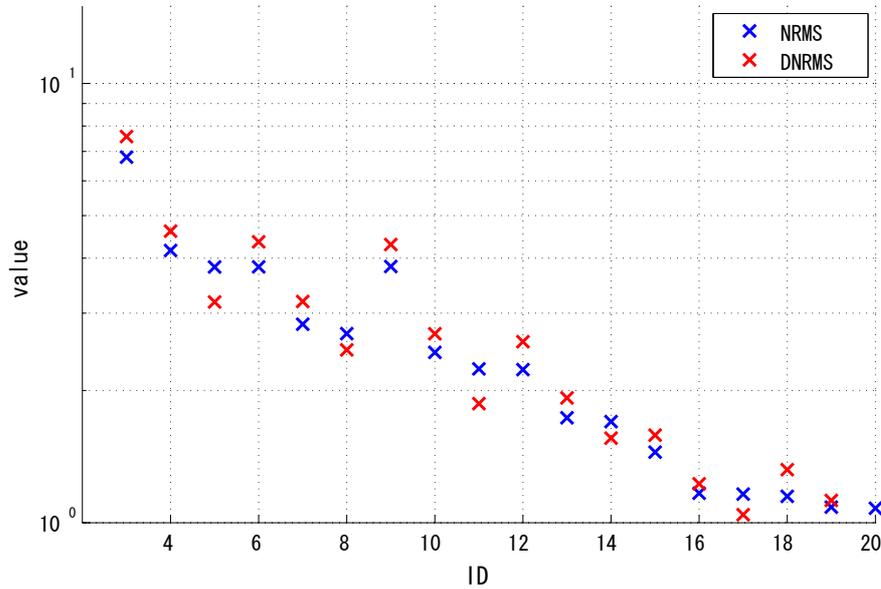


Figure II.3.7: The rating value of two-dimensional averaging of NRMS (blue crosses) and DNRMS (red crosses).

where  $\Phi_{ab}$  denotes the crosscorrelation between traces  $a_t$  and  $b_t$  computed within the time window  $t_1 - t_2$ . Because predictability is sensitive to the length of the correlation time window, absolute numbers are not meaningful. In this simulation, because I do not simulate time shift or phase shift, the difference of NRMS or predictability is not critical.

In fact, I must estimate the amount of noises from before placing semi-permanently OBC for applying this indicator and using the construct sources and receivers. Thus, my idea is that I use ordinary OBC at first and decide the number of receivers for placed OBC.

### II.3.3 Application of seismic interferometry

Judging from the above, I would be able to detect the anomaly since  $CO_2$  injection in receiver pattern A when  $S/N > 0.5$ . Possibly, I can further reduce the number of receivers using seismic interferometry (Schuster, 2009). Although I have much multiples when I get the shotgather, I use only primary wave. By using multiples, I can construct pseudo shot gathers. I indicate the wavefield  $u(r, s)$

Table II.3.2: The comparison of value in each analyzing method

ID	one-dimensional averaging value			two-dimensional averaging value		
	NRMS	RMS	DNRMS	NRMS	RMS	DNRMS
1	22.71	34.47	24.33	43.25	57.14	46.76
2	10.58	16.40	11.67	28.20	42.63	31.29
3	2.315	3.720	2.585	6.795	12.25	7.567
4	1.725	2.620	1.881	4.165	7.890	4.611
5	1.666	2.115	1.487	3.821	5.881	3.180
6	1.594	1.780	1.839	3.824	3.871	4.359
7	1.407	1.560	1.577	2.830	2.919	3.189
8	1.366	1.385	1.347	2.695	2.409	2.474
9	1.586	2.403	1.768	3.831	7.124	4.299
10	1.315	1.808	1.416	2.442	4.544	2.693
11	1.297	1.482	1.171	2.240	3.211	1.866
12	1.217	1.338	1.395	2.231	2.244	2.582
13	1.133	1.263	1.254	1.734	1.832	1.923
14	1.127	1.108	1.106	1.699	1.513	1.558
15	1.080	1.261	1.156	1.447	2.098	1.582
16	1.052	1.123	1.089	1.167	1.544	1.226
17	1.057	1.019	0.981	1.162	1.148	1.043
18	1.007	1.123	1.105	1.148	1.224	1.321
19	1.016	1.083	1.070	1.085	1.128	1.124
20	1.008	0.995	0.972	1.079	0.935	0.975

that is excited at  $s$  and received at  $r$  in frequency domain. When I define my medium is lossless, I can apply the principle of time-reversal invariance and I can replace the source and receiver. Therefore, I can get this equation,

$$u(s_A, s_B) = \int_r u^*(r, s_A) u(r, s_B) dr. \quad (\text{II.3.3})$$

I can reduce the noises that are not useful to pseudo shot gather because of applying integral all receivers. I show the concept of this equation (Figure II.3.8). When I have a few receivers, it is really important that I have a wave of stationary phase (Snieder et al., 2006). Because I can not recognize well before taking integral, it is impossible to analysis when I have one receiver. (Mehta et al., 2008b) is using virtual source method that are similar or completely same as seismic interferometry, for time-lapse monitoring whose receivers are also permanently placed OBC. Although using virtual source

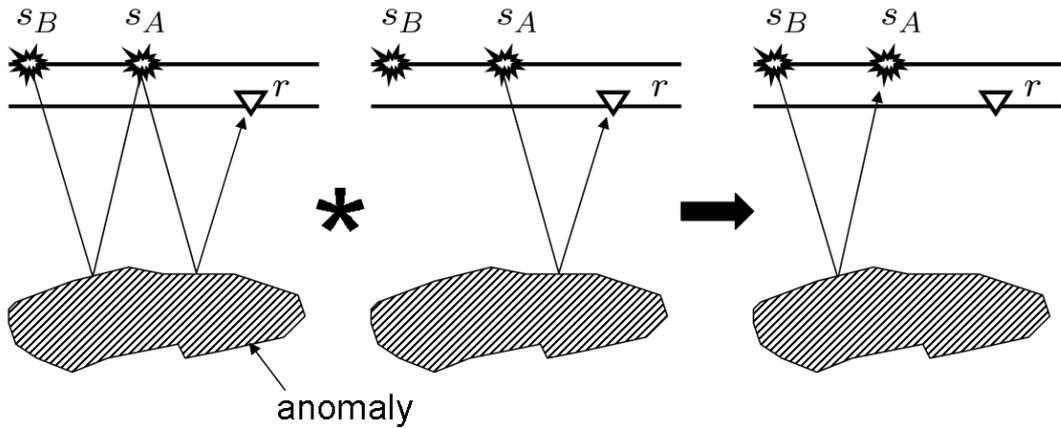


Figure II.3.8: The concept of seismic interferometry using multiples for construct the pseudo shot record.

method, it is not same as my idea. (Mehta et al., 2008b) is redatuming the data down to the OBC line by virtual source method for getting high quality data without the effect of seawater. Although the purpose of using virtual source method is different, it seems useful the techniques of combination of up-down wavefield separation, and time-windowing the direct arrival. In other words, I can say virtual receiver method in my idea whose concept is not so different from virtual source method. If I put a receiver in the well, I would get really interest result that are high resolution.

## Chapter II.4

### Conclusion

I estimate the number of requisite receivers for time-lapse monitoring seismic survey of  $CO_2$  injection by using ray tracing simulation. Furthermore, I make indicators that are helpful for survey design and improve the NRMS to DNRMS. Needless to say, the number of receivers is depended on the amount of noise. In noise free simulation, I can detect anomaly even if this anomaly is quite small. When I have many receivers, I can stack hundreds of times and reduce the effect of white noise. Eventually, the difference of either detectable or undetectable is depended that I can or can not retrieve the anomaly waves that are covered by noise. The best way of reduction of white noise is stacking many times. Thus, I think it become improving that I shot very many times. This is depended on the cost. Although the semi-permanently placed OBC is takes much initial cost, the running cost is not so much. However, If I take many shots in each survey, the running cost is getting high. I think seismic interferometry is one of valid methods for reducing receivers.

Comparison from synthetic model, realistic model is really difficult to detect the anomaly. Certainly, it is different the amount of injected  $CO_2$ . Thus, if I have simple structure, I can choose small number of receivers even if there is much noisy. I think the importance of simulation increases by likely in these surveys because I can not failure.

Velocity error has critical influence for imaging 3D seismic volume. Two-dimensional averaging value of NRMS reduces about 30% by velocity error. Because I can not know the exact velocity, I

should take it into account. In fact, I think 95% velocity is good accuracy because this range is easy to vary when I do velocity picking.

I should use full wave simulating method like FDTD if I can because ray tracing has many limits; for example, it can not simulate diffraction waves and all multiples. By simulating FDTD, I can get better results that are almost reality. I should test the indicator of predictability and define the number of receivers before  $CO_2$  injection in fact.

## **Part III**

# **Three-dimensional elastic wave propagating simulation using FDTD method on GPU**

## Chapter III.1

# Introduction

In geophysics, it is very important to simulate wave behavior because no one can look subsurface structure directly. Simulation is a helpful tool to understand the phenomenon. Many researchers have used wave propagating simulator for this reason. For example, (Juhlin, 1995) made structure model from geologic apriori information and simulating wave propagation for understand the response of seismic wave. The applications of simulation are seen in various fields, e.g. setting receiver array or source point, for inversion, for migration, confirming the effect of noise, and verifying the appropriateness of new method.

Solving wave equation analytical is a difficult thing to do, even impossible for almost cases. Only for simple situation can be done analytically, e.g. (Saito, 1993). I, however, would like to simulate of waveform through complex structure, which represents the real subsurface structure will be presented. It is not significant that simulating model is completely different from real structure. From this stance, I should simulate the wavepropagation through a three-dimensional (3D) model, not a two-dimensional (2D) model. It is already know that there are many limitations for simulation, which are the accuracy, computation time, computation resource, making structure model. In this part I describe a parallelization of the 3D elastic wave propagation simulation by using Finite-Difference Time-Domain (FDTD) method. The simulation usually calculates wave equation or two equations that construct the wave equation. These equations are displacement-stress formulation and equation of motion. I use stag-

gered grid method to calculate those equations alternately and this method has enough accuracy for geophysical usage. The purpose of this part is shortening the computation time while still keeping the accuracy.

When I want to simulate wave phenomena for 3D modeling, I usually use ray tracing algorithm. In the ray tracing, the wave propagation is approximated high frequency wave and I can estimate wave propagation along the raypaths through a model. (Cerveny, 2001; Vinje et al., 1999) This modeling process can be calculated in small computation resources and only needs shorter time, that is valid for practical usage, comparing than simulating full waves. Ray tracing, however, can of course not construct all wavefields, such as modeling surface waves or the variation of frequency.

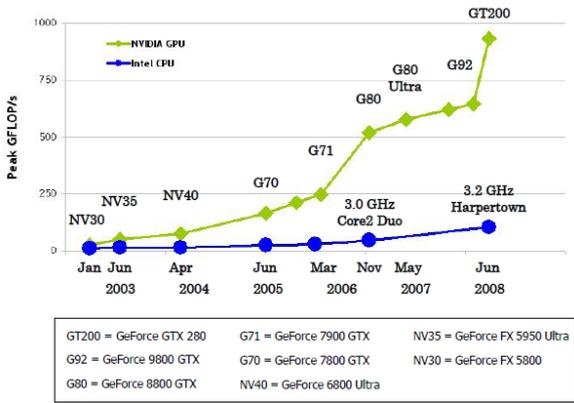
On the other hand, the modeling method using wave theory has been developed. One of these methods is FDTD. By using FDTD, I can simulate all wavefield including surface wave and body wave. Although the algorithm of FDTD is easy, I need much computation resource. In spite of the rapid development of the performance of computer, I can not finish the calculation of all data set in ordinary geophysical exploration using 3D FDTD method in practicable time by keeping enough accuracy. For example, I assume a whole 3D survey that has 1000 shots, 1000 receivers and computation area is  $2 \times 2 \times 2km$  whose grid size is  $5 \times 5m$  and all grids are  $6.4 \times 10^7$  grids. If I would like to simulate this survey by FDTD, I need over 12 hours by one shot using 1 CPU core in my program. This computation time depends on the amount of output data. It will take 500 days that I calculate all shots. It is also difficult to finish in practicable, which is in one or two days if you parallelize this calculation using many CPU cores. Therefore, I propose the usage of an accelerator for calculation that you usually have in your machine.

I use Graphic Processing Unit (GPU) for accelerating calculating time via NVIDIA's CUDA. Usually GPU handles graphic data. The calculation speed of GPU is improved quite faster than one of CPU (Figure III.1.1(a), (NVIDIA, 2009)). The GPU technique for scientific calculation is developed in early 2000's, which is then called as General-Purpose computing on GPU (GPGPU) (Owens et al., 2005). In the beginning era of development, the GPGPU was used to solve a "many-body" problem in

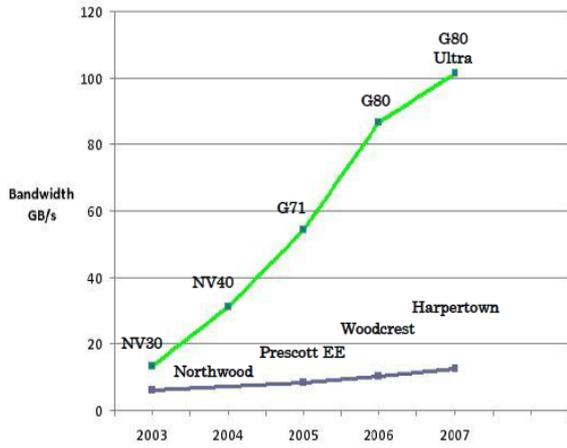
astronomy. GRAPE is famous special-purpose computing board for this problem. In that day, GPGPU is used for particle method, molecular dynamics and fluid mechanics. This GPGPU programming was actually quite difficult to be implemented. Thanks for CUDA, so that I can use the GPU for simulation in easy way. CUDA was launched in 2007 from NVIDIA, which is one of programming language that is similar to C programming language. GPU programming is also very useful for geophysical problem. Geophysicists use GPU for migration of seismic data (Moussa, 2009; Abdelkhalek et al., 2009; Bo et al., 2009), and for acoustic, magnetic or electric wave propagation simulation (Micikevicius, 2009; Takada et al., 2008) and visualization and interpretation (Kadlec et al., 2009).

The characteristics of GPU computing are high performance, wide memory bandwidth (Figure III.1.1(b)), conservation of space and electric power, cheap and easy to be set. The GPU that has about ten times higher flops than CPU. The wide memory bandwidth is another important thing to be considered. In FDTD program, it usually takes almost time for data writing and reading. This is the reason why Intel Core i7 CPU can improve computation time in many program (BARKER et al., 2008). Although Core i7 has wide memory bandwidth, GPU has wider memory bandwidth than Core i7 CPU. Because of conservation of space, electric power, and price, I can calculate some programs in a laptop machine. Whenever one bring a laptop, one can simulate something by the GPU on laptop. Price is also one of the biggest consideration for users. I can get over 1 TFlops performance under 100,000 yens.

By accelerating of GPU, GPU has a potential to realize this calculation in practical time and I may simulate in my laptop computer. In this part, I show the acceleration of FDTD method by using GPU for 3D elastic wave propagating simulation.



(a) Computing performance



(b) Memory Bandwidth

Figure III.1.1: Comparison of (a) computing performance and (b) memory bandwidth between GPU and CPU. Green and blue lines denote GPU and CPU, respectively.

## Chapter III.2

# FDTD computation by staggered grid on GPU

### III.2.1 Programing for CUDA

The hardware implementation of GPU is quite different if it is compared to CPU. GPU has hundreds processors and a variety of memories that are differed in the volume and bandwidth (Figure III.2.1). Some of processors make a group, which is then called as multithreaded streaming multiprocessor (SM). Although the clock rage of processor in GPU is approximately  $1.0GHz$ , which is not a high clock, but GPU contains over 100 processors and I can parallelize my program. There are several types of memories on GPU (on-chip memory) and Device memory on GPU board (off-chip memory). Because GPU cannot read system memory that is situated on the mother board, I must have additional data copy between system memory and device memory. Data copy from device memory to shared memory is also additional task for GPU computing. Thus, I should reduce the number of data copy. Device memory is a few *GB* capacity and storing data are used by all processors. Although the access speed of device memory is faster than system memory, its speed is not enough to simulate wave propagation. Memories on GPU are quite faster and I can use these memories like cache memory on

CPU. Difference from CPU programming is that I should declare how many on-chip memory I want to use. The usage of memories, especially shared memory, is the key point for accelerating processes. Because shared memory is quite fast and has only 16 KB, I have to define how much data could be stored in the memory.

In CUDA programming, CPU acts as a host that controls GPU, and GPU acts as a device. I need to write two type of codes namely CPU (host) code and GPU (device) code. These codes are similar to C programming code. Host code calls kernel function, whose contents is stored in device code, and uses GPU. All processors on GPU execute same device code. The concept of GPU programming is shown in Figure III.2.2. One kernel function makes one grid that includes some blocks. One grid executes on one GPU board and one block is corresponded to one multiprocessor's calculation range. Thread is execution unit of device code. I can parallelize by executing program simultaneously in many processors. For example, because NVIDIA Tesla C1060 board has 240 processors, I can accelerate 240 times speedup theoretically. In fact, I can find some papers that accelerate over 100 times using GPU (2009,Fang,Boas.pdf, 2009,Wong,Wong,etal.pdf, 2008,Meel,Arnold,etal.pdf). It is, however, difficult to accelerate all type of codes because of low computational clocks of one processor and additional memory copy. Thus, I should adapt my code for CUDA for getting effective accelerating.

## III.2.2 Example of CUDA

CUDA is applied in many categories and propounded the good results ((NVIDIA, 2009), (Aoki and Nukada, 2009), (Nguyen, 2007)). I show simple example of CUDA programming, e.g. particle motion simulation using Runge-Kutta method in order to show the advantage of GPU. It has been used NVIDIA Tesla C1060 as the device. I decide velocity that varies by position and, time and particles move according to this velocity value. I set many particles to a round shape for initial model and

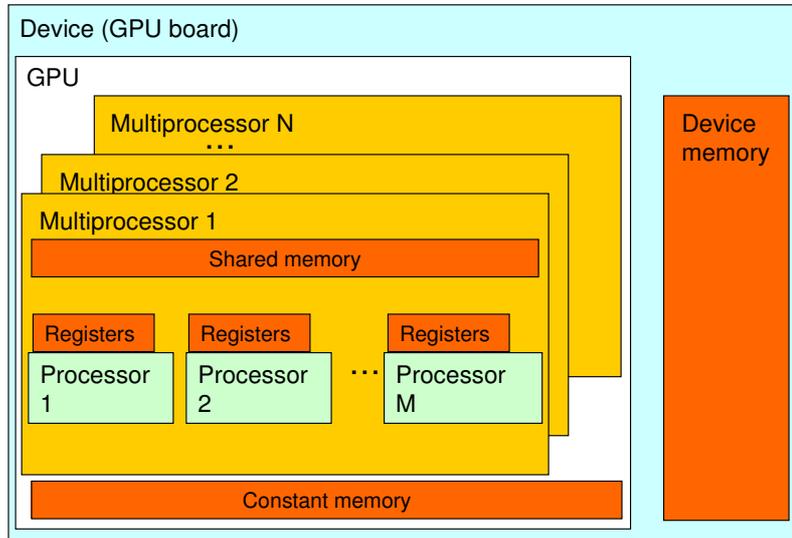


Figure III.2.1: Hardware implementation of GPU. Device is connected at PCI Express. GPU is the computation unit on the GPU board and includes processors (green) and memories (orange). Device memory could be found outside of the GPU.

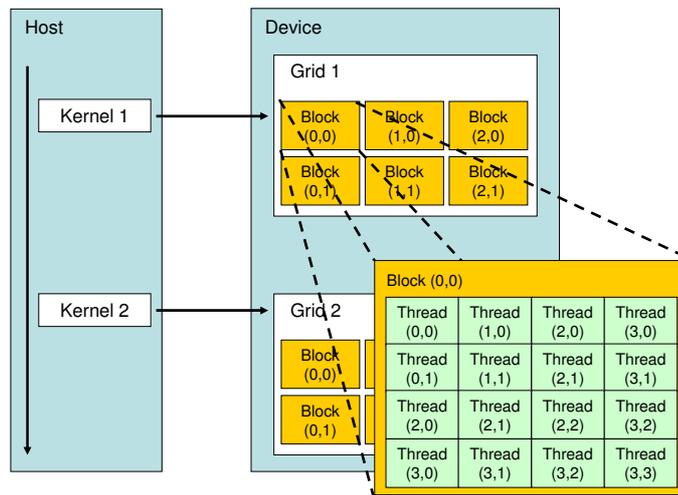


Figure III.2.2: The concept of GPU programming. The colors in this figure corresponds to the same explanation as stated in Figure III.2.1. One thread is calculated at one processor, one block is being input to one multiprocessor.

calculate the position of particles in each time. I set the velocity field,

$$\begin{aligned} u(x, y, t) &= -\cos\left(\frac{\pi t}{T}\right) \sin(\pi x) \cos(\pi y) \sin(\pi y) \\ v(x, y, t) &= -\cos\left(\frac{\pi t}{T}\right) \sin(\pi x) \cos(\pi y) \sin(\pi y), \end{aligned} \quad (\text{III.2.1})$$

$u(x, y, t)$  and  $v(x, y, t)$  is the x-direction and y-direction velocity at  $(x, y)$  position of time  $t$ , respectively.

Thus, the differentiation of position of each particle is

$$\begin{aligned} \frac{dx_i}{dt} &= u(x_i, y_i, t) \\ \frac{dy_i}{dt} &= v(x_i, y_i, t), \end{aligned} \quad (\text{III.2.2})$$

$i$  denotes the sequential number of particles. Equation III.2.2 is normal differential equations and it is difficult to solve analytically except  $u$  and  $v$  is quit simple case. Thus, I use fourth-order Runge-Kutta method for solving Equation III.2.2. Fourth-order Runge-Kutta method is

$$\begin{aligned} x_i^{n+1} &= x_i^n + \frac{p_1 + p_2 + p_3 + p_4}{6} \Delta t \\ y_i^{n+1} &= y_i^n + \frac{q_1 + q_2 + q_3 + q_4}{6} \Delta t; \end{aligned} \quad (\text{III.2.3})$$

$p_1$  through  $p_4$  and  $q_1$  through  $q_4$  is

$$\begin{aligned}
 p_1 &= u(x_i^n, y_i^n, t^n) \\
 p_2 &= u\left(x_i^n + \frac{1}{2}p_1\Delta t, y_i^n + \frac{1}{2}q_1\Delta t, t^n + \frac{1}{2}\Delta t\right) \\
 p_3 &= u\left(x_i^n + \frac{1}{2}p_2\Delta t, y_i^n + \frac{1}{2}q_2\Delta t, t^n + \frac{1}{2}\Delta t\right) \\
 p_4 &= u(x_i^n + p_3\Delta t, y_i^n + q_3\Delta t, t^n + \Delta t) \\
 q_1 &= v(x_i^n, y_i^n, t^n) \\
 q_2 &= v\left(x_i^n + \frac{1}{2}p_1\Delta t, y_i^n + \frac{1}{2}q_1\Delta t, t^n + \frac{1}{2}\Delta t\right) \\
 q_3 &= v\left(x_i^n + \frac{1}{2}p_2\Delta t, y_i^n + \frac{1}{2}q_2\Delta t, t^n + \frac{1}{2}\Delta t\right) \\
 q_4 &= v(x_i^n + p_3\Delta t, y_i^n + q_3\Delta t, t^n + \Delta t).
 \end{aligned}$$

I show the snapshots of this calculation (Figure III.2.3). Then, I compare the computation time between GPU and CPU (Table III.2.1). GPU takes about 126.8 - 251.1 times shorter than CPU. This simulation

Table III.2.1: The comparison of computation time between CPU and GPU. This computation time do not include the data output time.

Number of particles	0.01 million		0.04 million		1 million		4 million	
time steps	1000	10000	1000	10000	1000	10000	1000	10000
CPU (sec)	8.467	89.87	33.87	359.7	860.5	8994.0	3386.0	35960.0
GPU (sec)	0.0668	0.668	0.182	1.82	3.606	36.07	14.31	143.2
accelerating ratio	126.8	134.5	186.1	197.6	238.6	249.3	236.6	251.1

is suitable for GPU computing because this simulation does not use big memories and does not need much synchronization that is synchronize the data between blocks for calculating next step. In small model which is 10,000 particles, the difference of computation time between CPU and GPU is smaller than the big model. This is because, it takes much time for data coping relatively and I can not take advantage of GPU parallelization well. I also simulate two type of time steps, but it is not contributed to the accelerating ratio very much. I may, therefore, conclude that the performance of GPU programming depends on the computing area rather than time steps in this simulation. When I calculate small area,

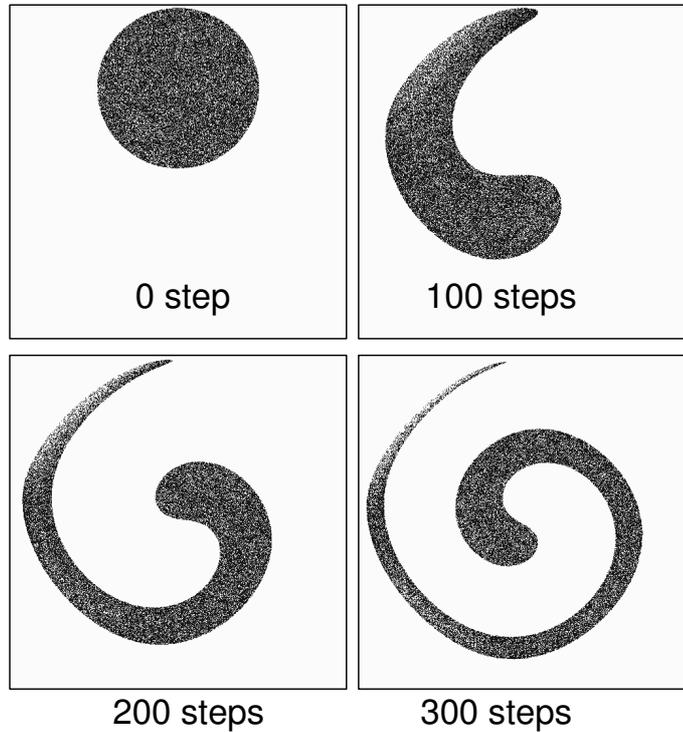


Figure III.2.3: Snapshots of particle motion. This figure shows 40,000 particles inside the model for several time steps. Black denotes the area inside the model, in which the particles are displaced.

the performance of GPU programming degrades.

### III.2.3 Mounting FDTD on GPU

I simulate 3D elastic wave propagation using FDTD by CUDA. I show the equation of FDTD method is shown in Appedix A. 3D elastic wave equation has 9 variables, which are 3 velocity factors and 6 stress factors. I should also set the model parameters that are density, P wave velocity and S wave velocity. Because these 12 factors are decided on each grid, I prepare at least  $4bytes(float) \times$

$12 \times modelgrids$  memories and take synchronization every time steps. Thus, I need big memory and many times synchronizations that are not suitable for GPU programming. I also take care the usage of shared memory that can store only about 4,000 float numbers. If I want to store all factors in shared memory, I can store 333 grids data by one multiprocessor. Because of shared memory limitation, I can set the block size lesser than  $8 \times 8 \times 5$  grids. In this algorithm, I need 13 grids' value to calculate one points (Figure III.2.4). If I want to calculate the next time step value of all shared memory's area, I must use device memory's value because the edge of grid in shared memory requires the outside value of shared memory. It is, however, so slow and I should complete the calculation using only on the shared memory data. For resolution of this slowness, I input surplus two grids (wing area) in each direction to shared memory; I show the shape of input data (Figure III.2.5). I can calculate the next timestep value in  $4 \times 4 \times 2$  grids by each block using this shape of input data. I show the computation time for comparing the effectiveness of shared memory usage. I test the memory by 2D elastic wave propagating simulation because it takes long computation time to calculate all test patterns by 3D simulation. Test items are using only device memory (Case A), using both device and shared memory (Case B) that means shared memory are not copied wing area, and using only shared memory (Case C) for calculating each timestep. These test items differ from the number of reading device memory. Case A is reading device memory by every calculations, Case B is reading this memory when I calculate the edge of grid in shared memory, and Case C is reading two times that are the start and end of timestep. Table III.2.2 is the computation time of CPU for comparing the time between GPU and CPU and Table III.2.3 is the computation time of GPU and accelerating ratio. The time step of both Table III.2.2 and III.2.3 is 1,000 steps.

Table III.2.2: Computation time of 2D elastic wave propagating simulation on CPU

Number of grids	$480 \times 480$	$960 \times 960$	$1440 \times 1440$	$1920 \times 1920$
computation time (sec)	24.08	90.44	189.1	324.9

I show the computation time of each case that show only fastest time and do not consider the differ-

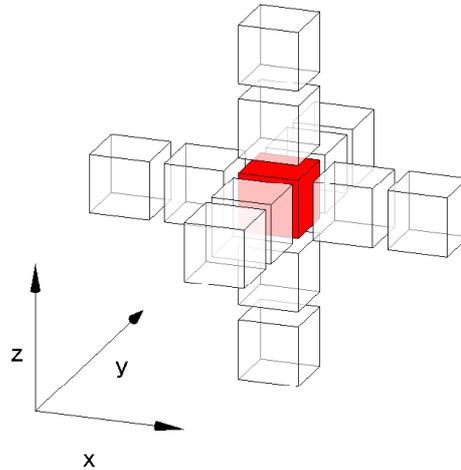


Figure III.2.4: Thirteen grid positions for computing one grid on the center of grids of next step. A cube shows one grid. Thirteen grids are needed (white and red cubes) for calculating one grid of next step (red cube). Two times from 13 grids to 1 grid for updating the data which calculated velocity and displacement, respectively should be simulated.

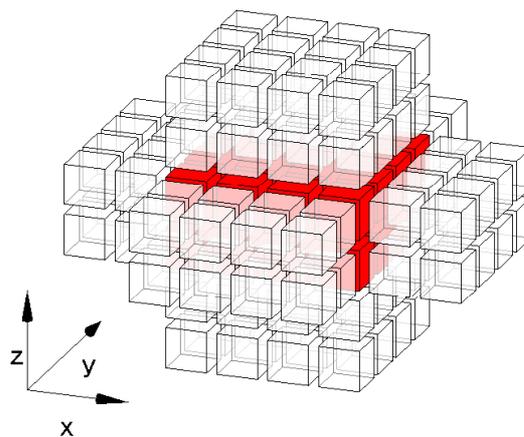


Figure III.2.5: Copied grids to shared memory including wing area. White cubes are wing area's cube and red cubes are calculating area for next time steps. The block size is  $4 \times 4 \times 2$ . The number of grids of wing area are 4 times bigger than one of calculating area.

Table III.2.3: Computation time of 2D elastic wave propagating simulation on GPU

Number of grids	Case	blocksize		
		$4 \times 4$	$8 \times 8$	$16 \times 16$
$480 \times 480$	A	3.152 (7.640)	4.586 (5.251)	7.343 (3.280)
	B	6.585 (3.657)	3.222 (7.476)	2.822 (8.536)
	C	3.127 (7.703)	2.573 (9.362)	2.462 (9.784)
$960 \times 960$	A	13.50 (6.699)	16.13 (5.605)	32.04 (2.822)
	B	25.28 (3.577)	12.56 (7.203)	10.08 (8.974)
	C	11.99 (7.542)	6.953 (13.01)	9.984 (9.058)
$1440 \times 1440$	A	35.10 (5.387)	44.26 (4.272)	88.72 (2.131)
	B	56.79 (3.329)	31.20 (6.060)	22.84 (8.279)
	C	29.32 (6.448)	14.38 (13.15)	23.56 (8.025)
$1920 \times 1920$	A	71.89 (4.519)	67.05 (4.846)	195.0 (1.666)
	B	100.0 (3.249)	57.52 (5.649)	40.72 (7.980)
	C	54.95 (5.912)	22.72 (14.30)	41.31 (7.865)

Case A is using only device memory, Case B is using both device and shared memory and Case C is using only shared memory. The unit of these values is second and parenthesis value is accelerating ratio between GPU and CPU computation. The underline shows the fastest time in each grid.

ence of blocksize (Figure III.2.6). Next, I show the computation time of 3D elastic wave propagating simulation that is using shared memory like Case C, because Case C is fastest in every grid sizes in 2D. Blocksize is  $4 \times 4 \times 2$ , which is the fastest blocksize.

Table III.2.4: Computation time of 3D elastic wave propagating simulation

Number of grids	$96 \times 96 \times 96$	$192 \times 192 \times 192$	$288 \times 288 \times 288$	$384 \times 384 \times 384$
CPU (sec)	220.04	1737.7	8603.4	14790.0
GPU (sec)	46.108	398.22	1378.7	2628.9
accelerating ratio	4.772	4.364	6.240	5.626

I show the snap shots of 3D elastic wave propagating simulation (Figure III.2.7), source is ricker wavelet whose peak frequency is  $7.5Hz$  at  $(x,y,z) = (250,250,0)$ . The model size is  $500 \times 500 \times 250$  grids that is assumed  $2.5 \times 2.5 \times 1.25km$  and additional 40 grids for absorbed boundary. I adopt  $\alpha = 0.007$  and 40 grids for absorbed parameter in Equation A.16. This model is flat horizontal two layered model and this horizon is exist at  $z = 120$  and  $z = 0$  is free surface. I set receiver line at  $(y,z) = (250,0)$  and shot gather, which would be seen in Figure III.2.8. I can see P and S wave, and

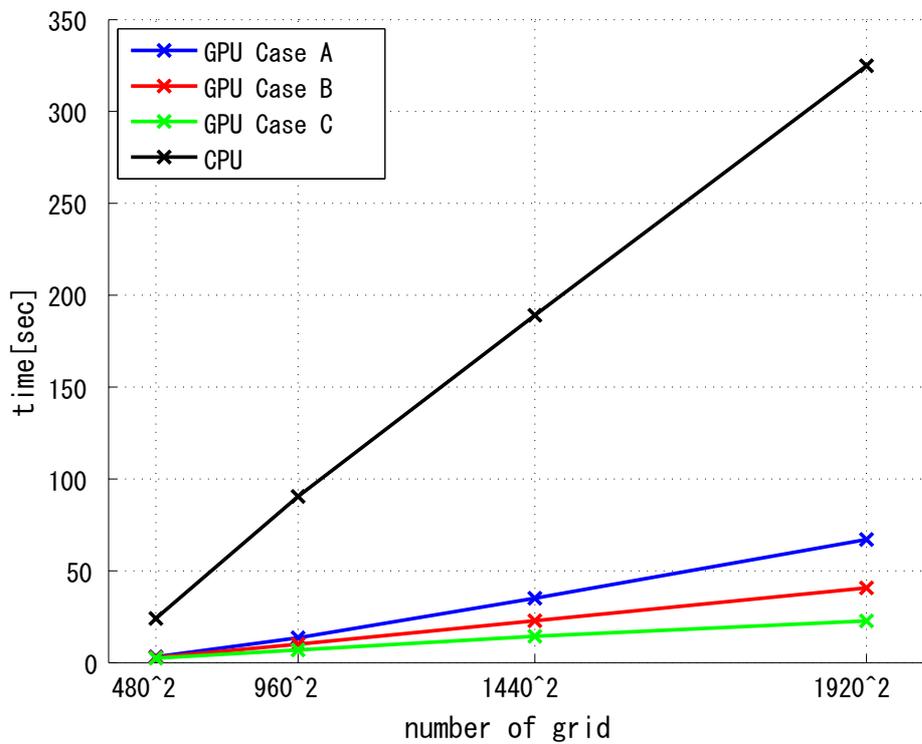


Figure III.2.6: Computation time of 2D elastic wave propagating simulation. GPU Case A (blue line) used only device memory, GPU Case B (red line) used both device and shared memory and GPU Case C (green line) used only shared memory. The value of this plot is shown in Table III.2.3. The fastest time of each grid and case without distinction of blocksize is chosen.

surface waves. I compare this result and analytic solution of Z-component displacement for confirming the appropriateness of this simulator (Figure III.2.9). This analytic solution is derived by (Saito, 1993). I estimate the error, that shows the accuracy of this simulator, between simulation result and analytical solution. I define the error

$$E = \frac{\sum_{i=t} (u_i - u_i^a)^2}{\sum_{i=t} (u_i^a)^2}, \quad (\text{III.2.4})$$

$u_i$  is displacement value of simulator and  $u_i^a$  is the value of analytical solution. The error of Figure III.2.9 is 2.174% and this error show my simulator keeps enough accuracy.

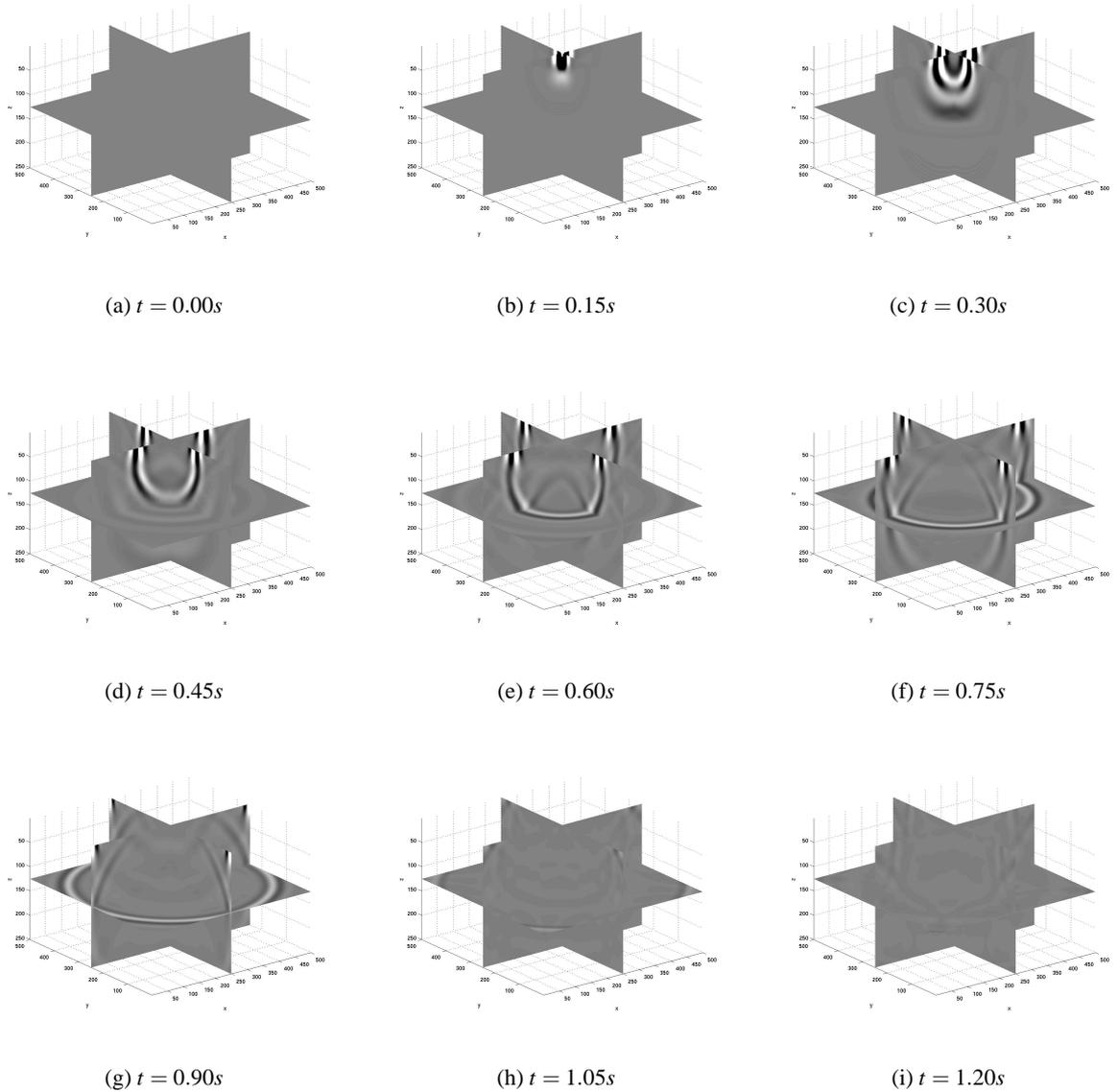


Figure III.2.7: Snapshots of 3D elastic propagating simulation by 0.15 sec timestep. On the depth at  $z = 120$  one can see the reflection wave of P wave at (d) and S wave at (e).

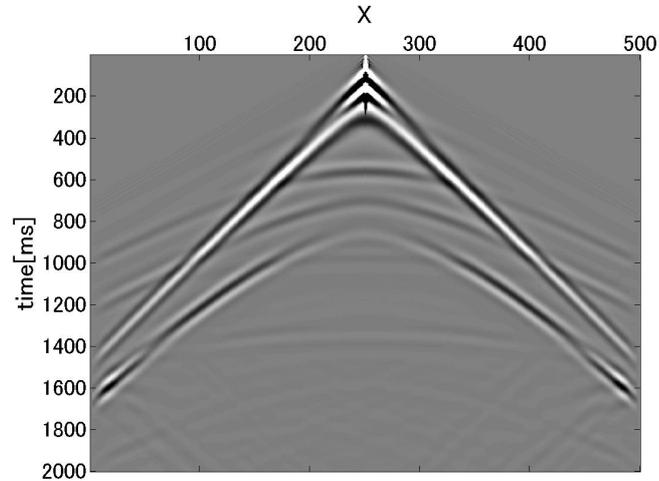


Figure III.2.8: Shot gather of Figure III.2.4. The receivers are set at  $(y, z) = (250, 0)$  and every 5  $x$  grid of 25m.

I show a application using this simulator.  $CO_2$  injection to subsurface structure is a hot issue. I should monitor the condition of  $CO_2$  to protect the leak. The influential method of monitoring is using seismic wave. Thus, I assume the  $CO_2$  injection and confirm the difference between before and after injection. The model is smoothing curve 7 layers model that is point symmetry  $(x, y, z) = (2km, 2km, 1km)$  3D model, and 5 layer is assumed an aquifer (Figure III.2.10). I show shot gathers whose source point is at the center of surface and receiver line is  $y = 1km$  of surface (Figure III.2.11). I can simulate P and S waves and reflected wave from each layer boundary. The difference clearly shows the wave from injection point clearly.

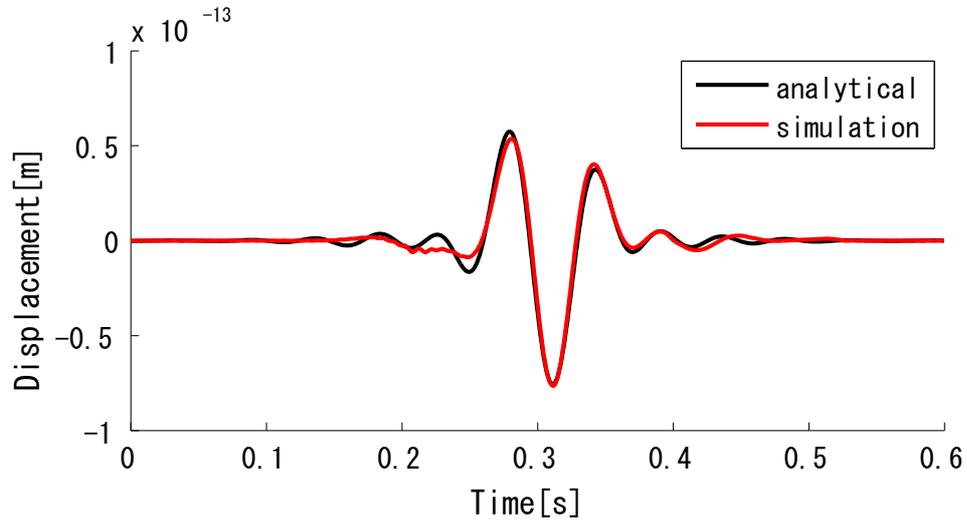


Figure III.2.9: Comparison of Z-component displacement between simulation result and analytic solution. Black line is analytical value and red line is simulating value. The distance between source and receiver point is 1200m.

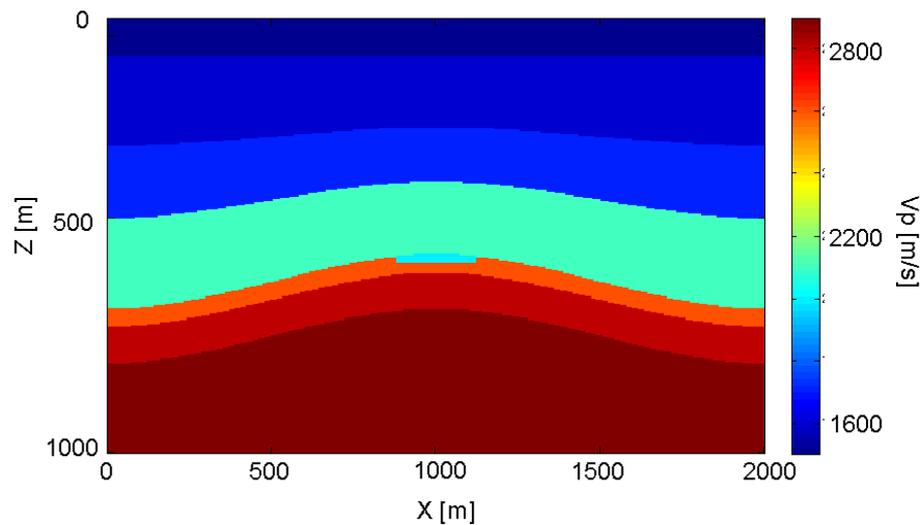
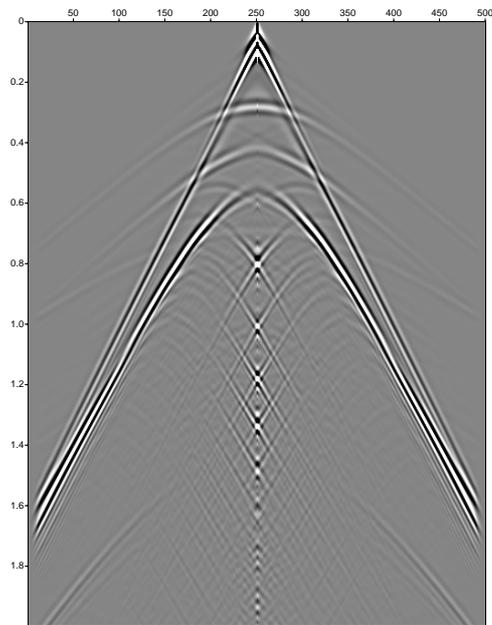
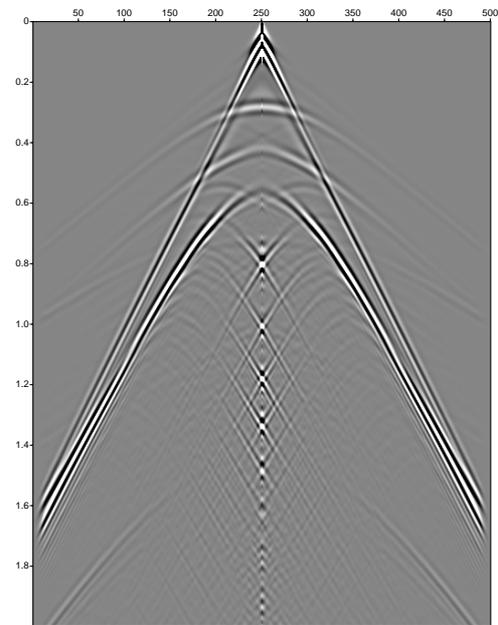


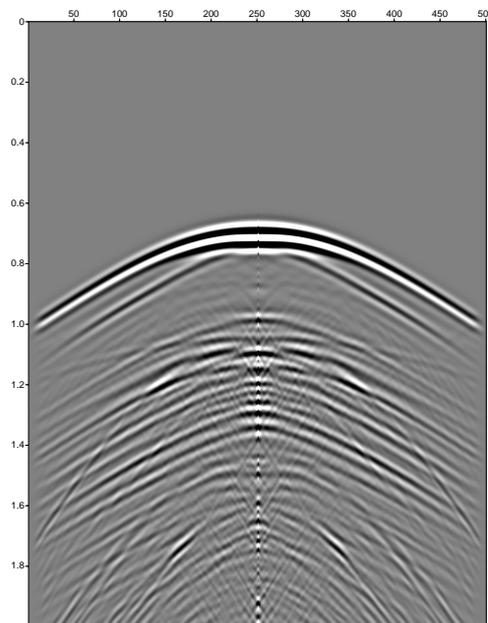
Figure III.2.10: The slice of model for  $CO_2$  injection simulation. The sky blue area in middle of the model shows  $CO_2$ .



(a) before injection



(b) after injection



(c) difference

Figure III.2.11: The shot gather of receiver line. (a)before injection, (b)after injection, (c)difference of (a) and (b). The amplitude of (a) and (b) is same, and (c) is quite small.

## Chapter III.3

### Discussion

Table III.2.3 proved that the usage of shared memory is important factor for computing time. When I only use device memory, I must access device memory so many times that are same times of the number of every terms of FDTD formula. Because the access speed of device memory is about 100 times slower than one of shared memory, I should use shared memory as possible as I can. I can see the availability of shared memory in Case B. The access ratio of device memory and shared memory shrinks as I make blocksize big.  $16 \times 16$  blocksize simulator accesses the least times to device memory and the fastest computation time at every grids. Moreover, Case C accesses only two times to shared memory and Case C is the fastest. The blocksize also influences the communication traffic volume by one access. Device memory bandwidth is used most efficiently when the simultaneous memory accesses by 16 threads, which can be coalesced into a single memory transaction of 32, 64, or 128 bytes. When I would like to copy 4 bytes data from device memory, it is not efficiently but I must copy 32 bytes data. Memory copy time of not coalescing is about 10 times slower than on of coalescing. Thus, I should access device memory sequentially for coalescing. The order of loop influences on the coalescing. If I can coalesce memory copy by coping x-direction, I may not coalesce by coping y-direction. Data are stored memory linearly. Because the data is stored on memory by  $(x,y) = (1,1), (2,1), (3,1), \dots, (n,1), (1,2), (2,2), \dots$ , coalescing is happened only on one direction. Consider the above result, I optimize the coalescing by order of loop. I show the example that change the order

of loop in 2D elastic wave propagating simulation using both shared and device memory (Case B) and blocksize of  $16 \times 16$  (Table III.3.1). I see computation time is improved 2 times by coalescing. Because coalescing may be done by big data, which is 64 or 128 bytes, the accelerating ratio is bigger in many grids.

Table III.3.1: Comparison of computation time of 2D elastic wave propagating simulation by coalescing

Number of grids	$480 \times 480$	$960 \times 960$	$1440 \times 1440$	$1920 \times 1920$
not coalescing (sec)	4.777	20.02	48.69	100.2
coalescing (sec)	2.822	10.08	22.84	40.72
accelerating ratio	1.693	1.987	2.132	2.461

Because of the big memory copy between memories, the 3D elastic wave propagating simulation has smaller accelerating ratio than 2D simulation and particle motion simulation. The additional memory copy takes more than half time of computation time (Table III.3.2).

Table III.3.2: Additional memory copy time by GPU computing of 3D elastic wave propagating simulation

Number of grids	$96 \times 96 \times 96$	$192 \times 192 \times 192$	$288 \times 288 \times 288$	$384 \times 384 \times 384$
time (sec)	31.738	271.95	1047.8	1898.4
percentile (%)	68.83	68.29	75.99	72.20

time is additional memory copy time and percentile is percentage of additional memory copy time in whole computation time

Because the ratio of additional memory copy is about 70% in this simulation, I can assume that the calculating time may be so small on GPU. This relationship between memory copy time and calculating time follows Amdahl's law that says the maximum expected improvement to an overall system when only part of the system is improved. Amdahl's law is shown as

$$\frac{1}{(1-P) + \frac{P}{S}}, \quad (\text{III.3.1})$$

P is improvement proportion of the whole computation area and S is speedup ratio. The answer of this expression shows the ratio of speedup of whole system. Although I usually improve the algorithm of calculation for improving the computation time, I should improve the method of memory copy in this case because memory copy proportion is much bigger than calculating proportion. This ratio may be getting small in 2D simulation because 2D simulation can take big blocksize. Because 3D simulation has 13 parameters and 3D grids, I can not take big blocksize and I must copy more surplus area as wing area. Thus, 2D simulation can take memory copy more efficiently. Particle motion simulation takes tiny memory space and certainly this simulation's ratio of additional memory copy is small. This is the reason that this simulation can accelerate over 200 times from CPU. These additional memory copies do each timestep for synchronizing data.

I would have some rooms for optimization in wave propagating simulation. For example, I tune device memory access for doing coalescing every time, and for reducing the data synchronizing, I copy the data from device memory to shared memory including more surplus area.

## Chapter III.4

### Conclusion

I can get over 14 times accelerating in 2D elastic wave propagating simulation and over 6 times accelerating in 3D elastic wave propagating simulation by using GPU. Because I have usually two or four cores in one CPU in my personal computer, I can possibly accelerating 3 or 4 times by parallelizing. If I would like to finish the simulation at shorter time, I must use network like gigabit ethernet for parallelizing. The bandwidth of gigabit ethernet is slow and it is not suitable for requiring accuracy because the concept of ethernet is best effort and could not support the bit error. Thus, it is significant that I can accelerate over 4 times.

I use shared memory as possible as I can in GPU computing because shared memory is quite faster than device memory. Other key points of accelerating computing speed are blocksize and coalescing. Eventually, I should improve the usage and communications of shared and device memory, this is based Amdahl's law. The potential of GPU is enough for completing the 3D elastic wave propagating simulation in practicable time because GPU can improve the computation time over 200 times in particle motion simulation. The possible accelerating ratio of wave propagating simulation is been able to approach one of particle motion simulation.

GPU device and CUDA is always improving. I will get new device and compiler in 2010, this device has big memory and supports double precision well, the new compiler would have strengthening synchronizing system and flexible shared memory usage. For practical usage of 3D elastic wave

propagation by FDTD, the key hardwares of my opinion are GPU, SSD (Solid State Drive). Due to computation hardware and software that is getting enhanced, I must take care and improve my programming skill.

## Appendix

### Staggered grid

In FDTD method, staggered grid calculation is one of most famous methods for solving wave equation (Virieux, 1986; Graves, 1996). It can be derived the wave equation from displacement-stress formulation and equation of motion, as their expressions are written as follow; These two expressions are

$$\tau_{ij} = \lambda \frac{\partial u_k}{\partial x_k} \delta_{ij} + \mu \left( \frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} \right) \quad (\text{A.1})$$

$$\rho \frac{\partial v_i}{\partial t} = \frac{\partial \tau_{ij}}{\partial x_j} + f_i, \quad (\text{A.2})$$

$i, j = 1, 2, 3$ ,  $\tau$  is stress,  $\rho$  is density,  $u$  is displacement and  $f$  is body force. A.1 is displacement-stress formation and A.2 is equation of motion. Because 3D description is complex, only the expressions for 2D is written here. The equation written in A.1 can then be rewritten into;

$$\frac{\Delta \tau_{xx}}{\Delta t} = (\lambda + 2\mu) \frac{\Delta v_x}{\Delta x} + \lambda \frac{\Delta v_z}{\Delta z} \quad (\text{A.3})$$

$$\frac{\Delta \tau_{zz}}{\Delta t} = \lambda \frac{\Delta v_x}{\Delta x} + (\lambda + 2\mu) \frac{\Delta v_z}{\Delta z} \quad (\text{A.4})$$

$$\frac{\Delta \tau_{xz}}{\Delta t} = \mu \left( \frac{\Delta v_x}{\Delta z} + \frac{\Delta v_z}{\Delta x} \right) \quad (\text{A.5})$$

and the equation written in A.2 can be rewritten into;

$$\rho \frac{\Delta v_x}{\Delta t} = \left( \frac{\Delta \tau_{xx}}{\Delta x} + \frac{\Delta \tau_{xz}}{\Delta z} \right) + f_x \quad (\text{A.6})$$

$$\rho \frac{\Delta v_z}{\Delta t} = \left( \frac{\Delta \tau_{xz}}{\Delta x} + \frac{\Delta \tau_{zz}}{\Delta z} \right) + f_z. \quad (\text{A.7})$$

Because denser data prefer to be set, staggered grid arranges displacement and velocity to different position whose interval is  $\Delta x/2$ ,  $\Delta y/2$ ,  $\Delta z/2$ , respectively. The scheme of second-order finite difference in the time domain can be developed, so that the Equation A.3 through A.5 and A.6 through A.7 become

$$\begin{aligned} \frac{\tau_{xx}^{t+\Delta t/2}(i, j) - \tau_{xx}^{t-\Delta t/2}(i, j)}{\Delta t} &= (\lambda + 2\mu) \frac{v_x^t(i + \Delta x/2, j) - v_x^t(i - \Delta x/2, j)}{\Delta x} \\ &\quad + \lambda \frac{v_z^t(i, j + \Delta z/2) - v_z^t(i, j - \Delta z/2)}{\Delta z} \end{aligned} \quad (\text{A.8})$$

$$\begin{aligned} \frac{\tau_{zz}^{t+\Delta t/2}(i, j) - \tau_{zz}^{t-\Delta t/2}(i, j)}{\Delta t} &= \lambda \frac{v_x^t(i + \Delta x/2, j) - v_x^t(i - \Delta x/2, j)}{\Delta x} \\ &\quad + (\lambda + 2\mu) \frac{v_z^t(i, j + \Delta z/2) - v_z^t(i, j - \Delta z/2)}{\Delta z} \end{aligned} \quad (\text{A.9})$$

$$\begin{aligned} \frac{\tau_{xz}^{t+\Delta t/2}(i, j) - \tau_{xz}^{t-\Delta t/2}(i, j)}{\Delta t} &= \mu \frac{v_z^t(i + \Delta x/2, j) - v_z^t(i - \Delta x/2, j)}{\Delta x} \\ &\quad + \mu \frac{v_x^t(i, j + \Delta z/2) - v_x^t(i, j - \Delta z/2)}{\Delta z} \end{aligned} \quad (\text{A.10})$$

and

$$\begin{aligned} \rho \frac{v_x^{t+\Delta t/2}(i, j) - v_x^{t-\Delta t/2}(i, j)}{\Delta t} &= \frac{\tau_{xx}^t(i + \Delta x/2, j) - \tau_{xx}^t(i - \Delta x/2, j)}{\Delta x} \\ &\quad + \frac{\tau_{xz}^t(i, j + \Delta z/2) - \tau_{xz}^t(i, j - \Delta z/2)}{\Delta z} \end{aligned} \quad (\text{A.11})$$

$$\begin{aligned} \rho \frac{v_z^{t+\Delta t/2}(i, j) - v_z^{t-\Delta t/2}(i, j)}{\Delta t} &= \frac{\tau_{xz}^t(i + \Delta x/2, j) - \tau_{xz}^t(i - \Delta x/2, j)}{\Delta x} \\ &\quad + \frac{\tau_{zz}^t(i, j + \Delta z/2) - \tau_{zz}^t(i, j - \Delta z/2)}{\Delta z}. \end{aligned} \quad (\text{A.12})$$

The solution of wave equation is obtained when I update these above equations are updated, alternately.

Next, by decomposing the spatial differentiation in Equation A.8 through A.12 and by using fourth-

order accuracy of space, it can be estimated;

$$\frac{\Delta v}{\Delta x} \approx \frac{1}{\Delta x} \left\{ -\frac{1}{24} v^t \left( i + \frac{3\Delta x}{2}, j \right) + \frac{27}{24} v^t \left( i + \frac{\Delta x}{2}, j \right) - \frac{27}{24} v^t \left( i - \frac{\Delta x}{2}, j \right) + \frac{1}{24} v^t \left( i - \frac{3\Delta x}{2}, j \right) \right\}.$$

By using this above relationship, it can be obtained the finite difference scheme of staggered grid method.

It has been calculated all simulation inside an area using this scheme, but not for the grids that are situated near the boundaries. Because this scheme is applied to calculate displacement of a point of the next time step by using the displacement values of neighboring 2 grid points from previous time step, this scheme is not implemented around boundary. I realize that this boundary does not exist in fact; this approach is applied only for computing reason. Off course it would be better if one can reduce the effect of boundary as possible as one can.

It has been adopted the free surface condition and absorbed boundary in the computation. For describing the effect of surface accurately, one should consider the free surface condition. It has been used the image method (Levander, 1988), which estimates the surface as boundary condition in a form of equation in space. Because free surface is free end as displacement and fixed end as pressure, the stress is 0 at  $z = 0$ .

$$\begin{aligned} \tau_{zz}(x, 0) &= 0 \\ \frac{\partial \tau_{zz}(x, 0)}{\partial t} &= 0 \\ \tau_{xz}(x, 0) &= 0. \end{aligned} \tag{A.13}$$

Thus, it can be rewritten the Equation A.8 and A.11 into

$$\frac{\tau_{xx}^{t+\Delta t/2}(i, j) - \tau_{xx}^{t-\Delta t/2}(i, j)}{\Delta t} = \left( \frac{4\lambda\mu + 4\lambda^2}{\lambda + 2\mu} \right) \frac{v_x^t(i + \Delta x/2, j) - v_x^t(i - \Delta x/2, j)}{\Delta x} \tag{A.14}$$

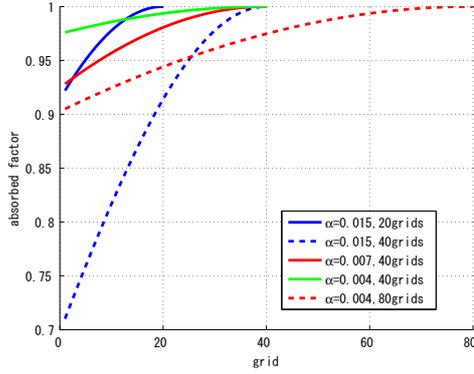
$$\rho \frac{v_x^{t+\Delta t/2}(i, j) - v_x^{t-\Delta t/2}(i, j)}{\Delta t} = \frac{\tau_{xx}^t(i + \Delta x/2, j) - \tau_{xx}^t(i - \Delta x/2, j)}{\Delta x} + \frac{2\tau_{xz}^t(i, j + \Delta z/2)}{\Delta z}. \tag{A.15}$$

In free surface boundary, it would be suitable if the wave is reflected. Because free surface boundary is assumed to be the surface of the earth, other boundaries are considered as computing limitation and the waves are not reflected at these boundaries. To fulfill this aim, it has been applied image method that can set a nonreflecting boundary condition (Cerjan et al., 1985), in which is formulation is written in A.16

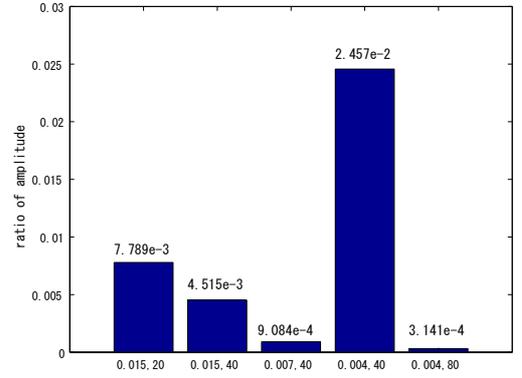
$$f(x) = e^{-(\alpha x)^2} \quad (\text{A.16})$$

Although (Cerjan et al., 1985) sets  $\alpha = 0.015$  and  $x = 20$  grids, I can adapt more than 20 grids for absorbed boundary because a large number of grids can be calculated easily by the advance of computational power. Afterward the ratio of absorption is confirmed when the variation of  $\alpha$  value and the number of absorbed grids are implemented. The absorbed factor ( $f(x)$ ) and the ratio of amplitude between direct waves and reflected wave that are observed near the absorbed boundary (Figure III.A.1). In this figure, the effect of boundary is reduced to about 10% compared to the ordinary parameter (e.g. the use of  $\alpha = 0.007$  and 40 grids parameters produces 10% of amplitude ratio than if  $\alpha = 0.015$  and 20 grids parameters are used, see Figure III.A.1(b)). If smaller  $\alpha$  is used, the absorption is not enough yet, so that big amplitude of reflected wave from the edge of boundary is appeared (see green line on Figure III.A.1(a)). For making absorbed boundary, the absorbed factor is multiplied to the wave that are exist in absorbed grids.

By using grid scheme simulation by computer, grid dispersion is often met when the digitizing data can not meet the exact behavior. In wave propagating simulation by FDTD, this phenomenon has been observed when I set too high velocity or too high frequency values. It has been shown the example of grid dispersion if too high frequency is applied, as can be seen in Figure III.A.2. Finally, the relationship between wave length and grid interval can be introduced based on experimental works



(a) absorbed factor



(b) ratio of amplitude

Figure III.A.1: (a) absorbed factor. Five cases that are different  $\alpha$  and the number of grids are shown. (b) the ratio of amplitude between direct and reflected from absorbed boundary wave. This ratio is calculated that reflected wave amplitude divided by direct wave amplitude. The label of  $x$  axis shows the  $\alpha$  and the number of grids. The numbers in the graph are exact values of each case. This amplitude is observed at the received point, which is 100 grids far from absorbed boundary and 50 grids far from source point. RMS amplitude for calculation and the calculation length is 40ms which corresponds to the length of source wavelet is used.

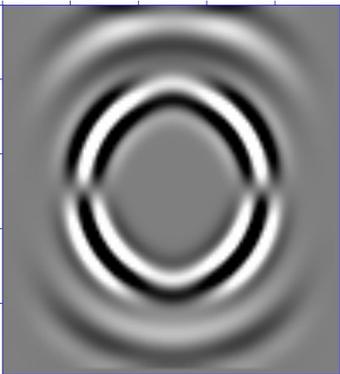
as follow;

$$\frac{\lambda}{dx} = \frac{V_{min}}{f \times dx} > 10 \sim 20, \quad (\text{A.17})$$

that is experimental relationship. The vague of  $10 \sim 20$  on the right side of equation A.6 is required as accuracy requirement. This is the stability limit by space domain. The stability limit in time domain proposed by (Graves, 1996),

$$0 < V_{max} \frac{dt}{dx} < 0.495. \quad (\text{A.18})$$

This relationship is analytical relationships and must be obeyed, otherwise, the answer of FDTD is always divergence.



(a) 30Hz



(b) 100Hz

Figure III.A.2: Grid dispersion by too high frequency. (b)100Hz of input source wavelet is too high to get stable result in this model.

# Conclusion

I showed the availability of virtual world in Geophysics.

From first part that is cross-coherence interferometry, I confirmed the features of cross-coherence from mathematics, numerical experience, and real data processing. I could retrieve stable valid wave and construct reflection section and S-wave velocity. This algorithm would be one of strong methods for shallow seismic survey in urban region where is difficult to investigate using active source.

In second part, numerical modeling is helpful for constructing seismic survey design of semi-permanently placed OBC. Although the result of this simulation is not fit exactly in any place, it is enough to make an indicator of the number of receivers. I need the verification of this indicator because this indicator is empirical not mathematical. I want to apply this indicator to real time-lapse data.

In third part, I succeeded to improve the computation time using GPU that is about 6 times faster than CPU. Although it is not enough speed, we may have much potential because I can improve 250 times speedup in another simulation. The efficiency of GPU is also improved day by day as same as our programming techniques. I surely be able to simulate whole survey data by three-dimensional elastic wave propagation in practical time keeping accuracy in near future that is a few years and I can come nearer to real world.

Virtual can make clear something before doing or make data that are difficult to do. By improving the computation power, virtual method certainly becomes more helpful and occupies bitter ratio in Geophysics.

# References

- Abdelkhalek, R., H. Calandra, O. Coulaud, G. Latu, and J. Roman, 2009, Fast seismic modeling and reverse time migration on a gpu cluster: HPCS, **2009**.
- Aoki, T., and A. Nukada, 2009, Cuda programming: Kougakusya.
- Arts, R., O. Eiken, A. Chadwick, P. Zweigel, L. van der Meer, and B. Zinszner, 2004, Monitoring of co2 injected at sleipner using time-lapse seismic data: Elsevier Energy, **29**, 1383–1392.
- BARKER, K. J., K. DAVIS, A. H. nd DARREN J. KERBYSONnd MIKELANG, S. PAKIN, and J. C. SANCHO, 2008, A performance evaluation of the nehalem quad-core processor for scientific computing: Parallel Processing Letters, **18**, 453–469.
- Behura, J., 2007, Virtual real source: SEG Expanded Abstracts, **26**, 2693–2697.
- Bo, L., L. Guo-feng, and L. Hong, 2009, A method of accelerating seismic pre-stack time migration by gpu: CPS/SEG Beijing, **2009**.
- Cerjan, C., D. Kosloff, R. Kosloff, and M. Reshef, 1985, A nonreflecting boundary condition for discrete acoustic and elastic wave equations: GEOPHYSICS, **50**, 705–708.
- Cerveny, V., 2001, Seismic ray theory: Cambridge university press.
- Draganov, D., X. Campman, J. Thorbecke, A. Verdel, and K. Wapenaar, 2009, Reflection images from ambient seismic noise: GEOPHYSICS, **74**, A63–A67.
- Fomel, S., 2009, Local seismic attributes: GEOPHYSICS, **72**, A29–A33.

- Graves, R. W., 1996, Simulating seismic wave propagation in 3d elastic media using staggered-grid finite differences: *Bulletin of the Seismological Society of America*, **86**, 1091–1106.
- Hale, D., 1984, Dip-moveout by fourier transform: *GEOPHYSICS*, **49**, 741–757.
- Ikelle, L. T., L. Amundsen, and O. Eiken, 1997, Multiple attenuation at primary/multiple interferences: The troll example: *The Leading Edge*, 1749–1751.
- Juhlin, C., 1995, Imaging of fracture zones in the finnsjon area, central sweden, using the seismic reflection method: *GEOPHYSICS*, **60**, 66–75.
- Kadlec, B. J., G. A. Dorn, and H. M. Tufo, 2009, Interactive visualization and interpretation of geologic surfaces in 3-d seismic data: *SEG Extended Abstract*, **2009**, 1147–1151.
- Koster, K., P. Gabriels, M. Hartung, J. Verbeek, G. Deinum, and R. Staples, 2000, Time-lapse seismic surveys in the north sea and their business impact: *The Leading Edge*, **2000**, 286–293.
- Kragh, E., and P. Christie, 2002, Seismic repeatability, normalized rms, and predictability: *The Leading Edge*, 640–647.
- Levander, A. R., 1988, Fourth-order finite-difference p-sv seismograms: *GEOPHYSICS*, **53**, 1425–1436.
- Liu, G., S. Fomel, L. Jin, and X. Chen, 2009, Stacking seismic data using local correlation: *GEOPHYSICS*, **74**, V43–V48.
- Lumley, D. E., J. F. Claerbout, and D. Bevc, 2001, Anti-aliased kirchhoff 3-d migration: *Stanford Exploration Project*, **80**, 1–444.
- Mehta, K., A. Bakulin, D. Kiyashchenko, and J. Lopez, 2008a, Comparing virtual versus real crosswell surveys: *SEG Expanded Abstracts*, **28**, 1372–1376.
- Mehta, K., J. L. Sheiman, R. Snieder, and R. Calvert, 2008b, Strengthening the virtual-source method for time-lapse monitoring: *GEOPHYSICS*, **73**, S73–S80.
- Micikevicius, P., 2009, 3d finite difference computation on gpus using cuda: *GPGPU2*.

- 
- Moussa, N. W., 2009, Seismic imaging using gpgpu accelerated reverse time migration cs315a final project report: Parallel Computer Architecture and Programming, **2009**.
- Neidell, N. S., and M. T. Taner, 1971, Semblance and other coherency measures for multichannel data: *GEOPHYSICS*, **36**, 482–497.
- Nguyen, H., 2007, Gpu gems 3: Addison-Wesley Professional, CUDA.
- NVIDIA, 2009, Nvidia cuda programming guide 2.2.1: NVIDIA.
- Owens, J. D., D. Luebke, N. Govindaraju, M. Harris, J. Kruger, A. E. Lefohn, and T. J. Purcell, 2005, A survey of general-purpose computation on graphics hardware: *EUROGRAPHICS*, **2005**.
- Roux, P., and M. Fink, 2003, Green's function estimation using secondary sources in a shallow wave environment: *J. Acoust. Soc. Am.*, **113**, 1406–1416.
- Saito, M., 1993, Branch line contribution in lamb's problem: *BUTSURI-TANSA*, **46**, 372–380.
- Schuster, G., 2009, Seismic interferometry: CAMBRIDGE UNIVERSITY PRESS.
- Snieder, R., M. Miyazawa, E. Slob, I. Vasconcelos, and K. Wapenaar, 2009, A comparison of strategies for seismic interferometry: *Surv Geophys*, **DOI**.
- Snieder, R., K. Wapenaar, and K. Larner, 2006, Spurious multiples in seismic interferometry of primaries: *GEOPHYSICS*, **71**, SI111–SI124.
- Takada, N., T. Takizawa, Z. Gong, N. Masuda, T. Ito, and T. Shimobaba, 2008, Fast computation of 2-d finite-difference time-domain method using graphics processing unit with unified shader: *IEICE*, **J91-D**, 2562–2564.
- Vasconcelos, I., and R. Snieder, 2008a, Interferometry by deconvolution, part 1 - theory for acoustic waves and numerical examples: *GEOPHYSICS*, **73**, S115–S128.
- , 2008b, Interferometry by deconvolution: Part 2 - theory for elastic waves and application to drill-bit seismic imaging: *GEOPHYSICS*, **73**, S129–S141.

- 
- Vinje, V., K. Astebol, E. Iversen, and H. Gjoystdal, 1999, 3-d ray modeling by wavefront construction in open models: *GEOPHYSICS*, **64**, 1912–1919.
- Vinje, V., E. Iversen, K. Astebol, and H. Gjoystdal, 1996a, Estimation of multivalued arrivals in 3d models using wavefront construction - part i: *Geophysical Prospecting*, **1996**, 819–842.
- , 1996b, Part ii: Tracing and interpolation: *Geophysical Prospecting*, **1996**, 843–858.
- Vinje, V., E. Iversen, and H. Gjoystdal, 1993, Traveltime and amplitude estimation using wavefront construction: *GEOPHYSICS*, **58**, 1157–1166.
- Virieux, J., 1986, P-sv wave propagation in heterogeneous media: Velocity-stress finite-difference method: *GEOPHYSICS*, **51**, 889–901.
- Wapenaar, K., 2003, Synthesis of an inhomogeneous medium from its acoustic transmission response: *GEOPHYSICS*, **68**, 1756–1759.
- , 2004, Retrieving the elastodynamic green's function of an arbitrary inhomogeneous medium by cross correlation: *PHYSICAL REVIEW LETTERS*, **93**, 254301.
- Wapenaar, K., D. Draganov, and J. Robertsson, 2006, Introduction to the supplement on seismic interferometry: *GEOPHYSICS*, **71**, SI1–SI4.
- Wapenaar, K., and J. Fokkema, 2006, Green's function representations for seismic interferometry: *GEOPHYSICS*, **71**, SI33–SI46.
- Wapenaar, K., E. Slob, and R. Snieder, 2008a, Seismic and electromagnetic controlled-source interferometry in dissipative media: *Geophysical Prospecting*, **56**, 419–434.
- Wapenaar, K., J. van der Neut, and E. Ruigrok, 2008b, Passive seismic interferometry by multidimensional deconvolution: *GEOPHYSICS*, **73**, A51–A56.
- Wapenaar, K., J. van der Neut, E. Ruigrok, D. Draganov, EvertSlob, J. Thorbecke, and R. Snieder, 2008c, Seismic interferometry by cross-correlation and deconvolution?: *SEG Expanded Abstract*.
- Yamanaka, H., and H. Ishida, 1995, Phase velocity inversion using genetic algorithms: *J. Struct. Constr. Eng.*, **468**, 9–17.