Important Notice

UNIVERSITY OF CALGARY

Fluid flow modelling and its seismic differencing in time-lapse

by

Vanja Vracar (nee Milicevic)

A 2012

SUBMITTED TO THE FACULTY OF GRADUATE STUDIES

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE

DEGREE OF MASTERS IN SCIENCE

DEPARTMENT OF GEOSCIENCE

CALGARY, ALBERTA

January, 2012

UNIVERSITY OF CALGARY

FACULTY OF GRADUATE STUDIES

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies for acceptance, a 2012 thesis entitled "Fluid flow modelling and its seismic differencing in time-lapse" submitted by Vanja Vracar (nee Milicevic) in partial fulfillment of the requirements for the degree of MASTERS IN SCIENCE.

Supervisor, Dr. Robert
J. Ferguson
Department of Geoscience

Dr. Gary F. Margrave
Department of Geoscience

Dr. Laurence R. Lines
Department of Geoscience

Dr. Michael
P. Lamoureux
Department of
Mathematics and
Statistics

Date

Abstract

In this thesis I explore alternatives to conventional seismic differencing and their application to seismic time-lapse analysis.

The motivation is my observation that conventional seismic differencing relies on a number of assumptions, and that these may not always represent reality. Systematic error, error associated with the use of conventional (imperfect) imaging algorithms, and error due to source/receiver coupling variations are assumed to be small relative to the seismic response of fluid transport in a reservoir for which source/receiver positioning and coupling must be the same between surveys in time-lapse. The result is that conventional differencing involves simple match filtering followed by subtraction where the interpretable product is an image of the change in fluid location superimposed upon some background noise level. In reality, however, errors are often very large.

I observe that though errors might be large, and with the exception of source/receiver location repeatability, coupling variation and systematic errors result in differences in seismic amplitude and not necessarily seismic phase so that any methodology beyond simple match-filtering and differencing might incorporate this observation.

I base my work on numerical experiments where I compare conventional differencing for seismic time-lapse analysis to a number of new algorithms that I have developed. The numerical experiments incorporate a fluid-flow simulator, a Gassmann-equation based conversion from time-lapse saturation and pressure values to density and elasticity, and then 3D, multi-component seismic simulation.

For comparison to my new approaches to seismic differencing, I perform conventional differencing through matrix subtraction. For simplicity, I assume that source and receiver error is negligible (or have been corrected for), and I compute differences by subtraction alone.

Five non-conventional seismic differencing algorithms are implemented: 1) inverse data space differencing (IDSD), 2) cross-correlation differencing (CCD), 3) pseudo cross-correlation differencing (PCCD), 4) conventional imaging condition differencing (CICD) and 5) imaging condition differencing (ICD). The IDSD is a filtering algorithm that employs inverse data matrix theory on migrated seismic models. It clears amplitude patterns by focusing differences and dimming similarities of two time-lapse steps. The CCD and PCCD algorithms are performed in the time and frequency domains, respectively. These algorithms consist of a cross-correlation operation, gaussian filtering and inversion. The algorithms' results are then multiplied by the conventional seismic differencing models and passed to a pre-stack depth migration (PSDM). A disadvantage of both algorithms is dependence on the user to manually move data from differencing to migration. The CICD is a combination of PSDM and conventional differencing. It is a pilot algorithm to combine the PSDM with the PCCD. It proves to be efficient and robust when compared to conventional differencing. The ICD method combines PCCD, differencing and migration in one algorithm, hence minimizes the user's dependence and improves computational time and imaging. The CCD, PCCD and ICD capture almost only fluid flow changes and eliminate mostly all similarities on the final differenced models.

The computational cost of non-conventional differencing methods varies. Assume the data is stored in an $M \times N$ matrix. The most expensive one is the CCD that takes $O(M^5N^5+3MN+MNlog(MN))$ operations to complete, where the ICD is the cheapest taking $O(M^2N^2 + 2MN + 6MNlogMN)$ operations to complete.

When compared to conventional seismic differencing, all non-conventional seismic differencing algorithms, except CICD, capture significant imaging improvements, hence aid in geophysical interpretation, reservoir monitoring, characterization and time-lapse studies.

Acknowledgements

I would like to express my gratitude to Dr. Robert Ferguson for the support, guidance, insightful and challenging supervision of the work. I am thankful to Dr. Ferguson for enabling my smooth transition from mathematics to highly dynamic and enriching science of geophysics.

Thank you to Doctors Gary Margrave, Don Lawton, Larry Lines, Kristopher Innanen and Stephen Hubbard for impressive lectures and thought provoking discussions.

Thank you to Dr. Michael Lamoureux for encouraging words of my development and work as a mathematician shaping in a geophysicist.

Thank you to CREWES (Consortium for Research in Elastic Wave Exploration Seismology) for exposing me to domestic and international conferences, industry and its professionals and numerous research resources. I thank all the staff of CREWES, especially Kevin Hall and Dr. Rolf Maier for technical support and Laura Baird for keeping me in good books with administration.

Thank you to my fellow students Virginia Vera, Diane Lespinasse, Heather Lloyd, Akshey Gulati, Faranak Mahmoudian and Joanna Cooper for sharing their knowledge and experience.

Last but not least, I would like to thank my family. Thank to my husband, Bojan Vracar, for being my inspiration and motivation without whom this work would not be possible. I am proud to call you my husband.

Thank you to my dad, Dr. Vlado Milicevic, for being my role model, who thought me how to learn, appreciate and love science. Thank you to my mom, Vojislava Milicevic, for her support and for teaching me to be passionate and responsible in my work. Thank you to my baby sister, Marija Ivanovic (nee Milicevic), for sharing her knowledge and university days with me. Thank you to my uncle, Nikola Nikic, for introducing me

to CREWES Project, industry and building my networking skills. Thank you to my grandpa, Mihail Nikic, for planting my interest in geophysics.

Dedication

To my husband, Bojan Vracar, with love.

Acronyms

IDS	Inverse data space
IDSD	Inverse data space differencing
SSF	Split-step Fourier migration
PSDM	Pre-stack depth migration
CD	Conventional differencing
CCD	Cross-correlation differencing
PCCD	Pseudo cross-correlation differencing
IPCC	Inverse pseudo cross-correlation
CICD	Conventional imaging condition differencing
ICD	Imaging condition differencing

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Motivation

Development of a reservoir depends on the alliance of geologists, geophysicists and engineers. These scientists work closely towards a common goal: reservoir localization, production and characterization under economical means (Hubbard, personal communication). To highlight prospective areas, geologists study the area, define source rocks, reservoir rocks and construct plays (Holstein, 2007). Geophysicists acquire and interpret seismic data to obtain subsurface images (Kearey et al., 2002). These images help identify formations, traps, folds and possible hydrocarbon reservoir existence (Shearer, 1999). After completion of wells, engineers collect data that aid in production planning and future developments. Each analysis is a significant measure in reservoir characterization (Holstein, 2007).

When analyses are studied and infrastructure is set, production begins. Eventually, primary production recovery becomes uneconomical due to reservoir depletion (Cosse, 1993). At this time artificial recovery methods are employed: injections of water, gas, chemicals or steam in heavy oil reservoirs (Cosse, 1993). Success in enhanced recovery requires reservoir familiarity (Rickett and Lumley, 2010). This is not difficult for reservoirs with long production history, however, it is a challenging task in reservoirs with short to no production history (Holstein, 2007). Then, numerical and seismic modelling of injection flow, usual secondary recovery modelling mechanisms, allow visualization and analysis of reservoir properties (Aarnes et al., 2007). At this time, recorded or modelled time-lapse studies become crucial tools as they allow continuous monitoring of property

changes during production (Rickett and Lumley, 2010).

Presently, conventional seismic differencing is a method employed by researchers and industry professionals in analyzing changes in such reservoirs recorded in time-lapse. Motivation of this work is my observation that mentioned conventional seismic differencing carries various assumptions that can be distorted. It is assumed source/receiver coupling variations error to be negligible relative to the seismic response of fluid transport in a reservoir whose source/receiver positioning must be the same between time-lapse surveys. By conventional differencing, filtering followed by subtraction, we obtain a seismic image to interpret. Essentially, this image represents a change in fluid location superimposed upon certain background noise level. In practice noise can be large and can mask responses we are looking for. This method can produce differences in seismic amplitude, but there is often a lesser effect on seismic phase.

1.2 Thesis objective

The objective of this thesis is to implement a seismic differencing algorithm that will highlight major fluid flow changes in a producing time-lapse reservoir by removing ambiguities caused by conventional differencing through implementation and analyses of methods prone to seismic amplitude errors.

1.3 Data used

There are two data sets used to verify theory in this thesis work. The first data set used to verify theory is the 10^{th} SPE Comparative solution project, a free data set that is publicly available (Christie and Blunt, 2001). The data set models a sandstone reservoir with one oil producing and two water injecting wells, where the reservoir has a 3D vertical cross-sectional geometry with no dips or faults (Christie and Blunt, 2001). Initially, the

reservoir is 100% oil saturated (Aarnes et al., 2007) and the reservoir boundaries are impermeable (Aarnes et al., 2007). The viscosity, porosity, and permeability are uniform and the oil and water phases are immiscible and incompressible in that there are no blending or density changes (Cosse, 1993). Water and oil saturations are irreducible, all reservoir oil is displaceable by water during production (Aarnes et al., 2007). The producer is located at the centre of the reservoir and the two injectors are situated on the left and right hand side of the producer at equal distances.

The second data set used is the EAGE/SEG salt velocity model (Aminzadeh et al., 1996). This model consists of complex salt structures with large velocity contrasts across the salt/sediment interface (Aminzadeh et al., 1996). To accommodate time-lapse study and analysis the author inserts a small reflector in the sub-salt region of the EAGE/SEG salt velocity model and assumes it to mimic changes due to production.

1.4 Software used

The work demonstrated in Chapters 2, 3 and 4 was developed on the computer network managed by the CREWES Project, a research group of the Department of Geosciences at the University of Calgary. The network was accessed using the X11 application of a MacBook computer through the Mac OS X operating system.

The main software used in thesis work generation and analysis is MATLAB (Matrix Laboratory) a numerical computing environment and fourth-generation programming language developed by MathWorks. Numerical fluid flow simulation employed in Chapter 2 is a series of MATLAB functions developed by SINTEF Petroleum Research of Trondheim in Norway. Rock physics theory, also shown in Chapter 2, is studied through a numerical experiment using a self-coded MATLAB function named *gassmann*. 2D acoustic synthetic seismic data was generated using the $afd\_explode$ and $afd\_shotrec$

MATLAB CREWES toolbox functions developed by Dr. Gary Margrave of the University of Calgary. 3C-3D elastic synthetic seismic data was generated employing $Tiger$, a commercial software designed by SINTEF Petroleum Research of Trondheim in Norway. The migration of synthetics is generated invoking MATLAB CREWES toolbox functions, $ss\_salt\_psdm\_diff\_script$ and ss_zero_migt performing split-step Fourier migration developed by Dr. Robert Ferguson of the University of Calgary. The non-conventional seismic differencing algorithms cross-correlation differencing (CCD) and pseudo cross-correlation differencing (PCCD) are self-coded MATLAB functions named cc_diff and pcc_diff, respectively. Conventional imaging condition differencing (CICD) and imaging condition differencing (ICD), named cic_diff and ic_diff, are also developed in MATLAB by Dr. Robert Ferguson of the University of Calgary and the author.

Word processing and thesis assembly was done on the MacBook with LaTeX.

1.5 Thesis chapter summary

In Chapter 2 I perform a time-lapse numerical fluid flow modelling to convert pressure and saturation models to density and velocity models using Gassmann's relations in MATLAB. The velocity and density models are further used to generate synthetic seismic models through the acoustic and elastic finite-difference modelling employing MATLAB and Tiger. The obtained time-lapse seismic models are further employed in conventional and non-conventional seismic difference modelling and analyses.

Conventional differencing (CD), that is matrix subtraction is performed and analyzed in Chapter 3. As it proves to be unreliable due to its sensitivity to seismic amplitude, the first non-conventional seismic differencing algorithm is developed and presented in the second half of Chapter 3. The new differencing algorithm is based on the inverse data space theory and matrix inversion. The algorithm is named inverse data space differenc-

ing (IDSD). As seismic differencing proves to be more sensitive to the seismic amplitude and less so to seismic phase, I develop four more non-conventional seismic differencing algorithms and present them in Chapter 4. They are: cross-correlation differencing (CCD), pseudo cross-correlation differencing (PCCD), conventional imaging condition differencing (CICD) and imaging condition differencing (ICD). These algorithms cross-correlate time-lapse steps and filter them by Gaussian function to eliminate zero lag. Then, the cross-correlation operation is inverted and data is migrated. At this point the migrated data only highlights differences in fluid flow and aids reservoir characterization in time-lapse. In Chapter 5 I state conclusions of findings and propose future work.

Chapter 2

Numerical Fluid Flow Modelling and Its Seismic Response in Time-lapse

2.1 Introduction

Numerical simulators offer time-lapse model generation that can be studied for various potential enhanced recovery schemes. Stoffa et al. (2008) show that fluid injection causes seismic response changes where Gassmann's relations tie fluid flow to density saturation, compressional (P) wave and shear (S) wave velocities (Mavko et al., 2009). Charkraborty (2007) shows fluid flow changes using Gassmann's relations trigger changes in time-lapse seismic responses. Bentley and Zou (2003) also show, using Gassmann's equations plus sonic and density well logs, that fluid substitution gives rise to a seismic response.

In this chapter, I map fluid flow to density and elasticity through various numerical experiments. Since P-wave velocity is a valuable tool in studying and describing rocks lithologic properties (Ferguson, 1995), the study will focus on P-wave models. The P-wave velocity models will be used in generation of acoustic and elastic synthetic models. Both velocity and seismic models will be the basis for the time-lapse seismic differencing algorithms in Chapters 3 and 4.

2.2 Workflow: reservoir properties to seismic responses

The designed time-lapse workflow consists of: 1) reservoir simulation, 2) rock physics, and 3) seismic modelling.

Reservoir Simulation

Input: Geometry Porosity Permeability Fluid properties	→	MATLAB: Numerical simulator	→	Output: Pressure Saturation

Rock Physics

Input: Saturation Pressure Porosity Dry rock properties	→	MATLAB: Gassmann's equations	→	Output: P-wave velocity S-wave velocity Density saturation

Seismic Modelling

Input: P-wave velocity Density saturation	→	MATLAB: Finite-difference algorithm	→	Output: Amplitude Phase

Table 2.1: Workflow I: Schematic map of study steps, showing input/output parameters and software used. Pressure and saturation of the reservoir are obtained from its geometry, porosity, permeability and fluid properties. Pressure and saturation models are further employed to calculate density saturation, P-wave and S-wave velocities through Gassmann's equations. These seismic models are then generated from density saturation, P-wave and S-wave velocities. Seismic models will be later used for generating seismic amplitude and phase spectra.

In table 2.1 I outline a workflow used to generate seismic models. Reservoir properties

are taken through a numerical simulator to produce pressure and saturation models. The numerical simulator is a set of MATLAB routines from SINTEF Petroleum Research of Trondheim in Norway. Detailed reservoir properties (geometry, porosity, permeability and fluid properties) will be stated in the upcoming sections. Further rock physics is used to calculate density and P-wave velocity from saturation models. This modelling is done employing a self-coded MATLAB function named *Gassmann*. The outlined P-wave velocity models are passed to finite difference software to generate synthetic seismic models. The software packages used are: a MATLAB function $afd\_explode$, found in the CREWES toolbox, and the commercial software $Tiger$, designed by SINTEF Petroleum Research of Trondheim. Seismic models are generated employing the finite-difference method for both acoustic and elastic media.

2.3 Theory

Assume a homogeneous and isotropic reservoir for simulation. The reservoir simulation system consists of two-phase flow with a hydrocarbon phase and water phase. Assume a 100 % oil saturated sandstone reservoir with one producing and two water injecting wells scenario.

2.3.1 Numerical fluid flow

Firstly, assume constant porosity and incompressibility (no density variation in time). Also, assume no-flow boundary conditions. In order to model phase flow through porous medium, I start with the continuity equation (Aarnes et al., 2007):

$$\frac{\partial(\phi_p \rho_p)}{\partial t} + \nabla \cdot \rho_p v_{f,p} = q_p, \tag{2.1}$$

where p, ϕ_p, ρ_p, t, $v_{f,p}$ and q_p are desired phase (water or oil), phase porosity, phase density, time, phase flow velocity and inflow/outflow per volume, respectively. Now,

consider Darcy's law that relates flow velocity, $v_{f,p}$ to pressure, p_p:

$$v_{f,p} = -\frac{k_p}{\mu_p}[\nabla p_p + \rho_p G], \tag{2.2}$$

where, k_p, μ_p, ρ_p, and G are the phase permeability, viscosity, density, and gravitational constant, respectively (Aarnes et al., 2007). Now, replacing $v_{f,p}$ in equation (2.2) with equation (2.1) I get an elliptic equation for phase pressure conserved in time-lapse (Aarnes et al., 2007):

$$\nabla \cdot v_{f,p} = \frac{q_p}{\rho_p}. \tag{2.3}$$

Equation (2.3) describes pressure gradient constant in each grid box over time and its variance from grid box to grid box. The temperature changes are neglected. Saturation can be derived from the continuity equation of each phase and pore saturation (s) as:

$$\phi\frac{\partial s}{\partial t} + \nabla f(s) v_{f,p} = \frac{q_p}{\rho_p}, \tag{2.4}$$

Hence assuming properties of incompressibility and time conservation I have a relation: $s_w + s_o = 1$. Equation (2.4) estimates saturation from reservoir conditions and water flow in each grid box. The numerical modelling of fundamental reservoir system is done employing equations (2.3) and (2.4).

2.3.2 Rock physics

Gassmann's equations are employed to create velocity models from saturation models. Recall medium to be homogeneous and isotropic. Mavko et al. (2009) states:

$$K_{sat} = K_d + \frac{(1 - \frac{K_d}{K_0})^2}{\frac{\phi}{K_f} + \frac{1-\phi}{K_0} - \frac{K_d}{K_0^2}} \quad \text{and} \quad \mu_{sat} = \mu_d, \tag{2.5}$$

where ϕ is porosity, and K_{sat}, K_f , K_d, and K_0 are the effective bulk modulus of saturated rock, the effective bulk modulus of pore fluid, the frame bulk modulus of dry rock and the bulk modulus of mineral material making up the rock, respectively. The saturated

shear modulus and the dry shear modulus, μ_{sat}, and μ_d, respectively, are independent of saturation (Mavko et al., 2009). Assume constant porosity in the sandstone reservoir. Now, invoke fluid density relation:

$$\rho_f = s_w\rho_w + s_o\rho_o, \tag{2.6}$$

where ρ_f, s_w, ρ_w, s_o, ρ_o are fluid density, water saturation, water density, oil saturation, oil density, respectively. Using results of equation (2.6) I obtain density saturation (density of saturated rock), ρ_{sat} from:

$$\rho_{sat} = (1-\phi)\rho_0 + \phi\rho_f, \tag{2.7}$$

where ρ_0 is matrix density. Combination of equation (2.5) and equation (2.7) yields P-wave velocity, α (Mavko et al., 2009):

$$\alpha = \sqrt{\frac{K_{sat} + \frac{4}{3}\mu_{sat}}{\rho_{sat}}}, \tag{2.8}$$

and S-wave velocity, β (Mavko et al., 2009):

$$\beta = \sqrt{\frac{\mu_{sat}}{\rho_{sat}}}. \tag{2.9}$$

P-wave and S-wave velocity models of the saturated rock are generated using equations (2.8) and (2.9), respectively.

2.3.3 Finite difference modelling in acoustic medium

Agreste and Ricciardello (2011) state seismic wave propagation can be described using the scalar wave equation. Youzwishen and Margrave (1999) define the variable velocity two-dimensional acoustic wave equation as:

$$\bigtriangledown^2\,\psi(x,z,t) = \frac{\partial^2\psi(x,z,t)}{\partial x^2} + \frac{\partial^2\psi(x,z,t)}{\partial z^2} = \frac{1}{v^2(x,z)}\frac{\partial^2\psi(x,z,t)}{\partial t^2}, \tag{2.10}$$

where $\bigtriangledown^2$, ψ, x, z, t, v denote Laplacian, wavefield, distance, depth, time and velocity, respectively. A second-order approximation for the time derivative of (2.10) is given by ((Youzwishen and Margrave, 1999) and (Agreste and Ricciardello, 2011)):

$$\bigtriangledown^2 \psi(x,z,t) = \frac{1}{\Delta t^2 v^2(x,z)}[\psi(x,z,t+\Delta t) - 2\psi(x,z,t) + \psi(x,z,t-\Delta t)], \qquad (2.11)$$

where Δt is the time discretization interval. Solving equation (2.11) for $t + \Delta t$ gives (Youzwishen and Margrave, 1999):

$$\psi(x,z,t+\Delta t) = [2 + 2\Delta t^2 v^2(x,z)\bigtriangledown^2]\psi(x,z,t) - \psi(x,z,t-\Delta t). \qquad (2.12)$$

Equation (2.12) defines a recursive process where t increases at every time step, that is the wavefield $t+\Delta t$ can be calculated if t and $t-\Delta t$ are known (Youzwishen and Margrave, 1999).

Recursive algorithm
Step I:
$\psi(x,z,t-\Delta t)$ and $\psi(x,z,t)$ $\rightarrow$ $\psi(x,z,t+\Delta t)$
Step II:
$\psi(x,z,t-\Delta t) \leftrightarrow \psi(x,z,t)$
Step III:
$\psi(x,z,t) \leftrightarrow \psi(x,z,t+\Delta t)$
Step IV:
$t \leftrightarrow t+\Delta t$

Table 2.2: Workflow II: Recursive finite difference algorithm. In step I $\psi(x,z,t+\Delta t)$ calculated from $\psi(x,z,t)$ and $\psi(x,z,t-\Delta t)$. In step II $\psi(x,z,t-\Delta t)$ is replaced by $\psi(x,z,t)$. In step III $\psi(x,z,t)$ is replaced by $\psi(x,z,t+\Delta t)$. In step IV the value of t is increased. Once recursive loop is complete, finite difference data is delivered.

Table 2.2 is a summary of recursive finite difference algorithm for acoustic medium (Youzwishen and Margrave, 1999). The algorithm is then extended to employ exploding reflector concept. The algorithm calculates $\psi(x,z,t=0)$ from velocity model assuming normal incidence reflectivity at constant density (Youzwishen and Margrave, 1999). Further, the initial wavefield is time stepped employing equation (2.12) to produce synthetics

seismic models.

2.3.4 Finite difference modelling in elastic medium

Yao and Margrave (1999) state employing elastic wave theory 3D wave propagation is described as:

$$\begin{aligned}
\partial_t v_x &= \frac{1}{\rho}(\partial_x \tau_{xx} + \partial_y \tau_{xy} + \partial_z \tau_{xz} + f_x), \\
\partial_t v_y &= \frac{1}{\rho}(\partial_x \tau_{xy} + \partial_y \tau_{yy} + \partial_z \tau_{yz} + f_y), \\
\partial_t v_z &= \frac{1}{\rho}(\partial_x \tau_{xz} + \partial_y \tau_{yz} + \partial_z \tau_{zz} + f_z),
\end{aligned} \tag{2.13}$$

where f_i, τ, v_i and ρ denote body force components, stress components, velocity components, density, respectively. The stress-strain equations of equations (2.13) are (Hokstad et al., 2007):

$$\begin{aligned}
\partial_t \tau_{xx} &= (\lambda + 2\mu)\partial_x v_x + \lambda(\partial_y v_y + \partial_z v_z), \\
\partial_t \tau_{yy} &= (\lambda + 2\mu)\partial_y v_y + \lambda(\partial_x v_x + \partial_z v_z), \\
\partial_t \tau_{zz} &= (\lambda + 2\mu)\partial_z v_z + \lambda(\partial_x v_x + \partial_y v_y), \\
\partial_t \tau_{xy} &= \mu(\partial_y v_x + \partial_x v_y), \\
\partial_t \tau_{xz} &= \mu(\partial_z v_x + \partial_x v_z), \\
\partial_t \tau_{yz} &= \mu(\partial_z v_y + \partial_y v_z),
\end{aligned} \tag{2.14}$$

where λ and μ denote Lamé constants.

Once the workflow is executed various models are delivered. The reservoir top and bottom reflections are expected to be stationary on all plots in time-lapse. The waterfronts are anticipated to map sooner as time progresses. Density decrease is expected with water inflow in time. No variation, when laterally correlating density above and below waterfronts in time-lapse, is anticipated. Density above waterfronts alone maps no change in time-lapse.

2.4 Examples

The above developed workflow is applied to the 10^{th} SPE Comparative Solution Project, a free data set that is publicly available (Christie and Blunt, 2001) for verification. The data set is also convenient for its capability to run on a single processor. The study comprises of a sandstone reservoir with two injecting and one producing well. The reservoir has a 3D vertical cross-sectional geometry with no dips or faults (Christie and Blunt, 2001). Its detailed properties are listed in the Table 2.3. Initially, the reservoir is 100% oil saturated. The reservoir boundaries are impermeable, or no-flow. The viscosity, porosity, and permeability are uniform.

Property	**Units**
Reservoir	64 x 64 x 1 grid boxes, each grid box: 7.62 m x 7.62 m x 7.62 m
Oil Density	700 kg/m^3
Water Density	1000 kg/m^3
Sandstone Density	2600 kg/m^3
Depth	3900 m
Distance Coverage	3900 m
Initial Pressure (injector)	655 002 Pa
Initial Pressure (producer)	689 476 Pa
Porosity	20 %
Viscosity	1 cp

Table 2.3: Reservoir properties used in reservoir simulation (Christie and Blunt, 2001).

2.4.1 Numerical fluid flow simulation

A public domain numerical simulator, provided by SINTEF Petroleum Research of Trondheim in Norway, consists of several MATLAB routines, whose main one is $runq5$ (Aarnes et al., 2007). It models reservoir fluid flow.

The study models two-phase flow, that is oil production simulation through water injection in 28 days. The study's duration is short due to the exaggerated reservoir properties and low mobility ratio, that is low oil and water viscosity ratio. The phases are immiscible and incompressible, namely there are no blending or density changes (Cosse, 1993). Water and oil saturations are irreducible; that is, oil is fully displaceable by water (Aarnes et al., 2007). The boundary effects are ignored, since the purpose of the work is to verify workflow and obtain seismic models for further study. The producer is located at the centre of the reservoir. The two injectors are situated on the left and right hand side of the producer at symmetrically equal distances.

For simplicity of illustration, I only present the right hand side of the reservoir, that is the producer with one injector. I assume the left hand side to be a mirror image of the right hand side.

Pressure behavior and saturation models are calculated using inputted reservoir properties.

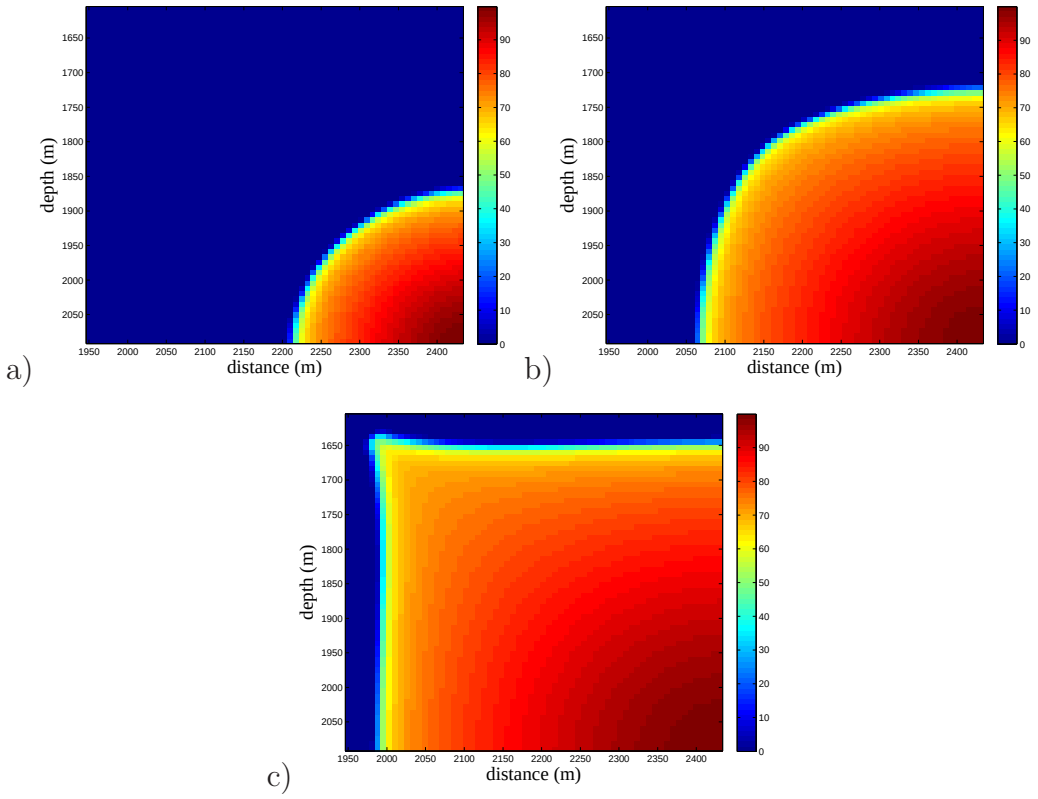


Figure 2.1: Saturation models. Reservoir is initially 100% oil saturated. The producer and one injector are located in the upper-left and lower-right corner of the model, respectively. The injector pumps water into the reservoir. The models 2.1(a), 2.1(b) and 2.1(c) show reservoir as water saturation increases towards the producer in time-lapse steps after day $\tau = 1, 14, 28$, respectively. The color bar denotes water saturation in *percentage*.

The oil producer and water injector are situated in the upper-left and lower-right corner of the grid in Figure 2.1, respectively. Figure 2.1(a) and 2.1(b) capture water saturation increase and in situ oil displacement after day 1 and day 14, respectively. Figure 2.1(c) illustrates leading waterfront after 28 days as it develops finger like flow up to the breakthrough in the production. Note the water injection is constant throughout

28 days.

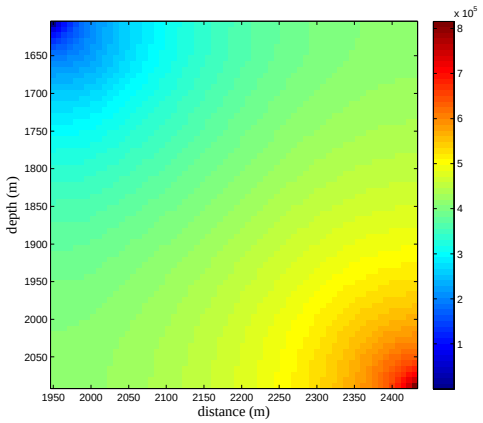


Figure 2.2: Pressure model. The initial pore pressure model. The producer and one injector are located in the upper-left and lower-right corner of the model, respectively. Pressure decreases from injector to producer. Assume pressure is constant through a 28 days simulation. The color bar denotes pressure values in *Pascals*.

Figure 2.2 illustrates initial pressure of the reservoir as it decreases from injector to producer. Assume initial pressure to be constant in each grid box through a 28 day simulation (Aarnes et al., 2007).

2.4.2 Rock physics

Using saturation models and Gassmann's relations, MATLAB code is designed to calculate density saturation. Assume constant porosity in the reservoir. This assumption yields constant K_d and K_0 precisely listed in Table 2.4.

Property	Units
Sandstone Shear Modulus	5.04 GPa
Sandstone Bulk Modulus	0.70 GPa
Water Bulk Modulus	2.20 GPa
Oil Bulk Modulus	2600 kg/m^3

Table 2.4: Constants used in Gassmann's equation to obtain the effective bulk modulus of saturated rocks, K_{sat} (Mavko et al., 2009) and (Beer and Maina, 2008).

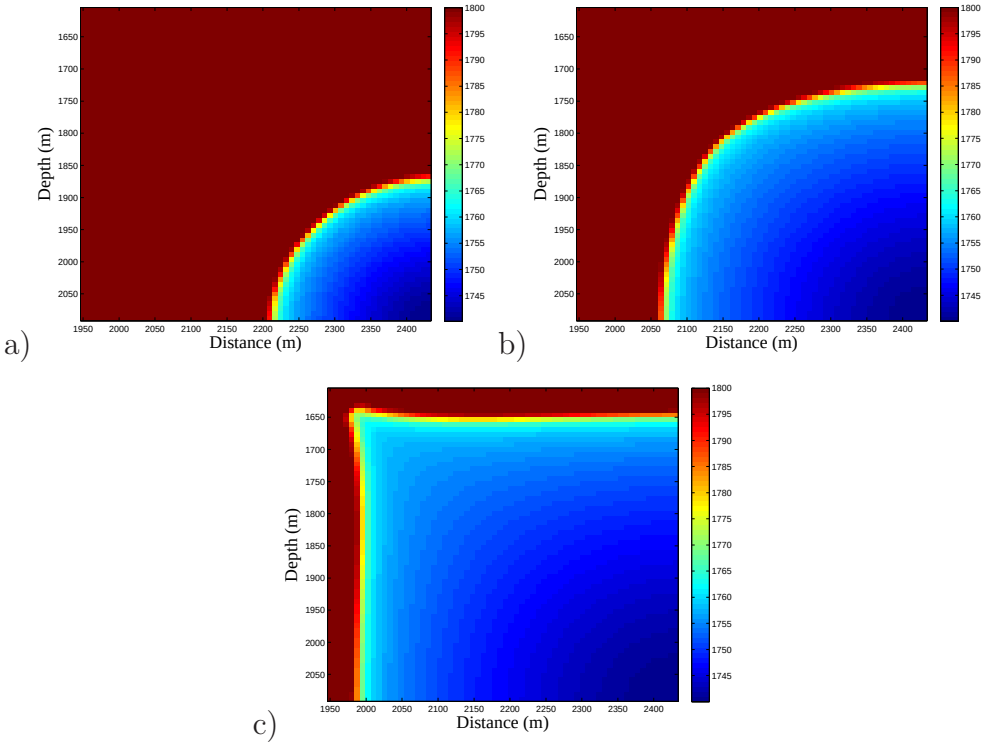


Figure 2.3: Density saturation models. The reservoir is initially 100 % oil saturated. The producer and one injector are located in the upper-left and lower-right corner of the model, respectively. The injector pumps water into the reservoir. The models 2.3(a), 2.3(b) and 2.3(c) show reservoir as density saturation increases towards the producer in time-lapse steps after day $\tau = 1, 14, 28$, respectively. The color bar denotes density saturation values in kg/m^2.

Figure 2.3(a), 2.3(b), and 2.3(c) show density saturation after day 1, 14, and 28, respectively. Note an increase of density saturation from injector to producer due to water saturation changes and density of fluid. These changes are directly related to the density of saturated rock. Water saturation increases with injection, rocks once saturated with oil are now saturated by water. Because of higher water density, the overall density of the rock consequently becomes higher.

Further, density saturation allows velocity model building. Since I assume irreducibly, the bulk modulus of pore fluid is constant. This assumption assures no changes in S-wave velocity, hence I only focus on P-wave velocity models. Also, P-wave velocity is a valuable tool in further studying and describing rocks lithologic properties (Ferguson, 1995) needed in reservoir characterization.

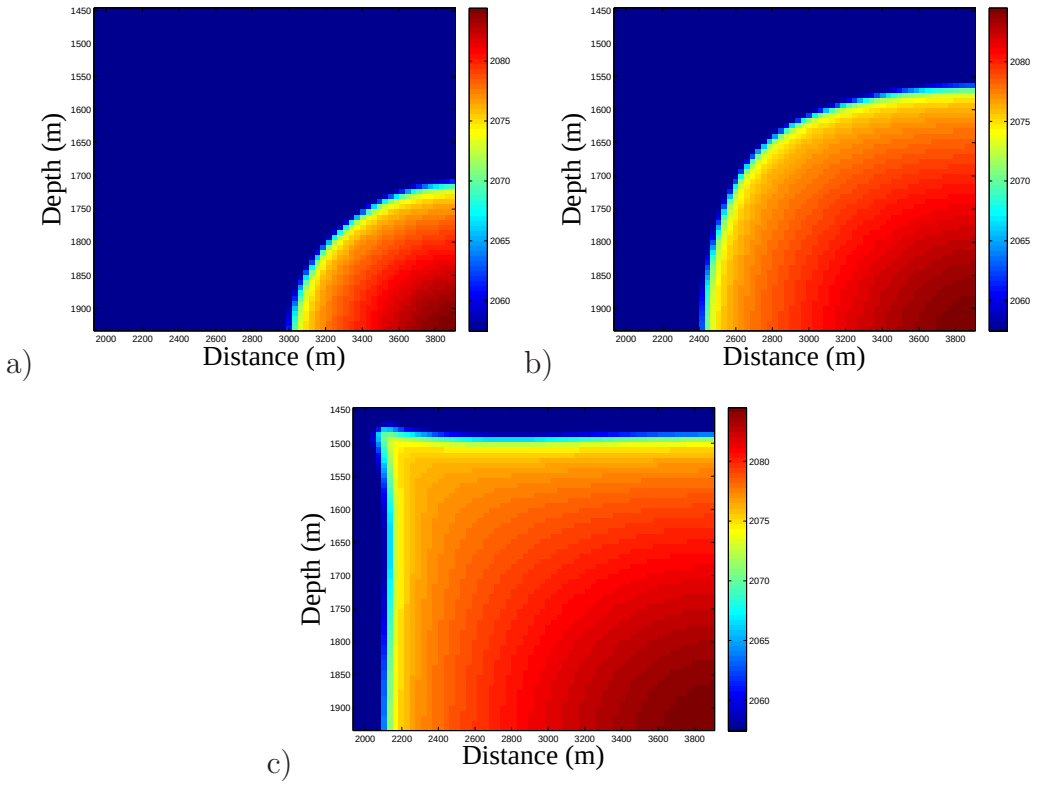


Figure 2.4: Velocity models. P-wave velocity calculated from density saturation using Gassmann's relations. The reservoir is initially 100 % oil saturated. The producer and one injector are located in the upper-left and lower-right corner of the model, respectively. The injector pumps water into the reservoir. The models 2.4(a), 2.4(b) and 2.4(c) show reservoir as P-wave velocity increases towards the producer in time-lapse after day $\tau = 1, 14, 28$, respectively. The color bar denotes velocity values in m/s.

Empirically, P-wave velocity is greater in water than in oil saturated rocks (Kearey et al., 2002). Figures 2.4(a), 2.4(b), and 2.4(c) illustrate exactly this, P-wave velocity decreases from injector to producer after day 1, 14, and 28, respectively. This occurs because the pressure is higher near the injectors and lower near the producer.

2.4.3 Seismic modelling

To obtain time-lapse seismic sections the above P-wave velocity model is padded. Linear velocity (gradual change in velocity values) is applied above the reservoir.

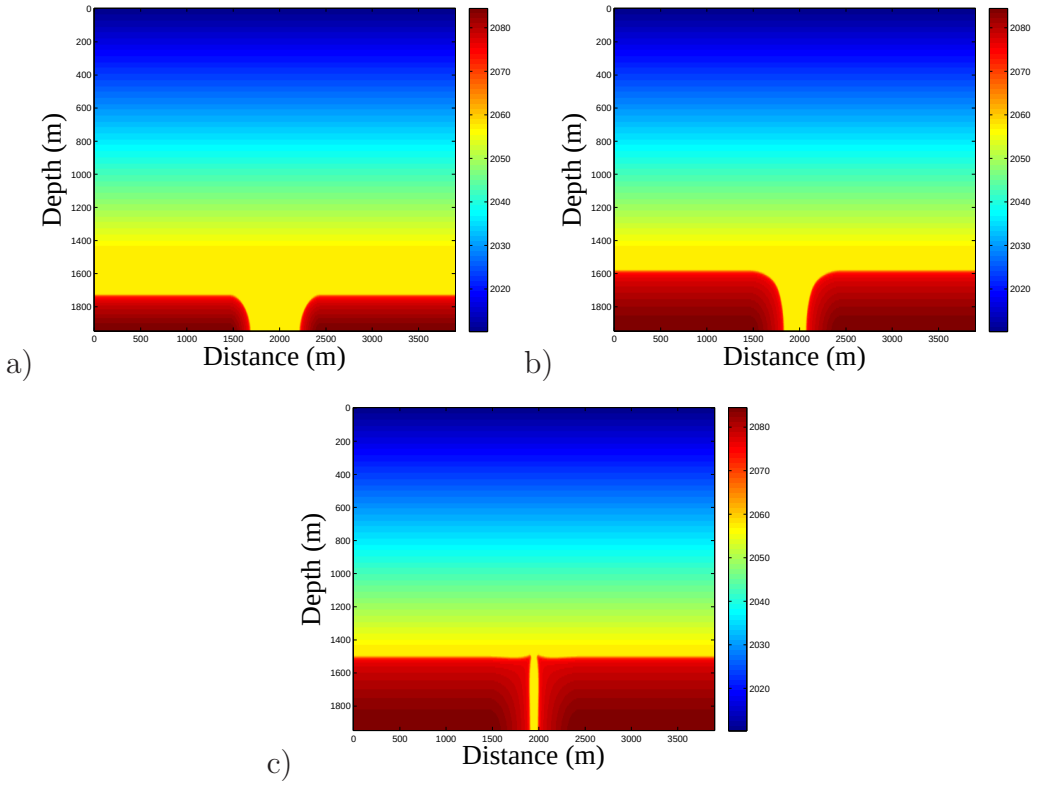


Figure 2.5: Padded velocity models used in generating seismic models: The profiles 2.5(a), 2.5(b) and 2.5(c) show reservoir as water saturation increases the P-wave velocity decreases from injector to producer in time-lapse steps after day $\tau = 1, 14, 28$, respectively. Note previous figures only show right hand side of the reservoir with one injector and the producer, where these figures show the entire reservoir. The producer is located in the centre and two injectors in the right and left hand side corners. The color bar denotes velocity values in m/s.

Figure 2.5(a), 2.5(b) and 2.5(c) illustrate padded P-wave velocity models now showing two injectors and one producer scenario after day 1, 14, and 28, respectively. These figures

are not to be confused with previous figures that only show right hand side of the reservoir with one injector and the producer. Recall, two injectors are situated in lower left and right hand corners. One producer is at a half way distance between injectors. Note the same trend in velocity measurements when compared to Figure 2.4.

Firstly, the above P-wave velocity models are used to create 2D exploding reflector models employing function *afd_explode* from the MATLAB CREWES Project toolbox. The wavefield is propagated in depth using finite difference method, and when convolved with a minimum phase wavelet produces a seismogram in acoustic medium. As model forces, density saturation is set constant. Samples are taken every $4ms$.

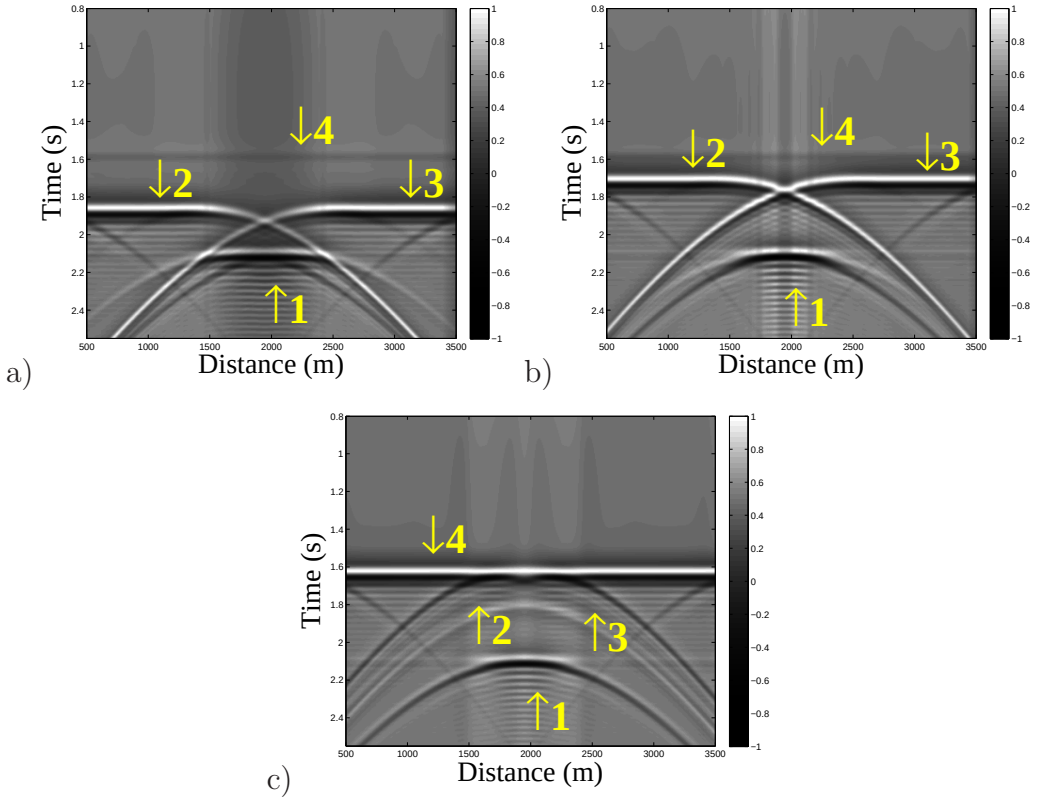


Figure 2.6: 2D exploding reflector seismic models. The profiles 2.6(a), 2.6(b) and 2.6(c) show reservoir in time-lapse after day $\tau = 1, 14, 28$, respectively. The arrows 4 and 1 point to the reservoir top and bottom, respectively. The arrows 2 and 3 point to the two waterfronts. Note waterfronts progress upwards in time-lapse.

Figure 2.6(a) shows the exploding reflector model after day 1. Observe the top and the bottom of the reservoir at about $1.6s$ and at about $2.1s$, denoted by arrows 4 and 1, respectively, that stay stationary until day 28. Both waterfronts, denoted by arrows 2 and 3, are seen at about $1.85s$. Figure 2.6(b) illustrates seismic responses after day 14. Note waterfronts to move upward with water injection and appear sooner at about $1.7s$. Figure 2.6(c) captures seismic responses after day 28. Observe a water breakthrough at the producer. The reservoir bottom is indicated as a strong low followed by a strong high amplitude. The amplitude dims as water saturation increases. The reservoir top is indicated by a relatively strong and high amplitude. The amplitudes dim as waterfront reaches the producer. Both waterfronts are captured by high amplitude, observed from sooner to later traveltime arrivals, creating a bow-tie effect. Also, note the reservoir top and bottom and waterfronts appear as reflection coefficients of opposite polarity. They are positive at the reservoir top and bottom and negative at waterfronts.

Then, the above 2D P-wave velocity model is extended to a 3D model in MATLAB. The model is a rectangular box with dimensions $512 \times 256 \times 512$. This model is used in generating 3C-3D seismic models also employing finite difference algorithm using $Tiger$, commercial software designed by SINTEF Petroleum Research of Trondheim in Norway. These models assume an elastic medium.

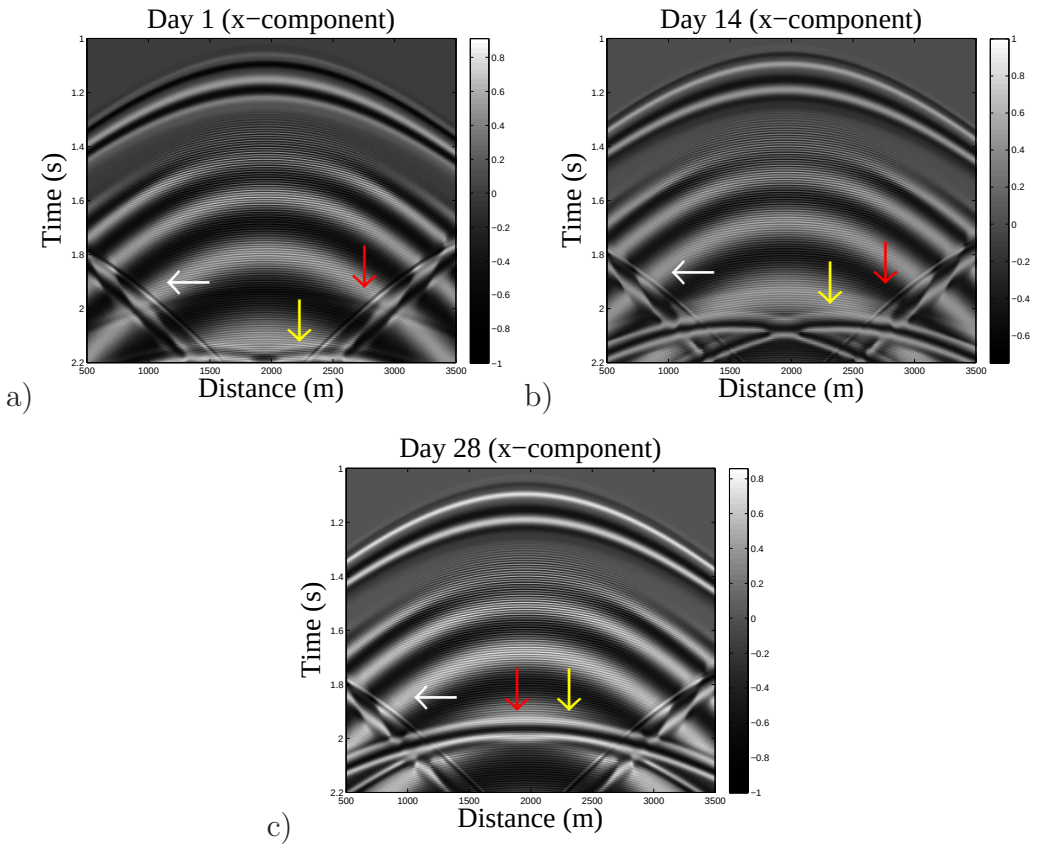


Figure 2.7: 3C-3D shot gather models: x-component seismic models in elastic medium: The x-component captures shear waves. The red arrow points to the top of the reservoir. The yellow arrow points towards two waterfronts. The waterfronts propagate upwards in time-lapse. Both are projections of P-wave velocity onto the shear wave velocity mode. The white arrow marks the reservoir boundary.

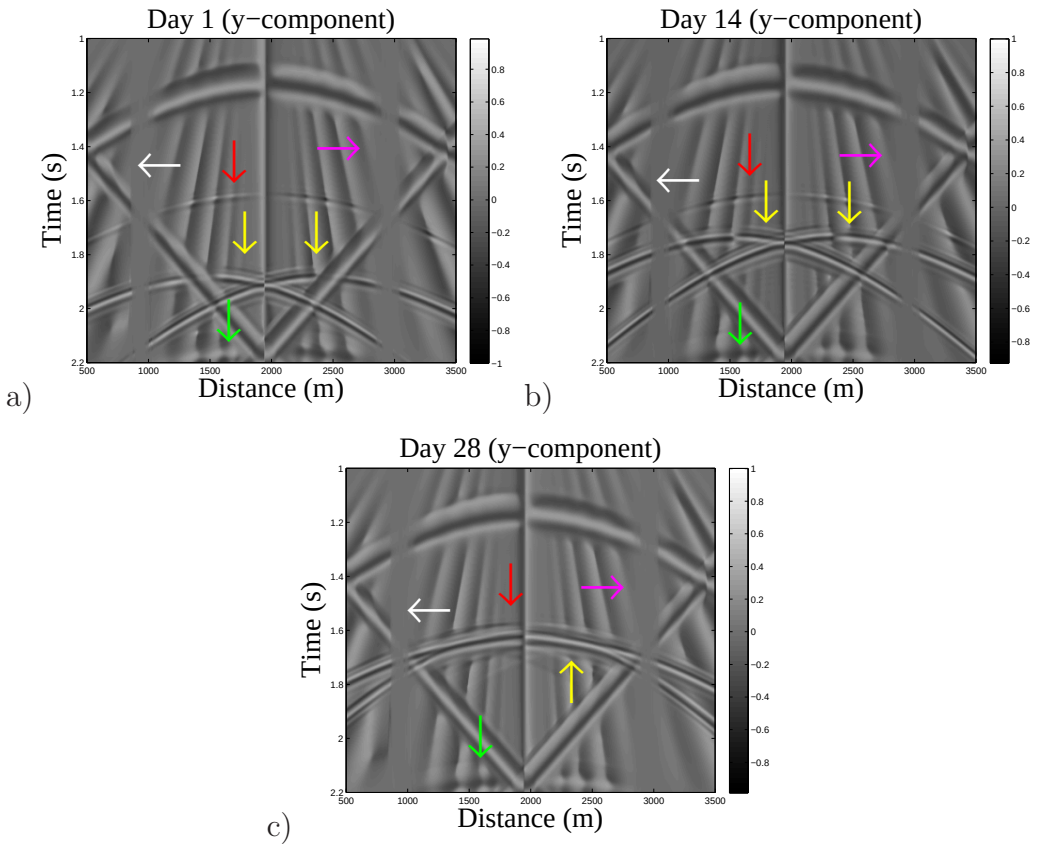


Figure 2.8: 3C-3D shot gather models: y-component seismic models in elastic medium: The y-component captures converted waves. The red and green arrows point to the top and bottom of the reservoir, respectively. The yellow arrows point towards two waterfronts. The waterfronts propagate upwards in time-lapse. The white and magenta arrows mark the reservoir boundary and numerical artifacts, respectively.

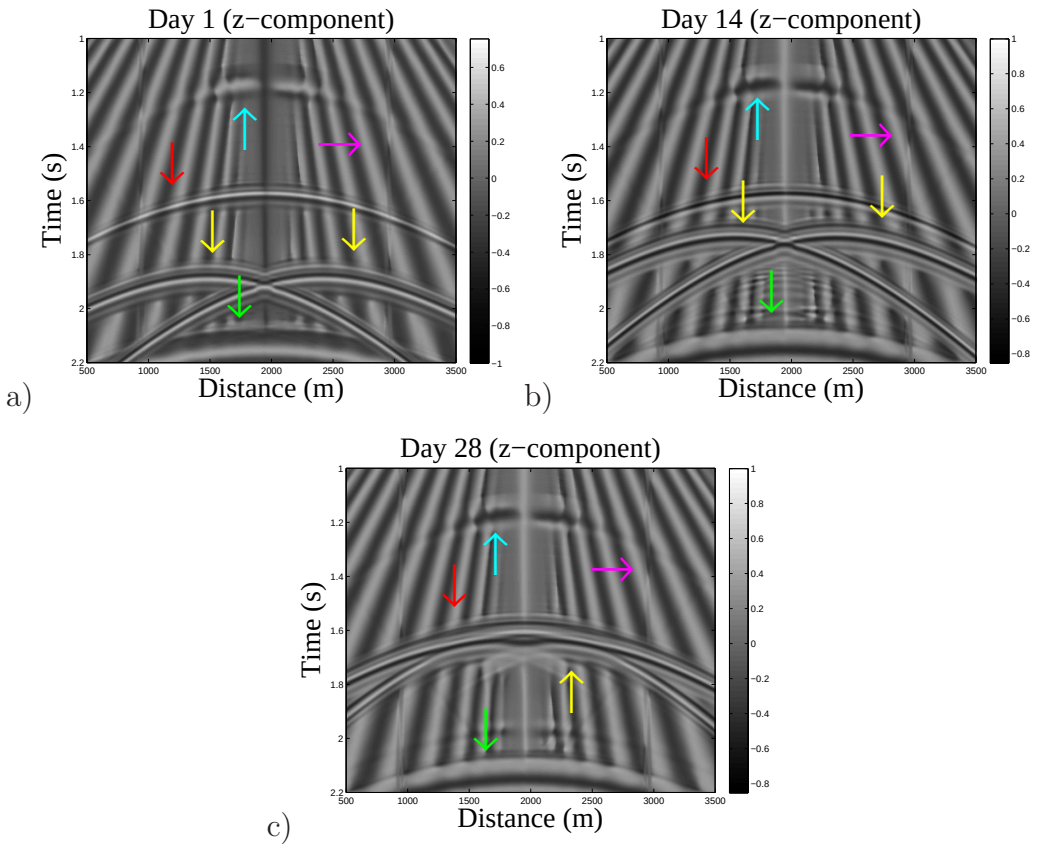


Figure 2.9: 3C-3D shot gather models: z-component seismic models in elastic medium: The red and green arrows point to the top and bottom of the reservoir, respectively. The yellow arrows point towards two waterfronts. The waterfronts on elastic models also progress upwards in time-lapse after day 1, 14 and 28. The 3C-3D models plot more details, hence I see numerical artifacts and projection of shear waves on the vertical component, pointed to by magenta and turquoise arrows, respectively.

Now, observe each component individually. Figures 2.7(a), 2.7(b) and 2.7(c), x-component seismic models, mainly show S-waves. At about $1.8s$ to about $2.2s$ reservoir boundaries, denoted by white arrow, appear as slanted linear events. A very weak projection of P-waves from z-component is seen at about $2.0s$ and at about $2.2s$. The two projections are inferred to be reservoir top and waterfronts, denoted by red and yel-

low arrows, respectively. Observed in time-lapse waterfronts progress upwards. Figures 2.8(a), 2.8(b) and 2.8(c), y-component seismic models, capture converted waves, that is P-waves reflected as S-waves. Both reservoir boundaries show at about $1.4s$ to about $2.2s$ also as slanted linear events annotated by the white arrow. I note reservoir top and bottom, pointed to by the red and green arrows, at about $1.6s$ and $2.1s$, respectively. Also note the two waterfronts, pointed to by the yellow arrows, progressing upwards with time. Numerical artifacts, denoted by magenta arrow, are present at about $800m$ and $3000m$ on the distance axis. Figures 2.9(a), 2.9(b) and 2.9(c) are z-component seismic models. These models are directly comparable to the acoustic models if I observe seismic models immediately under the shot location to depth. The reservoir top and bottom, at about $1.6s$ and at about $2.1s$, respectively, show on 3D elastic models as stationary as well. In Figure 2.9 the reservoir top and bottom are denoted by red and green arrows, respectively. The reservoir bottom is characterized by a set of high-low-high amplitudes from sooner to later traveltime arrivals. The reservoir top characterized by a set of low-high-low amplitudes, is smeared by a set of high-low-high amplitudes after 28 days. The waterfronts, denoted by yellow arrows, as in 2D models also create a bow-tie effect. Both waterfronts show as high amplitudes and progress upwards in time-lapse. The same pattern of reversed polarity between reservoir top and bottom and waterfronts still applies. Also, note S-wave projection from x-component, marked by the turquoise arrow, at about $1.18s$, stationary in time-lapse. Again, numerical artifacts, denoted by magenta arrow, are present at about $800m$ and $3000m$ on the distance axis.

The acoustic and elastic medium models reflect the major expected events, such as the two waterfronts, reservoir top and bottom. I do note more details on the 3C-3D plots. The two approaches both prove to be valuable and its use depends on the reservoir characterization study. The examples prove workflow feasible and expectations verified. Generation of seismic models allows seismic difference analyses.

2.5 Discussion

In practice, reservoirs are neither commonly homogeneous nor 100 % oil saturated. In reality, viscosity, porosity, and permeability are almost never completely uniform. The phases are not immiscible and incompressible, namely there are blending and density changes. Further, water and oil saturations are not fully irreducible, that is all of reservoir oil is not fully displaceable by water. Since the study is a model of workflow, it only lasts 28 days. Even though the model assumes the above, it does not considerably affect general trends and the workflow can be applied to real data sets.

2.6 Chapter summary

A time-lapse study is done on a model of a reservoir employing one producing and two injecting wells. The study consists of: numerical simulation, Gassmann's relations and finite difference algorithms. The numerical simulation of fluid flow produces saturation and pressure models. Then, the saturation models deliver P-wave velocity models as a result of Gassmann's relations. Further, P-velocity models, through finite difference algorithms, generate 2D acoustic and 3C-3D elastic seismic models.

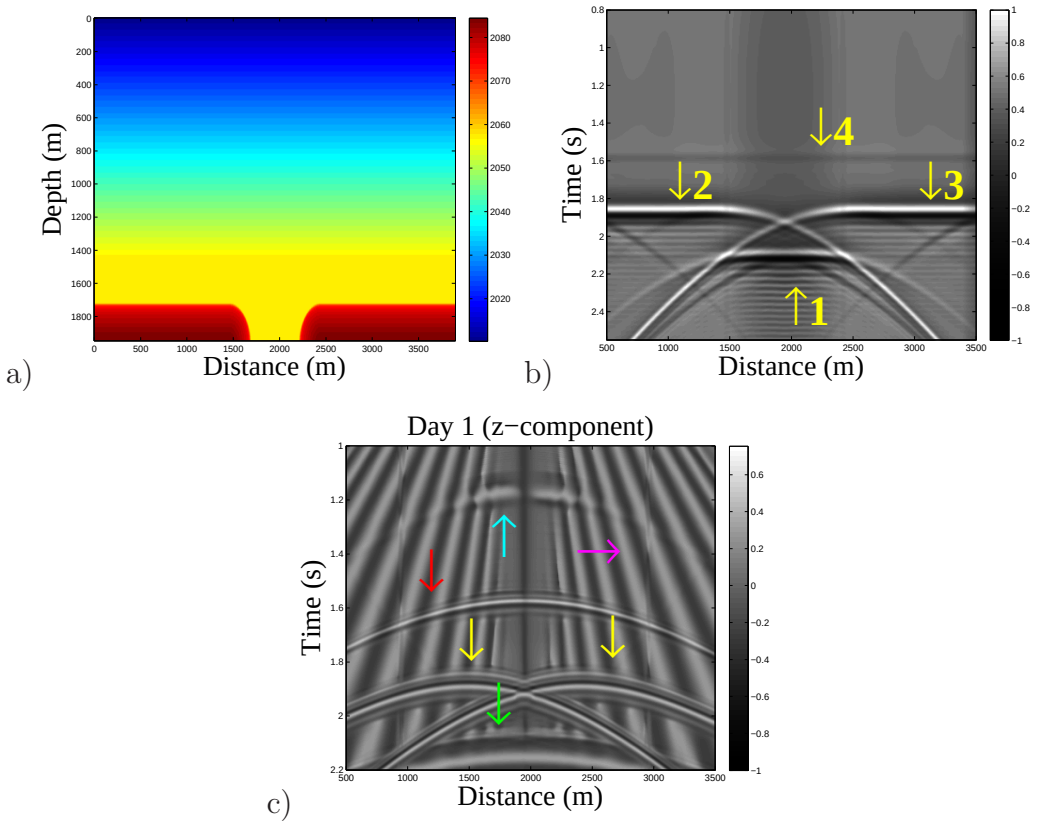


Figure 2.10: Summary of numerical fluid flow modelling and its seismic response: modelling the reservoir after day 1: a) velocity model, b) 2D seismic model and c) 3C-3D seismic model.

Figure 2.10 captures results of the reservoir after day 1, hence theoretical concepts are verified through numerical examples. There are subtle similarities and differences between acoustic and elastic models. Study shows both, acoustic and elastic models, to be assets to reservoir characterization and gives tools for further thesis work development specifically, seismic differencing analyses.

Chapter 3

Conventional and Inverse Data Space Seismic Differencing

3.1 Introduction

Seismic imaging is a process through which waves recorded on the surface are mapped into an image of the subsurface (Ferguson, 2010). In particular, seismic imaging is used in hydrocarbon reservoir exploration and development (Huang et al., 1998). Its success is directly related to familiarity with reservoirs (Lines and Newrick, 2004). As mentioned in Chapter 2, modelling is a simple task, when dealing with reservoirs with long production history; however, it is challenging when dealing with reservoirs with short to no production history (Holstein, 2007). To study reservoirs and their properties actual or synthetic testing and modelling take place (Cosse, 1993). This process generates subsurface images used for reservoir monitoring or evaluation (Ferguson, 2010).

As production influences reservoir properties with fluid flow displacement, reservoirs are observed in time-lapse (Cheng et al., 2009). Time-lapse observation images are generated in various production days (Zou et al., 2004). The spatio-temporal changes in hydrocarbon reserves are evaluated to define their effects on reservoir properties and further exploitation planning (Jin and Chen, 2008). Numerous analysis procedures exist to optimize production. Some of these analyses include: core, pressure transient, fracture, stratigraphic dip, conventional differencing, etc (Holstein, 2007). Of these analyses procedures, conventional seismic differencing is of interest in this thesis work.

Many studies focus on seismic differencing methods and analyses. Huang et al. (1998) observe amplitude patterns on seismic difference models in time-lapse. They generate three synthetic models for comparison. Their models are based on: production history

only, measured differences in physical parameters only and combination of the previous two. The study points the importance of monitoring/modelling differences in a producing reservoir for further reservoir characterization. Schinelli (2006) highlights complex seismic attributes, such as signal to noise ratio and tuning, to be valuable when apprising conventional amplitude differencing. He proposes complex attributes to significantly limit fluid flow displacement observation on simple amplitude subtraction imaging. Bertrand et al. (2005) present a method to highlight amplitude differencing through the removal of traveltime between baseline and monitor surveys. Jin and Chen (2008) propose methods to enhance time-lapse seismic anomaly and reduce noise decomposing on differenced models using wavelet transform and filters.

Berkhout and Verschuur (2005) develop a seismic processing algorithm and call it inverse data space (IDS). Their method is based on linear algebra where they situate data into a matrix and then generate inverse data space through matrix inversion. Berkhout and Verschuur (2005) illustrate this space as suitable for data processing in time-lapse, specifically to surface related multiple elimination (SRME).

In this thesis, time-lapse conventional and IDS differencing are evaluated through numerical experiments of 2D data set. Namely, I evaluate some aspects of Berkhout and Verschuur (2005) data processing method for use in reservoir characterization. I evaluate how the inverse data space directly mapped into estimates of time-lapse differences (Innanen, 2009) benefit reservoir studies. MATLAB code is used to illustrate both 2D data imaging after conventional and IDS differencing. I analyze fluid flow displacement imaging in time-lapse. Results of conventional and IDS differencing are compared and examined for use in reservoir characterization.

The designed time-lapse study can be summarized in a workflow.

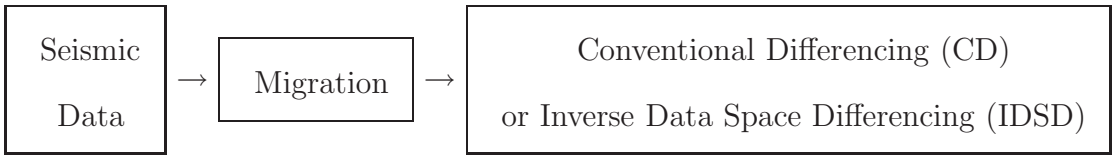


Table 3.1: Workflow III. The seismic differencing in this chapter will be studied using this workflow. Seismic data matrices are migrated and then conventionally (CD) and IDS differenced. The results from CD and IDSD are analyzed and compared.

Table 3.1 captures workflow applied to 2D velocity plots: I) seismic modelling, II) migration III) conventional and IV) inverse data space differencing. The workflow generates time-lapse synthetic seismic models. Then, time-lapse seismic data steps are migrated employing a choice algorithm. The resulting models are then differenced employing CD and IDSD algorithms. The output models are analyzed and compared.

Both conventional and IDS differencing methods are valuable. The conventional differencing is expected to be fast. It will trace large scale reservoir characteristics with fluid flow displacement and no certain amplitude patterns. IDS differencing, although requiring longer computation time, will trace large and fine scale reservoir characteristics with fluid flow displacement.

3.2 Theory

The 10^{th} Comparative Solution Project data set is used again (Christie and Blunt, 2001). Recall the data set models a reservoir with one producing and two injecting wells as defined in Chapter 2. The workflow is to take velocity models, to zero-offset seismic models, migrate them and allow differencing, that is conventionally and non-conventionally (IDSD).

3.2.1 Velocity and seismic synthetics modelling

A laterally varying 2D velocity model is assumed to model the above reservoir in time-lapse. Suppose waterfronts to dip at about 90°. Passing the velocity models to finite difference method I generate synthetic data. The resulting synthetics are 2D zero-offset exploding reflector models.

3.2.2 Migration modelling

Stoffa et al. (1990) introduce the split-step Fourier (SSF) migration algorithm, which handles lateral changes in velocity at each depth level. Also, the algorithm takes into account dipping events. Assume 2D propagation of P-waves in acoustic medium and constant density. Wave propagation is defined as (Stoffa et al., 1990) :

$$\nabla^2 p - u^2 \frac{\partial^2}{\partial t^2} p = 0, \tag{3.1}$$

where t, $p = p(x, z, t)$ and $u = u(x, z)$ are time, pressure and slowness, respectively. The inverse of the half of the propagation velocity $u(x, z) = 2/\alpha(x, z)$, where α, x, z are velocity, horizontal and vertical distance, respectively, denotes slowness. The above statement is imperative by the exploding reflector model when zero-offset data is migrated (Mi, 2002). As the migration by SSF takes place partially in the frequency domain, equation (3.1) is Fourier transformed to:

$$\nabla^2 P + \omega^2 u^2 P = 0, \tag{3.2}$$

where ω is frequency and $P = P(x, z, \omega) = \int_{-\infty}^{+\infty} p(x, z, t) e^{-i\omega t}\, dt$. Now, Stoffa et al. (1990) decompose the slowness term from equation (3.2) in two components:

$$u(x, z) = u_0(z) + \Delta u(x, z), \tag{3.3}$$

where $u_0(z)$ and $\Delta u(x, z)$ are the reference and perturbation slowness. The reference slowness in equation (3.3) is the mean of u(x,z) and as per Ferguson and Margrave (1999)

named stationary. The perturbation slowness accommodates all velocity variations, hence is non-stationary (Ferguson and Margrave, 1999). Thus the homogeneous wave equation transforms into the inhomogeneous, constant-slowness wave equation (Stoffa et al., 1990):

$$\nabla^2 P + \omega^2 u_0^2 P = -U(x, z, \omega), \tag{3.4}$$

where $U(x, z, \omega) = \omega^2[2u_0 \Delta u(x, z, \omega) + \Delta u^2(x, z, \omega)]P$ is a source-like term. The second order term in equation (3.4) is ignored as perturbation slowness is small when compared to the reference slowness.

The solution of equation (3.4) is summarized as (Du, 2007):

I. Transform wavefield from the spatial to the wavenumber domain and apply a phase-shift based on the vertical wavenumber, k_z, computed by the reference slowness:

$$P^*(z + \Delta z, k_x, \omega) = P(z, k_x, \omega) e^{\pm i \sqrt{\omega^2 u_0^2 - {k_x}^2} \Delta z}, \tag{3.5}$$

where k_x denotes horizontal wavenumber.

II. Inverse Fourier equation (3.5), that is transform $P^*(z + \Delta z, k_x, \omega)$ back to $P^*(z + \Delta z, x, \omega)$ as:

$$P^*(z + \Delta z, x, \omega) = \int_{-\infty}^{+\infty} P^*(z + \Delta z, k_x, \omega) e^{-ik_x x} dk_x. \tag{3.6}$$

III. In the space and frequency domains, generated by equation (3.6), apply a second phase-shift due to the perturbation in the slowness:

$$P(z + \Delta z, x, \omega) = P^*(z + \Delta z, x, \omega) e^{\pm i(\frac{\omega}{\alpha(x,z)} - \frac{\omega}{\alpha_0(z)})\Delta z}. \tag{3.7}$$

Now, integrate equation (3.7) over all frequencies of interest to deliver the migrated data (Mi, 2002).

3.2.3 CD

Time-lapse migrated seismic models are presented as matrices D_i, where i denotes time step. These models are differenced employing conventional matrix subtraction:

$$D_{diff} = D_{i+1} - D_i. \tag{3.8}$$

Equation (3.8) captures large scale physical changes of reservoir as production progresses. Namely, hydrocarbon volume and its displacement changes are expected to be interpretable for use in enhanced recovery schemes development and monitoring.

3.2.4 IDSD

Berkhout and Verschuur (2005) developed an algorithm for surface related multiple elimination (SRME) using the concept of the inverse data space. I use their algorithm and modify its parts to develop the inverse data space differencing algorithm. Here, I give a brief review of SRME method. Berkhout and Verschuur (2005) define the model for primary wavefields as:

$$W(z_0, z_0) = R(z_0)T_0(z_0, z_0)S(z_0), \tag{3.9}$$

where R, T_0 and S denote receiver array, transfer matrix and source array, respectively. The variable z_0 indicates that source and receivers are situated at the surface. The transfer matrix consists of columns that store one spatial impulse response. In general, the transfer matrix in equation (3.9) relates input and output data based on the subsurface conditions. Now, Berkhout and Verschuur (2005) create the feedback model accounting for physics related to surface multiples and complicated data (Berkhout and Verschuur, 2005):

$$W = W_0 + (W_0A)W_0 + (W_0A)^2W_0 + \cdots, \tag{3.10}$$

where $A = S^{-1}Q^*W^{-1}$ and Q^* is surface reflectivity and W_0 contains primaries only. The surface operator A does not contain traveltime. The equation (3.10) can also be written

as (Berkhout and Verschuur, 2005):

$$W = [I - W_0 A]^{-1} W_0, \tag{3.11}$$

as a continuous form through the use of binomial expansion. Multiplication with $(W_0 A)$ in equation (3.10) and (3.11) represents spatial convolution, that is adding one roundtrip through subsurface (Berkhout and Verschuur, 2005). Simplification of equation (3.11), gives:

$$W = W_0 + W_0 A W, \tag{3.12}$$

that is a multiple scattering equation of known Lippmann-Schwinger structure (Innanen, 2010). Equation (3.12) represents the theoretical bases of multiple removal algorithms such as SRME (Berkhout, 2006). Equation (3.12) is the surface-related version of equation (3.11). Employing matrix inversion, multiple scattering data in forward data space (FDS), described by equation (3.11), transforms into the inverse data space (IDS) (Berkhout, 2006) as:

$$W^{-1} = W_0^{-1} - A. \tag{3.13}$$

Equation (3.13) describes a much simpler data set based on surface-free earth response and surface related properties at and around zero time.

In time-lapse studies baseline and monitor surveys are defined. The baseline migrated survey is denoted by a matrix D_i and any subsequent time-lapse monitor surveys are denoted by D_{i+n} for i starting at 0 and n being the number of day after baseline survey took place.

Further, to analyze data in time-lapse, the migrated baseline survey can be written as (Berkhout, 2006):

$$D_i = W_0 + W_0 A W \tag{3.14}$$

and the monitor surveys as (Berkhout, 2006):

$$D_{i+n} = W_0^{'} + W_0^{'} A^{'} W^{'}. \tag{3.15}$$

Due to change in acquisition system and surface conditions A and $A^{'}$ can be different for real data sets, however, dealing with synthetics allows to keep them constant.

To account for any reservoir parameters both equations, (3.14) and (3.15) can be further complemented by specific variables of interest (Berkhout and Verschuur, 2005):

$$D_{ids} = (W_0 + W_0 A^{'} W^{\prime}) - (\delta W_0 + \delta W_0 A^{'} W^{'}), \tag{3.16}$$

where δW_0 denotes reservoir and overburden responses due to production.

The use of inverse data space is hence summarized as:

I. Conversion of data from FDS to IDS through least-squares algorithm, that is $D_i => D_i^{-1}$ and $D_{i+1} => D_{i+1}^{-1}$.

II. Separation of surface operators from reflection data in Radon domain.

III. Conversion of reflection data from IDS to FSD, that is is $D_i^{-1} => D_i$ and $D_{i+1}^{-1} => D_{i+1}$.

IV. Identify surface transfer function, in FDS and IDS, that is $T_i = -AD_i$ and $T_i^{'} = -A^{'} D_i^{'}$.

V. Compute difference data employing least-squares subtraction to obtain $\lambda T_i = T_i - F_{ls} T_i^{'}$, where F_{ls} is a scaled version of the correlation between the overburden Green's functions of the baseline and monitor surveys (Berkhout and Verschuur, 2005).

Hence, the inverse data space differencing is executed as:

$$D_{ids} = D_{i+n} - D_i \tag{3.17}$$

or

$$D_{ids} = (W_0 + W_0 A W) - (W_0^{'} + W_0^{'} A^{'} W^{'}), \tag{3.18}$$

where $W_0 = RT_0S$, $W_0^{'} = RT_0^{'}S$, $W = RTS$, $W^{'} = RT^{'}S$, $A = S^{-1}QS^{-1}T^{-1}R^{-1}$ and $A^{'} = S^{-1}QS^{-1}(T^{'})^{-1}R^{-1}$. Equations (3.17) or its expanded version (3.18) describe time-lapse differencing. Considering the outlined use of the inverse data space, equation (3.18)

is simplified as:

$$D_{ids} = (W_\lambda + W_\lambda A W), \tag{3.19}$$

for $\lambda W = R\lambda T_0 S$. Equation (3.19) delivers the final solution of the IDSD algorithm. The IDSD models are expected to capture large and small scale physical changes as well as filter seismic amplitude so it highlights fluid flow changes.

3.3 Examples

3.3.1 Velocity models

As shown in Chapter 2, saturation models, through Gassmann relations, deliver velocity models in time-lapse (Milicevic and Ferguson, 2009).

Recall velocity models are mirrored over the left hand side and padded on bottom and top to accommodate for energy to propagate and avoid wraparound (Ferguson, personal communication). I pad a linear velocity matrix from about $0m$ to about $1450m$ of depth and a constant velocity matrix from about $1950m$ to about $3900m$ of depth.

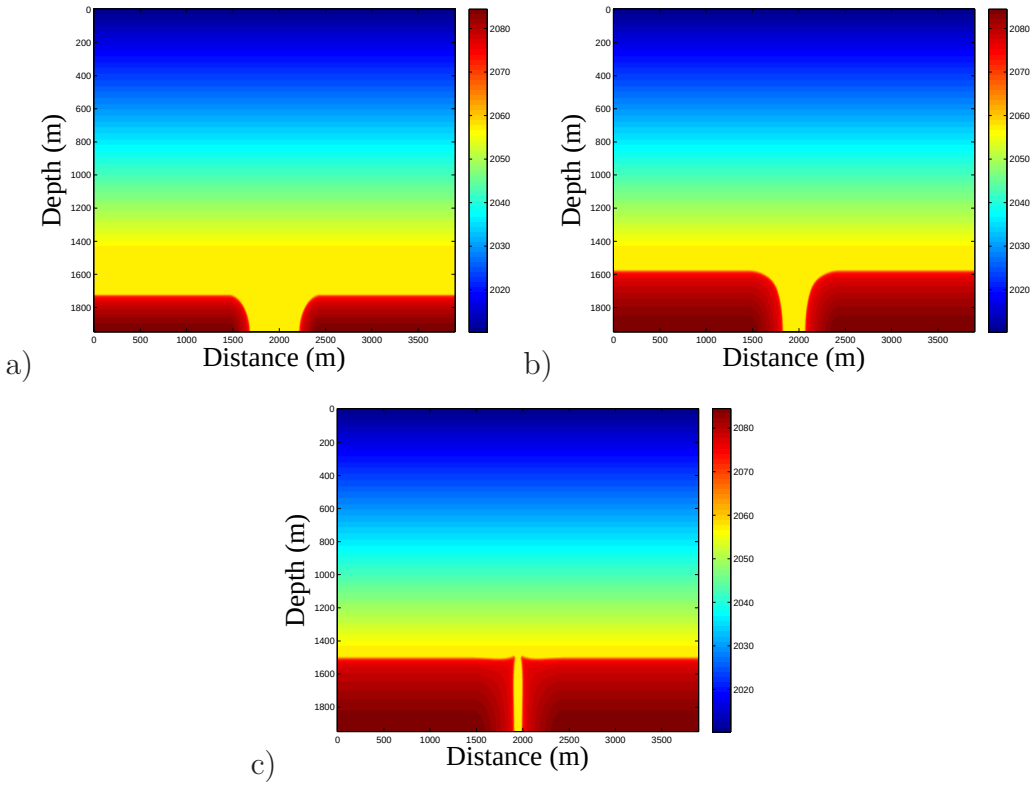


Figure 3.1: Padded velocity models describing 100 % oil saturated sandstone reservoir. Models 3.1(a), 3.1(b) and 3.1(c) show reservoir as water saturation increases. Two injectors are situated in lower left and right hand corners, while producer sits at half distance between them. P-wave velocity decreases from injector to producer in time-lapse steps after day $\tau = 1, 14, 28$, respectively.

Figure 3.1 shows the end result of the modified velocity models after day 1, 14 and 28. The two injectors are situated in the lower left and right hand corners and the producer at their half distance. Note the velocity and water saturation increase with time.

3.3.2 Zero-offset synthetic seismogram models

These velocity models are passed to a finite-difference function, $afd\_explode$, that simulates exploding reflector concept. 2D synthetics are produced as presented in Chapter 2.

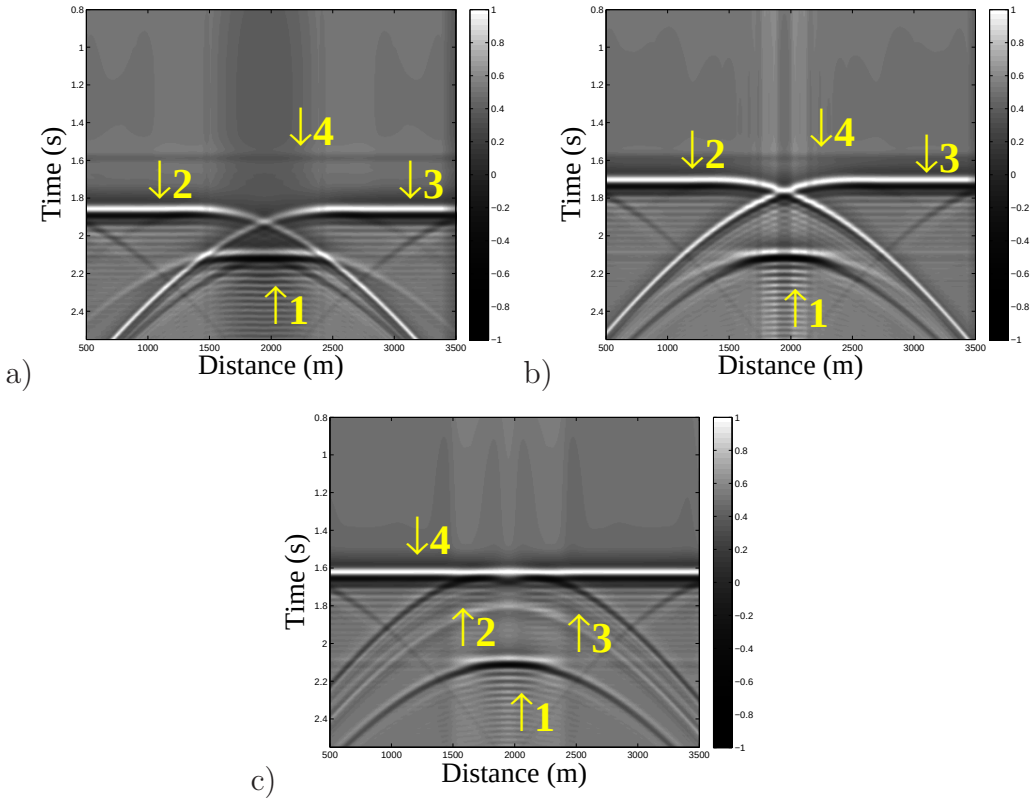


Figure 3.2: 2D synthetic seismic models generated employing exploding reflector algorithm. Models 3.2(a), 3.2(b) and 3.2(c) show reservoir in time-lapse steps after day $\tau = 1, 14, 28$, respectively. Reservoir bottom and top, denoted by arrows 1 and 4, respectively, stay stationary in time. Arrows 2 and 3 mark waterfronts as they progress upward in time. Water saturated zones show linear trends.

Figure 3.2 shows zero-offset synthetics created after day 1, 14 and 28. The reservoir top and bottom are denoted by arrows 4 and 1, respectively. Note reservoir top and bottom as stationary events in time-lapse at about $1.6s$ and $2.1s$, respectively. The two waterfronts are denoted by arrows 2 and 3 on synthetic models and create a bow-tie effect. Note two waterfronts as non-stationary reflections, as they advance in time from about $1.95s$ to about $1.6s$. The reservoir top amplitude is dark gray. The reservoir bottom

amplitude is almost white followed by dark gray-to-black reflection. These two reflections of different polarity are due to high velocity contrast between oil and water saturated zones. The waterfronts are of almost black, white and black amplitude sequence. The overall amplitude of the reservoir dims with water saturation increase. Note horizontal linear trends to reflect injected water.

3.3.3 Migrated models

Previously generated 2D zero-offset synthetics in time, t, and distance, d, domain are converted to frequency, ω, and wavenumber, k_x domain invoking Forward Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) (Ferguson and Margrave, 2005). The ω axis is band-limited and positive and the k_x axis is not centered. Then, data is migrated calling *ss_zero_mig*, a MATLAB routine of the CREWES Project toolbox. The routine performs SSF depth migration (Ferguson, 2009).

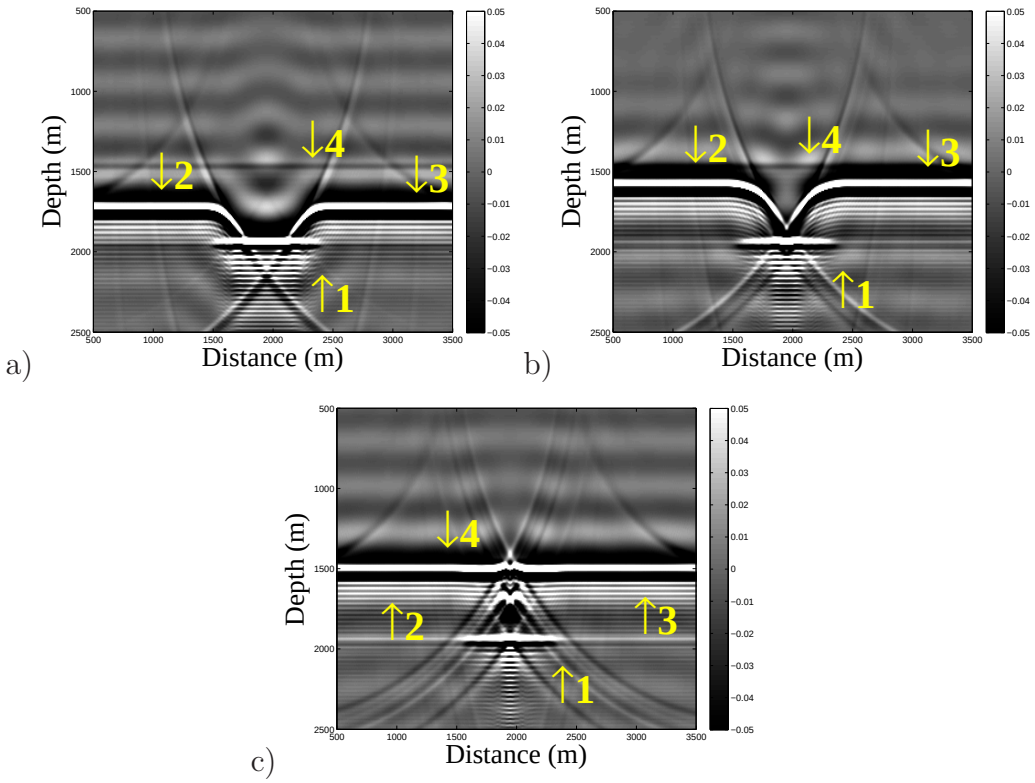


Figure 3.3: Split-step Fourier migrated seismic models generated from velocity and synthetic models. Models in 3.3(a), 3.3(b), 3.3(c) capture flattening of hyperbolic events after day $\tau = 1$, 14 and 28, respectively. Arrows 1 and 4 point to the stationary events reservoir bottom and top, respectively. Arrows 2 and 3 point to two waterfronts propagating upwards in time. Water and oil saturated zones are better focused.

Figure 3.3 illustrates migrated models after day 1, 14 and 28. The expected events, such as reservoir top and bottom, denoted by arrows 4 and 1, respectively, appear stationary. The amplitudes correspond to the amplitudes of the zero-offset unmigrated sections. The reservoir top and bottom are captured at about depth of $1450m$ and $1950m$, respectively. The reservoir top again shows as a horizontal event, but better focused. The reservoir bottom instead of a hyperbola shows as horizontal and also better focused. Its amplitude is purely black followed by a purely white color. The waterfronts, denoted

by arrows 2 and 3, propagate upwards with time. The reservoir overall amplitude still shows linear reflections where saturated with water. These reflections are better focused and of high amplitude.

3.3.4 CD models

I conventionally difference migrated sections.

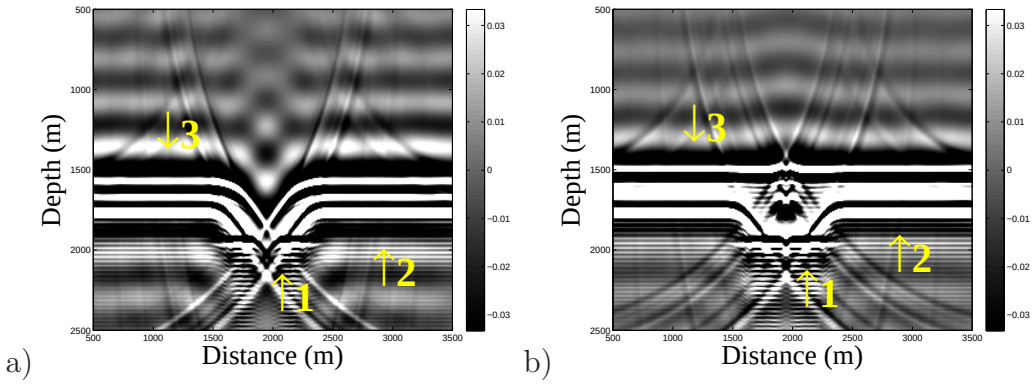


Figure 3.4: Conventionally differenced migrated models. Models 3.4(a) and 3.4(b) capture conventional difference of models after days 1 and 14 and days 1 and 28, respectively. Arrow 1 denotes reservoir bottom, whereas, no reservoir top reflection can be identified. Arrows 2 and 3 mark two waterfronts corresponding to differenced models. It is hard to identify areas of remaining production on both conventional differenced models and fluid flow changes in time are hard to see.

Figure 3.4(a) is a plot of conventional difference between day 1 and 14. The reservoir top is not identifiable, as it is of the same amplitude on both models. The amplitude of reservoir bottom is of reverse polarity when compared to migrated sections, namely black and white. The reservoir bottom, denoted by arrow 1, is not a horizontal event, but an intersection of curves described as square root function and its inverse. The waterfronts,

denoted by arrows 2 and 3, are of the same amplitude as they are on migrated section of day 1 and 14, at depths of 1750m and 1625m, respectively.

Figure 3.4(b) is a plot of conventional difference between days 1 and 28. Similarly, reservoir top cannot be observed on differenced models. Reservoir bottom follows the same pattern as in Figure 3.4(a). The reservoir bottom is almost entirely horizontal event, it is an intersection of two almost straight lines. The waterfronts are of the same amplitude as they are on migrated section of day 1 and 28, at depths of 1750m and 1450m, respectively.

3.3.5 IDSD models

I difference the same set of migrated sections employing the non-conventional differencing method, IDSD, based on Berkhout and Verschuur (2005) work.

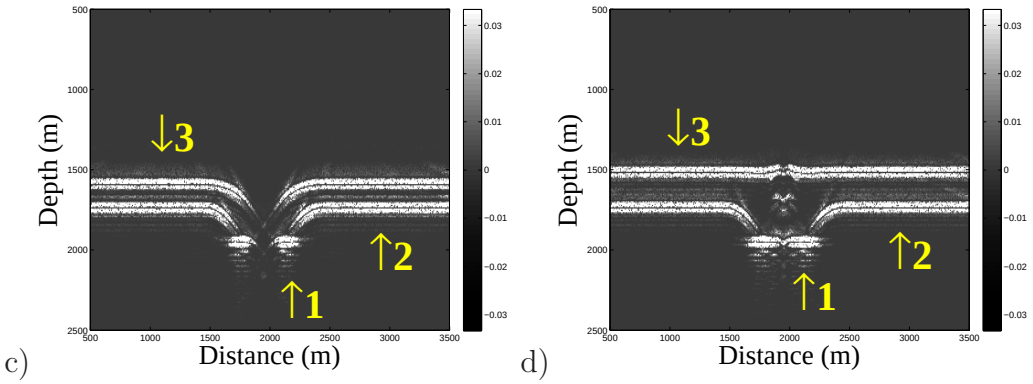


Figure 3.5: IDSD models. Models 3.5(a) and 3.5(b) capture IDSD of models after days 1 and 14 and days 1 and 28, respectively. Arrow 1 denotes reservoir bottom, whereas, no reservoir top reflection can be identified. Arrows 2 and 3 mark two waterfronts corresponding to differenced models. Produced areas are easily identifiable on IDS differenced models. It is hard to identify areas of remaining production on both conventional and IDS differenced models. The fluid flow change becomes more obvious and easier to see.

Figure 3.5(a) is a plot of IDS difference between days 1 and 14. Reservoir top cannot be observed where as reservoir bottom is a white linear reflection. It is clear to note the waterfronts, denoted by arrows 2 and 3, belong to their progression after day 1 and 14. The area between waterfronts is defined by weak white amplitude. It has been oil saturated and replaced by water as waterfronts progress after day 1 to day 14. The area below waterfront 2 shows even more weak white amplitude, associated with primarily water saturation.

Figure 3.5(b) is a plot of IDS difference between days 1 and 28. Reservoir bottom again shows as a strong white linear amplitude and reservoir top cannot be identified. Waterfronts, denoted by arrows 2 and 3, image their progression after day 1 and 28. The area between waterfronts is defined by weak white amplitude and it is the area of oil produced between day 1 and 28. Water saturation prior to day 14 shows as weak white amplitude.

Conventional differencing proves to be of limited use in reservoir characterization as it captures no certain amplitude patterns and high noise levels. IDS differencing proves to be an improved tool in reservoir characterization although identifying areas of remaining oil reserves seems hard. Hence, method triggers future development. Fluid flow changes in time-lapse are easy to spot.

3.4 Chapter summary

Seismic difference analysis study is performed on a 100 % oil saturated reservoir in time-lapse. 2D variable velocity matrix is created. Velocity matrix, invoking finite-difference algorithm and simulating exploding reflector concept is used to generate zero-offset synthetics in time-lapse. Synthetics are migrated using split-step Fourier algorithm. Migrated sections are conventionally and non-conventionally differenced and compared.

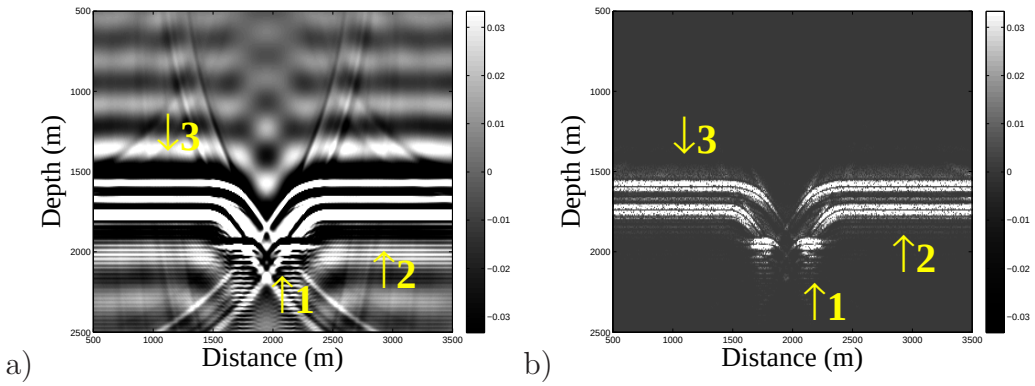


Figure 3.6: IDS models offer cleaned image of differences when compared to conventionally differenced models for difference of days 1 and 14. Arrow 1 denotes reservoir bottom, whereas, no reservoir top reflection can be identified. Arrows 2 and 3 mark two waterfronts corresponding to differenced models. The fluid flow change is easy to observe.

Figure 3.6 is an example of conventional seismic differencing and non-conventional differencing, IDS, for days 1 and 14. Conventional seismic differencing presents little value to reservoir characterization and optimization as it captures high noise and error levels masking fluid flow changes. IDS differencing presents some improvement to reservoir characterization. It needs advancements as remaining oil in reservoir is hard to interpret. Cross-correlation operation can be used for further advancements and better focusing of fluid flow changes in time-lapse studies.

Chapter 4

Cross-correlation Imaging Condition Differencing and PSDM

4.1 Introduction

The cross-correlation imaging condition of seismic depth migration essentially uses a model of the source wavefield to identify reflection amplitudes on an input data set. This identification occurs most simply through cross-correlation, and this is a process that relies on good phase fidelity in both the model and the data; reflection amplitude in the data whose corresponding phase matches that of source amplitude is mapped to zero lag. Data at zero lag are then mapped to the image space and, when this is done for all depth grid levels, the migrated seismic image is produced.

Central to the imaging condition, then, that reflection data are identified, captured, and used in the image, while all matching data are discarded. In application to time-lapse analysis, the imaging idea seems well suited as it is sensitive to phase, and phase is recorded most reliably in the seismic responses. Rather than forward model a reflection and then look for a similar reflection shape in the data as shown in conventional differencing of Chapter 3, it seems reasonable to use a reference data set (with all reflections, multiples, and so on) - the completed seismic wavefield instead of a model. Used instead of a source model in the imaging condition, the reference wavefield will act to find all similar energy in the monitor survey and then map that energy to zero lag. A simple modification to this approach causes all similar events to be discarded and all difference, in fluid flow, to be imaged.

In this section conventional differencing is reviewed and four new differencing methods are proposed:

1. cross-correlation differencing (CCD),
2. pseudo cross-correlation differencing (PCCD),
3. conventional imaging condition differencing (CICD) and
4. imaging condition differencing (ICD).

The CCD is an algorithm implemented in the time domain. It calculates the cross-correlation of two time-lapse steps and multiplies it by the Gaussian filter (chosen for its robustness and low computational cost) to notch out data at zero lag. The algorithm concludes by transformation to the time domain through inverse cross-correlation and multiplication by the conventional difference. As the CCD algorithm is computationally costly the same method is implemented in the frequency domain. I anticipate saving computational time by replacing matrix inversion with inverse Fourier transformation.

Theoretically, cross-correction in the time domain is equivalent to multiplication by a complex conjugate in the frequency domain. Hence to indicate a direct correspondence between the time and frequency domain operations, I name cross-correlation in frequency domain pseudo cross-correlation and develop the PCCD algorithm. The PCCD algorithm computes pseudo cross-correlation of two Fourier transformed time-lapse steps, and by convolving it with the Gaussian filter notches out data at zero lag. Inverse Fourier transform is applied to the result to invert the pseudo cross-correlation and restore phase. This result is then scalar multiplied by the conventional difference.

The difference filtering output (CCD and PCCD models) must be passed to the migration algorithm manually by the user, hence to eliminate the need for user to do so I develop another algorithm called CICD. CICD algorithm is a pilot algorithm to ICD. It performs pre-stack depth migration and conventional differencing at the imaging condition. As CICD proves to be robust, I develop ICD that also performs pre-stack depth migration and replaces conventional with non-conventional differencing (PCCD)

at the imaging condition. CICD and ICD algorithm completely eliminate need for the user to manually pass the output. ICD also improves resolution and computational cost by this automated approach.

4.2 Theory

I recall conventional differencing and discuss the theoretical development of four non-conventional differencing algorithms. Since I modify pre-stack depth migration to develop non-conventional differencing algorithms, I review the concept of the imaging condition, which includes a review of convolution and cross-correlation. I visit the definition of Gaussian filter in the time and frequency domain as it is used to scratch data from the zero lag in cross-correlation space. I conclude each non-conventional algorithm by estimating its computational cost.

4.2.1 Conventional differencing

I define baseline and monitor surveys. The baseline survey represents the reference data set, recorded before any changes to fluid flow (in pre-production stage) could happen. The monitor survey represents the data set recorded after time passed and allows for possible changes in fluid flow (during production). Conventional differencing is obtained employing conventional matrix subtraction (Vracar and Ferguson, 2010) as:

$$\mathbf{D} = \mathbf{M} - \mathbf{B}, \tag{4.1}$$

where $\mathbf{B}$ and $\mathbf{M}$ define baseline and monitor surveys, respectively. Baseline and monitor surveys are recorded in time.

Baseline and monitor survey	→	Migrate	→	Difference

Table 4.1: Workflow IV: Conventional differencing workflow. The baseline and monitor surveys are migrated and conventionally differenced.

The conventional seismic differencing is shown in Table 4.1 through a workflow. The baseline and monitor surveys are migrated and conventionally differenced. Recall conventional seismic difference models in Figure 3.4 that shows ambiguity in reflection amplitudes. In order to interpret differenced models reflection shapes from original models are mapped to reflection shapes of differenced models. This is an inefficient method for the time-lapse studies, because it introduces ambiguity as seismic amplitude provides limited fidelity.

4.2.2 Cross-correlation differencing (CCD)

As the conventional seismic differencing is ambiguous I consider developing an algorithm that will employ cross-correlation to locate and through Gaussian filter eliminate all similar events. This means that only difference in fluid flow will be highlighted. I name this algorithm the cross-correlation differencing (CCD). CCD is implemented entirely in the time domain and as such, it is the most natural of the methods that I will present to those familiar with the conventional differencing.

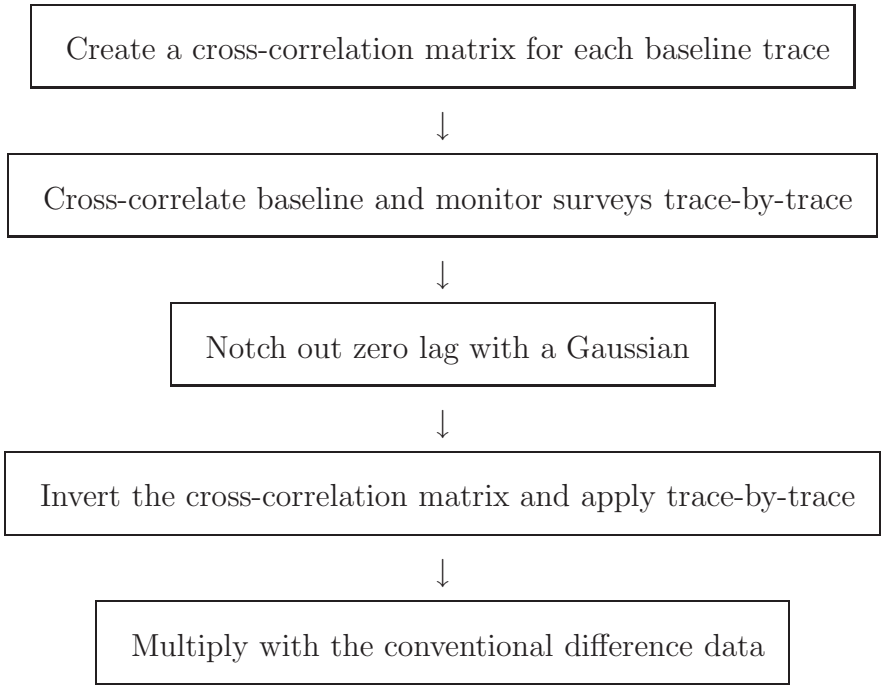


Table 4.2: Workflow V: CCD workflow starts by creating a cross-correlation matrix for each baseline trace. Then, the baseline and monitor surveys are cross-correlated trace-by-trace. The zero lag is notched out using the Gaussian filter in cross-correlation domain. The resulting matrix is then inverted to the time domain and scalar multiplied by conventional difference.

The algorithm takes in the time domain baseline and monitor surveys and then creates a cross-correlation matrix for each baseline trace. The cross-correlation matrix is trace-by-trace multiplied by the monitor survey to calculate the cross-correlation. The cross-correlation data is used to locate zero lag that contains information of similarities between the two and delete it. The deletion is done through a multiplication by the time domain Gaussian filter. Then, the cross-correlation operation is reversed. To complete the CCD algorithm, I use "the old processors trick": I multiply the non-conventional differencing (CCD) result by the conventional differencing (CD) result. If this step is skipped, the CCD differencing still hides dissimilarities. This is because the seismic amplitude of

similar events is not zero (or very small). The CD produces low amplitude values at both similar and dissimilar events. The CCD difference produces models of high amplitudes at dissimilar events and low amplitude values at similar. Now, multiplying these will give high amplitude values at dissimilar events and very small or nearly zero amplitude values at similar events. This means applying "the processors trick" will attenuate events of high energy and gain events of low energy. Hence, the end result will only highlight difference of two time-lapse steps.

Convolution and cross-correlation share similar characteristics, hence I refer to convolution theory and modify it to accommodate for cross-correlation matrix generation. The convolution is defined as:

$$n(t) \equiv a(t) \bullet h(t), \tag{4.2}$$

where $\bullet$ stands for the convolution operation and $n(t)$, $a(t)$ and $h(t)$ denote convolution of two signals, signal one and signal two, respectively. In mathematical terms, convolution is an operation on two functions, $a(t)$ and $h(t)$, producing the third function, $n(t)$, that can be described as manipulated version of one of the original functions. Convolution honors commutativity, associativity and distributivity (Margrave, 2008). In geophysical terms, convolution is an operation acting on two signals, such that one is viewed as the filter to the other. In practice, convolution models filtering of seismic energy by various rock layers in the Earth (Schlumberger, 2011).

Now, the convolution stated in equation (4.2) can be further examined mathematically through a definition of the convolutional integral (Margrave, 1998):

$$n(t) = \int_{-\infty}^{\infty} a(t-\tau)h(\tau)d\tau, \tag{4.3}$$

where $n(t)$, $a(t-\tau)$ and $h(\tau)$ denote filtered output, filter impulse response and input signal, respectively. Equation (4.3) can be rewritten in terms of matrix operations according

to (Margrave, 1998):

$$\mathbf{n} = \mathbf{A}\,\mathbf{h}, \tag{4.4}$$

where $\mathbf{A}$ is a convolution matrix and $\mathbf{n}$ and $\mathbf{h}$ are column vectors. If expanded equation (4.4) yields (Margrave, 1998):

$$\begin{bmatrix} \vdots \\ n_0 \\ n_1 \\ n_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & a_0 & a_{-1} & a_{-2} & \vdots \\ \vdots & a_1 & a_0 & a_{-1} & \vdots \\ \vdots & a_2 & a_1 & a_0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ h_0 \\ h_1 \\ h_2 \\ \vdots \end{bmatrix}. \tag{4.5}$$

Equation (4.5) highlights matrix A to have constant entries at each descending diagonal from left to right. In mathematical terms, this matrix is known as Toeplitz matrix, named after Otto Toeplitz, a German mathematician working on functional analysis (Bini, 1995). Common applications of Toeplitz matrices include the numerical solution of some differential and integral equations, the computation of splines, time series analysis, Markov chains, signal and image processing (Bini, 1995). In geophysical terms, this matrix structure is known as the convolution matrix (Innanen, 2010). It is created by populating each row by a filter with zero time shifted to the diagonal. Since convolution is briefly reviewed, it gives basis for analysis and manipulation.

The cross-correlation operation is defined as (Lines and Newrick, 2004) :

$$c(t) \equiv m(t) \otimes b(t), \tag{4.6}$$

where $\otimes$ stands for cross-correlation operation and $c(t)$, $m(t)$ and $b(t)$ denote cross-correlation of two signals, signal one and signal two, respectively. Recall the basic definition of cross-correlation to be a measure of similarity of two waveforms.

Its definition in equation (4.6) can be expressed as the cross-correlation integral given by:

$$c(t) = \int_{-\infty}^{\infty} a(t+\tau)b(\tau)d\tau, \tag{4.7}$$

where $c(t)$, $a(t+\tau)$, $b(\tau)$ denote filtered output, reverse filter impulse response and impulse signal, respectively. Now, the cross-correlation in equation (4.7) can be related to convolution in equation (4.3) by time reversing filter impulse and populating matrices as:

$$\mathbf{c} = \mathbf{A}\,\mathbf{b}, \tag{4.8}$$

that is

$$\begin{bmatrix} \vdots \\ c_0 \\ c_1 \\ c_2 \\ \vdots \end{bmatrix} = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots \\ \vdots & a_0 & a_1 & a_2 & \vdots \\ \vdots & a_{-1} & a_0 & a_1 & \vdots \\ \vdots & a_{-2} & a_{-1} & a_0 & \vdots \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{bmatrix} \begin{bmatrix} \vdots \\ b_0 \\ b_1 \\ b_2 \\ \vdots \end{bmatrix}, \tag{4.9}$$

where $\mathbf{c}$ and $\mathbf{b}$ denote column vectors and $\mathbf{A}$ denotes the cross-correlation matrix. Generating matrix A is my goal to start the execution of the CCD algorithm.

The cross-correlation matrix is multiplied by the monitor survey to calculate the cross-correlation of the baseline and monitor surveys. The above statement can be described in mathematical terms through a matrix operation as:

$$\mathbf{x}_{corr} = \mathbf{A}\,\mathbf{m}, \tag{4.10}$$

where $\mathbf{A}$ and $\mathbf{m}$ are the cross-correlation matrix for one baseline trace, one trace from the monitor survey. Equation (4.10) groups all similarities around zero lag and all dissimilarities elsewhere.

Then the time domain Gaussian filter is created. I employ the time domain Gaussian

function defined as (Margrave, 2008):

$$g(t) = e^{-\sigma^2 t^2}, \tag{4.11}$$

where $g(t)$, σ and t denote Gaussian filter, Gaussian width and time, respectively.

The cross-correlation matrix in time is scalar multiplied by the Gaussian filter defined in equation (4.11) as

$$\mathbf{f} = (1 - \mathbf{g}) \odot \mathbf{x}_{corr}, \tag{4.12}$$

where $\odot$ indicates scalar multiplication between the elements of vectors. Equation (4.12) deletes zero lag, hence leaves only dissimilarities of the two surveys.

Now, that the data is cross-correlated and filtered, the cross-correlation matrix $\mathbf{A}$ is employed to provide inverse cross-correlation operator, $\mathbf{A^{-1}}$. This operator reverses the cross-correlation operation.

The filtered cross-correlation is then multiplied by the inverse cross-correlation matrix:

$$\mathbf{r} = \mathbf{A^{-1}f}. \tag{4.13}$$

The result of equation (4.13) is multiplied by the conventional difference:

$$\mathbf{d^\star} = \mathbf{r} \odot [\mathbf{m} - \mathbf{b}]. \tag{4.14}$$

Equation (4.14) is the final result of the CCD algorithm. At this point all differences are highlighted and almost all similarities are eliminated.

As a single analytic process, CCD is

$$\mathbf{d^\star} = \mathbf{A}^{-1} \left[(1 - \mathbf{g}) \odot [\mathbf{A}\ [\mathbf{m} \odot [\mathbf{m} - \mathbf{b}]]]\right]. \tag{4.15}$$

Note, though much of the computational effort in CCD is restricted to scalar multiplication, convolution and then de-cross-correlation by $\mathbf{A}$ and $\mathbf{A}^{-1}$ which can be quite expensive.

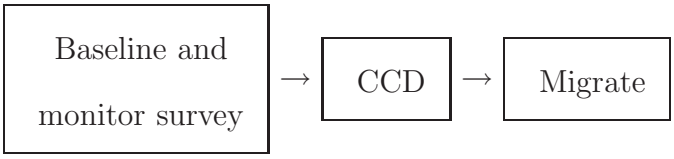


Table 4.3: Workflow VI: General differencing. To further study seismic data after the baseline and monitor surveys are CCD differenced, the result can be migrated and compared to the CD result.

The result of CCD algorithm can be used for further analysis. In Table 4.3 I propose the result of CCD differencing to be migrated and compared to the CD result.

4.2.3 Computational cost of CCD

To measure the computational efficiency of an algorithm, one may want to see how much time the algorithm requires to solve a problem. The above can be done by measuring clock running time, however, such analysis may not be accurate. The clock running time can be affected by various factors not related to the computational performance of the algorithm itself. These could be the machine power (memory available), the data set size; if run on server, how many users are running other jobs, how demanding these jobs are, etc. Thus the time required to solve a problem is measured as a function and expressed in the big "O" notation. In computer science, the big "O" notation is used to classify algorithms by how they respond based on the operations that take place and the input size. Generally, the big "O" notation defines operations according to their growth rates. For example, the Fourier transform will take $O(NlogN)$ operations to compute.

I will use the big "O" notation to analyze the cost of the CCD algorithm by analyzing the cost of each major operation that takes place and express it as a function growth. I assume the baseline and monitor surveys to be two $M \times N$ matrices.

operation	cost
build cross-correlation matrix	$O(MN)$
multiplication	$O(M^2N^2)$
build Gaussian filter	$O(MN)$
build inverse cross-correlation operator	$O(M^3N^3)$
multiplication	$O(MN)$
migration	$O(MNlog(MN))$

Table 4.4: Workflow VII: Computational cost of CCD workflow in time domain. It takes $O(M^3N^3 + M^2N^2 + 3MN + MNlog(MN))$ operations to complete the workflow. The process is successful, but costly.

CCD's computational cost is outlined in Table 4.4. Although, the algorithm is successful, it is computationally costly as it takes $O(M^3N^3+M^2N^2+3MN+MNlog(MN))$ operations to complete. The most expensive operation seems to be the creation of the inverse cross-correlation operator. This operation is considered for optimization, as I anticipate if substituted by Fourier transform, it can reduce cost significantly. Hence, the CCD algorithm motivates modification.

4.2.4 Pseudo cross-correlation differencing (PCCD)

Based on my observation of CCD's computational cost, I implement the pseudo cross-correlation differencing (PCCD) algorithm as the equivalence of the CCD algorithm in the frequency domain. Since the discrete Fourier transform can be written in terms of matrix multiplication, then the matrix inversion in the time domain is equivalent to the inverse discrete Fourier transform in the frequency domain. Hence, the costly matrix inversion is replaced by the Fourier transformation.

The algorithm is named after the main operation taking place, that is cross-correlation.

Since the cross-correlation in the time domain is equivalent to multiplication by complex conjugate in frequency domain, I name the multiplication pseudo cross-correlation to accommodate for easy comparison between the two algorithms.

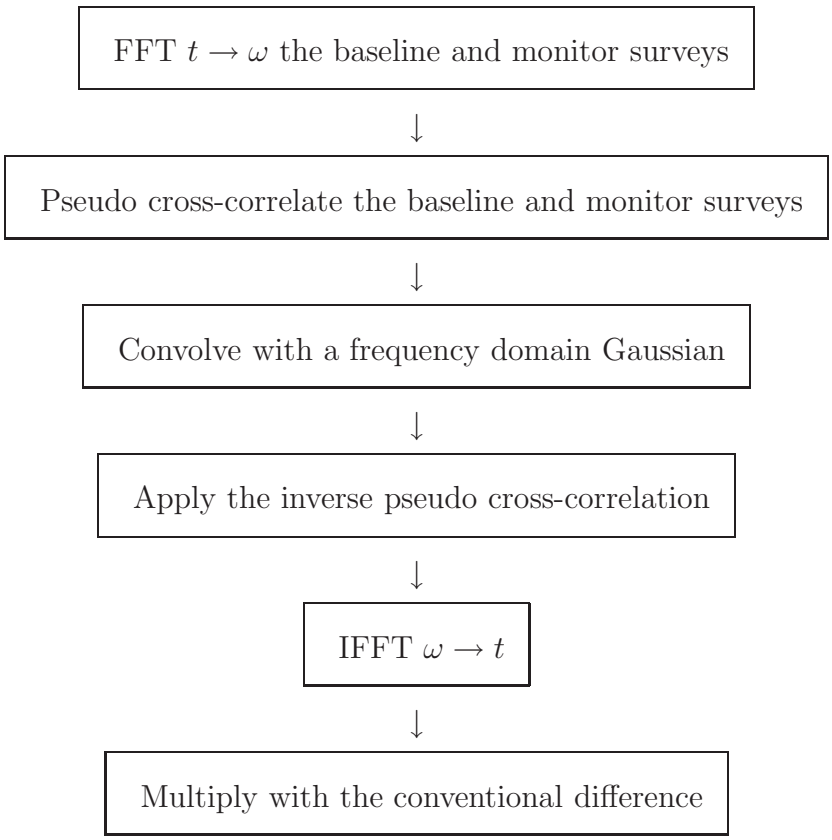


Table 4.5: Workflow VIII: The PCCD algorithm performed in the frequency domain. The algorithm takes in the baseline and monitor surveys, forward Fourier transforms (FFT) them to the frequency domain, calculates the pseudo cross-correlation and convolves the result with a frequency Gaussian filter. The output is then multiplied by the inverse pseudo cross-correlation and inverse Fourier transformed (IFFT) from the frequency to time domain. The final operation is filtered by the conventional difference through a matrix multiplication.

Table 4.5 states the workflow of the PCCD algorithm. The algorithm takes in the baseline and monitor surveys in the time domain, then through forward Fourier transform

takes them to the frequency domain. The frequency baseline and monitor surveys are pseudo cross-correlated and convolved with a frequency domain Gaussian. Then, the inverse pseudo cross-correlation is applied to the result and inverse Fourier transformed to the time domain. The final output is filtered by the conventional difference through matrix multiplication.

I now present detailed mathematics of the PCCD algorithm. The user passes the baseline and monitor surveys in the time domain to the PCCD algorithm. The baseline survey, $b(t)$, and monitor survey, $m(t)$, are FFTed to the frequency domain as:

$$B(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} b(t) e^{iwt} dt, \tag{4.16}$$

$$M(w) = \frac{1}{2\pi} \int_{-\infty}^{\infty} m(t) e^{iwt} dt, \tag{4.17}$$

where $B(t)$, $M(t)$ and w denote baseline and monitor surveys and frequency in the frequency domain, respectively. $B(t)$ and $M(t)$ are further decomposed into their phase and amplitude spectra as:

$$B(w) = A_b(w) e^{i\phi_b(w)}, \tag{4.18}$$

$$M(w) = A_m(w) e^{i\phi_m(w)}, \tag{4.19}$$

where $A_b(w)$, $A_m(w)$, are the amplitude spectra of the baseline and monitor surveys, respectively, and $\phi_b(w)$ and $\phi_m(w)$ are the phase spectra of the baseline and monitor surveys, respectively.

$B(t)$ and $M(t)$ are cross-correlated as:

$$P_{cc} = A_m(w) e^{i(\phi_m(w) - \phi_b(w))}. \tag{4.20}$$

Note that the amplitude spectrum of baseline survey is omitted in equation (4.20). This is because the attention is only focused on seismic phase spectrum and its response on differencing. The deletion of all data that are similar between the baseline and monitor

surveys, requires phase information from both spectra and amplitude information only from the monitor spectrum.

Then, the Gaussian filter in frequency domain is created as:

$$G(w) = \sqrt{\frac{\pi}{\sigma}} e^{(-\frac{\pi w}{\sigma})}, \tag{4.21}$$

where $G(w)$ and σ denote Gaussian filter in the frequency domain and Gaussian width, respectively.

The frequency domain Gaussian filter and pseudo cross-correlation are convolved:

$$S(w) = \int_{-\infty}^{\infty} G(w^{\star}) P_{cc}(w - w^{\star}) dw^{\star}, \tag{4.22}$$

where w^{*} denote a dummy frequency variable.

The inverse pseudo cross-correlation (IPCC) is computed from the following equation:

$$R(w) = e^{i\phi_m(w)} S(w). \tag{4.23}$$

In order to reverse the pseudo cross-correlation I perform the inverse pseudo cross-correlation (IPCC). To compute IPCC I only use the definition of the phase of the monitor survey to restore the phase of the filtered data. This is because the phase spectrum only depends on seismic traveltime, and as such provides confidence in results (Ferguson, 2010). The amplitude spectrum, not used in this calculation, can be ambiguous. Any variations in shot coupling, geophone coupling or shot strength seem to reflect most strongly in the amplitude rather than in the phase spectrum (Ferguson, 2010). Seismic noise sources along with model estimation and imaging errors appear to distort amplitudes much more than phases (Ferguson, 2010). This observation is tested by Ferguson (2010) in his unpublished Seismic Imaging class notes through an experiment. Ferguson takes an image of two guitars and computes its amplitude and phase to test how amplitude and phase affect restoration of an image.

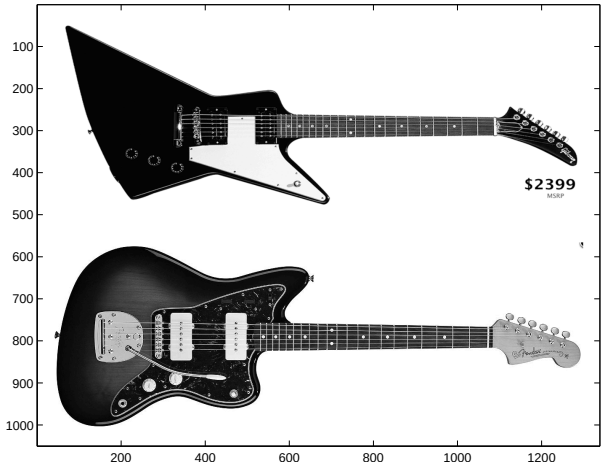


Figure 4.1: Two guitars manufactured by different companies are presented: Gibson Explorer at the top and Fender Jazzmaster at the bottom.

These two guitars are presented in Figure 4.1: Gibson Explorer guitar at the top and Fender Jazzmaster guitar at the bottom. They are manufactured by different companies and will serve for the experiment. Firstly, employing MATLAB functions amplitude and phase spectra are extracted to yield:

$$B(w) = A_1(w)e^{i\phi_1(w)}, \tag{4.24}$$

$$M(w) = A_2(w)e^{i\phi_2(w)}, \tag{4.25}$$

where A_1, A_2, ϕ_1 and ϕ_2 stand for the amplitude spectrum of Gibson Explorer, the amplitude spectrum of Fender Jazzmaster, the phase spectrum of Gibson Explorer and the phase spectrum of Fender Jazzmaster, respectively. Now, a third image is created employing the amplitude spectrum of Fender and the phase spectrum of Gibson as:

$$B(w) = A_2(w)e^{i\phi_1(w)}. \tag{4.26}$$

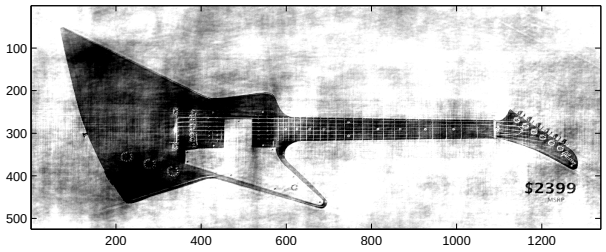


Figure 4.2: Restored image using amplitude spectrum of Fender Jazzmaster image and the phase spectrum of Gibson Explorer image.

The resulting image in Figure 4.2 resembles much like the Gibson, and the effect of the Fender amplitude shows as blurring of the image. This experiment suggests that amplitude spectrum is less informative when compared to phase spectrum, hence proves the choice of using just monitor survey's phase spectrum in computing IPCC sufficient.

Now, the computed result in equation (4.23) is inverse Fourier transformed from the frequency to time domain:

$$r(t^{\star}) = \int_{-\infty}^{\infty} R(w) e^{iwt^{\star}} dw. \tag{4.27}$$

The computed result multiplies the conventional difference as:

$$\mathbf{D}^{\star} = \mathbf{r} \odot [\mathbf{M} - \mathbf{B}]. \tag{4.28}$$

As in CCD, the final operation is multiplication by the conventional difference. Recall, this is the "the old processors trick" that will attenuate events of high energy (similarities) and gain events of low energy (dissimilarities). Hence, the final result will only focus difference between two time-lapse steps.

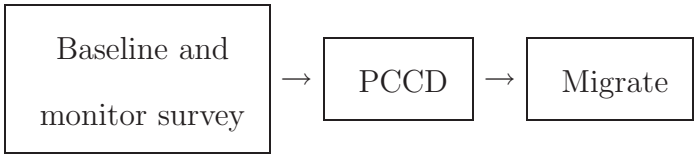


Table 4.6: Workflow IX: To further study seismic data after the baseline and monitor surveys are PCCD differenced, the result can be migrated and compared to the CD and CCD results.

The result of PCCD algorithm can be used for further analysis. In Table 4.6 I propose the result of PCCD differencing to be migrated and compared to the CD and CCD results.

4.2.5 Computational cost of PCCD

I estimate the computational cost of the PCCD algorithm using the big "O" notation again. Recall, I assume the baseline and monitor surveys are two $M \times N$ matrices.

operation	cost
fast Fourier transform (FFT) X 2	$O(2MNlog(MN))$
pseudo cross-correlation	$O(M^2N^2)$
build Gaussian filter	$O(MN)$
convolution	$O(MNlogMN)$
build inverse pseudo cross-correlation operator	$O(MN)$
inverse fast Fourier transform (IFFT)	$O(MNlogMN)$
migration	$O(MNlog(MN))$

Table 4.7: Workflow X: Computational cost of the PCCD algorithm. It takes $O(M^2N^2 + 2MN + 5MNlogMN)$ operations to complete the workflow. The process is successful and much cheaper if executed in the frequency domain.

PCCD's computational cost is outlined in Table 4.4. The most expensive calculation is convolution with $O(M^2N^2)$ operations, and the least expensive calculation is fast for-

ward/inverse Fourier transformation with $O(MNlongMN)$ operations. PCCD algorithm is successful and computationally much cheaper, then CCD algorithm:

$$O(M^2N^2+2MN+5MNlogMN) << O(M^3N^3+M^2N^2+3MN+MNlog(MN)). \quad (4.29)$$

Simplifying equation (4.29) to:

$$O(4MNlogMN) << O(M^3N^3 + M^2N^2 + MN) \quad (4.30)$$

significant decrease in computational cost from CCD to PCCD is noted.

4.2.6 Conventional imaging condition difference (CICD)

CICD method is based on the modification of pre-stack depth migration (PSDM) and integration of conventional differencing at the imaging condition. Pre-stack depth imaging requires that both the recorded wavefield on the Earth's surface and the seismic source impulse be extrapolated downward into the Earth, employing the velocity model. I name this algorithm conventional imaging condition difference (CICD). In CICD approach the model source is replaced by the baseline survey and the surface recording by the monitor survey. The reference wavefield of the monitor survey will act to find all differences and image them to the output. This simple modification allows for fluid flow to be the only reflection on seismic models.

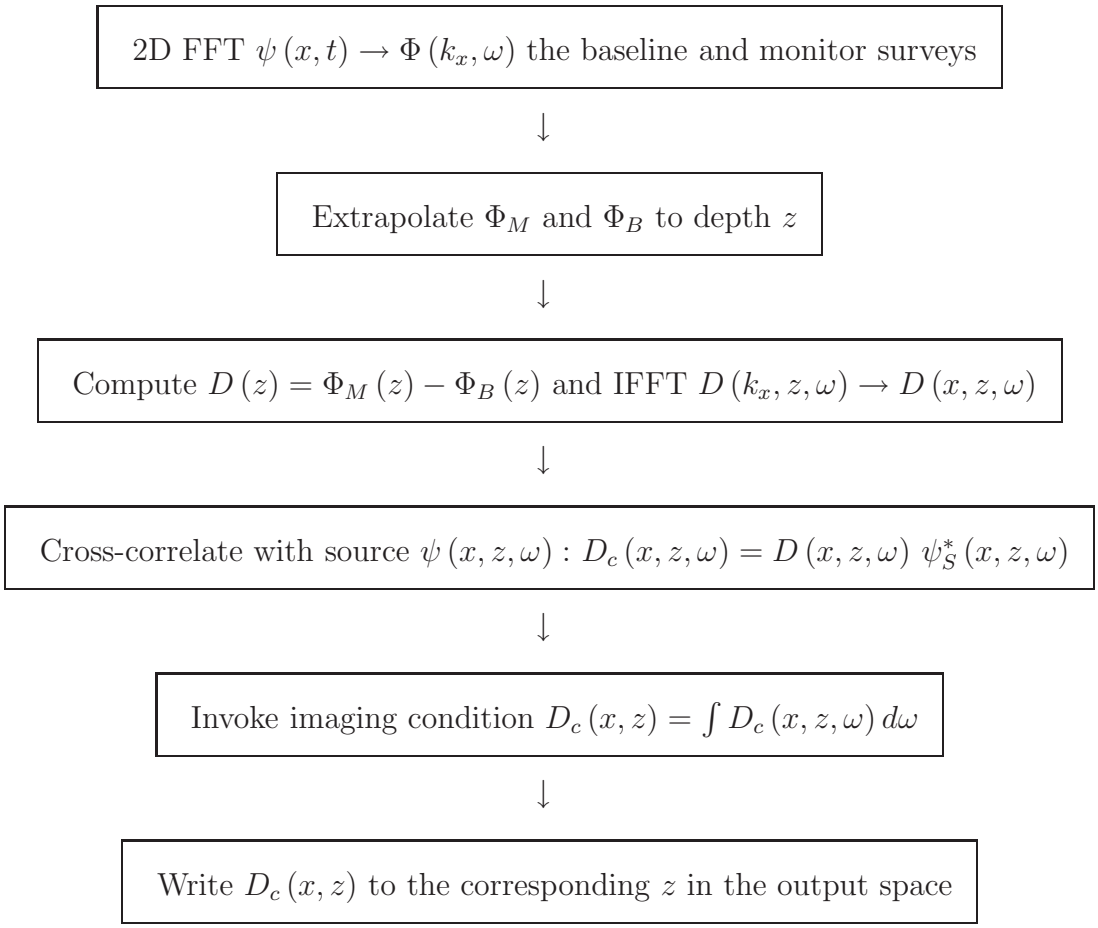


Table 4.8: Workflow XI: The CICD algorithm is outlined as a workflow. The inputs are the time domain baseline and monitor surveys that are 2D FFTed and its corresponding wavefields extrapolated. The two are conventionally differenced using matrix subtraction and its output is IFFTed to the space domain. Then, the result is cross-correlated by the source (baseline) wavefield. Now, the imaging condition is invoked and final output is written is the space domain.

In Table 4.8 I present a summary of the CICD workflow. The algorithm starts by taking a 2D FFT of the baseline and monitor surveys. The baseline and monitor survey wavefields are extrapolated, conventionally differenced and IFFTed. The result is cross-correlated by the baseline (source) wavefield and imaging condition is invoked. The final

output is written in the space domain for the corresponding depth level z. The CICD is a recursive algorithm and will repeat for all depth levels.

I present here mathematics of the CICD algorithm. The algorithm's inputs are $\psi_B(x,t)$ and $\psi_M(x,t)$, the baseline and monitor surveys in the time domain, respectively. They are 2D FFT to:

$$\Phi_B(k_x,z,\omega)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\psi_B(x,z,t)e^{2\pi i(xk_x-\omega t)}dxdt, \tag{4.31}$$

$$\Phi_M(k_x,z,\omega)=\int_{-\infty}^{\infty}\int_{-\infty}^{\infty}\psi_M(x,z,t)e^{2\pi i(xk_x-\omega t)}dxdt. \tag{4.32}$$

The wavefield extrapolation is done on the 2D FFTed wavenumber domain baseline and monitor surveys shown in equations (4.31) and (4.32) as:

$$\Phi_B^{\star}(k_x,z,\omega)=\Phi_B(k_x,z,\omega)e^{2\pi ik_z\Delta z}, \tag{4.33}$$

$$\Phi_M^{\star}(k_x,z,\omega)=\Phi_M(k_x,z,\omega)e^{2\pi ik_z\Delta z}. \tag{4.34}$$

The CD is performed between the results of equations (4.33) and (4.34):

$$D(k_x,z,\omega)=\Phi_M^{\star}(k_x,z,\omega)-\Phi_B^{\star}(k_x,z,\omega). \tag{4.35}$$

The result of equation (4.35) is IFFTed as:

$$D(x,z,\omega)=\int_{-\infty}^{\infty}D(k_x,z,\omega)e^{2\pi ik_z\Delta z}dk_x. \tag{4.36}$$

To calculate the cross-correlation, I multiply the result of equation (4.36) by the conjugate of the extrapolated wavefield of the baseline (source) survey, $\psi_S^{\star}(k_x,z,\omega)$, as:

$$D_c(x,z)=D(x,z,\omega)\psi_B^{\star}(x,z,\omega). \tag{4.37}$$

Then, I invoke the imaging condition as:

$$D_c(x,z)=\int_{-\infty}^{\infty}D_c(x,z,\omega)d\omega. \tag{4.38}$$

The solution of equation (4.38) is written out for every depth level z.

Note seismic reflectivity can then be estimated by the ratio between the backward-extrapolated monitor wavefield and the forward-extrapolated baseline wavefield immediately above a certain depth level, where the traveltime becomes zero. Estimation of reflectivity from wavefields at a certain depth level is called the imaging condition (Claerbout, 1971) stated by equation(4.38). The conventional difference is computed in the frequency domain at the imaging condition for every depth level using matrix subtraction.

4.2.7 Computational cost of CICD

The computational cost of CICD is estimated as well. Recall the baseline and monitor surveys to be two $M \times N$ matrices.

operation	cost
PSDM	$O(MNlog(MN))$
CD	$O(MN)$

Table 4.9: Workflow XII: Computational cost of the CICD workflow. It takes $O(MN + MNlog(MN))$ operations to complete the workflow. The process is successful.

Table 4.9 is a summary of computational cost for CICD. It takes the same number of operations as execution of the conventional differencing and migration, however, eliminates user dependence.

4.2.8 Imaging condition differencing (ICD)

Having successfully tested combination of PSDM and CD producing the CICD algorithm, I proceed to combine PSDM and PCCD.

As I mention above the cross-correlation imaging condition of seismic depth migration employs a source wavefield model to identify reflection amplitudes on the input data set. This identification happens through cross-correlation, namely a process that relies on good phase fidelity in both the model and the data. The reflection amplitude in the data whose corresponding phase corresponds to that of source amplitude is mapped to zero lag. Data at zero lag are mapped to the image space for all depth grid levels, the migrated seismic image gives the output.

The idea is to discard all matching data at the imaging condition. In application to time-lapse analysis, this idea seems reasonable as it is sensitive to phase that is recorded most reliably in the seismic method. The reference wavefield will act to find all similar energy in the monitor survey and then map that energy to zero lag. All similar events are eliminated and the difference highlights, hence the fluid flow, is captured. Therefore, the ICD algorithm is based on the same idea as the CICD algorithm and the only difference is the equation (4.35) where CD is computed will now be replaced by the PCCD algorithm outlined in Section 4.2.6.

2D FFT $\psi(x,t) \rightarrow \Phi(k_x,\omega)$ the baseline and monitor surveys

$\downarrow$

Extrapolate Φ_M and Φ_B to depth z

$\downarrow$

Compute PCCD of $\Phi_M(z)$ and $\Phi_B(z)$ and IFFT $D(k_x,z,\omega) \rightarrow D(x,z,\omega)$

$\downarrow$

Cross-correlate with source $\psi(x,z,\omega) : D_{pccd}(x,z,\omega) = D(x,z,\omega)\ \psi_S^*(x,z,\omega)$

$\downarrow$

Invoke imaging condition $D_{pccd}(x,z) = \int D_{pccd}(x,z,\omega)\, dw$

$\downarrow$

Write $D_{pccd}(x,z)$ to the corresponding z in the output space

Table 4.10: Workflow XIII: The ICD algorithm is outlined here. The algorithm's inputs are the time domain baseline and monitor surveys. They are 2D FFTed and their corresponding wavefields are extrapolated. The result is PCCD differenced and its output is IFFTed to the space domain. Then, the cross-correlation by the source (baseline) wavefield is performed on the result. After invoking the imaging condition the final output is written in the space domain.

In Table 4.10 I present a summary of the ICD workflow. ICD takes in the time domain baseline and monitor surveys and computes their 2D FFT. The baseline and monitor survey wavefields are extrapolated, PCCD differenced and IFFTed. Then the result is cross-correlated by the baseline (source) wavefield and imaging condition is invoked. The

final output is written in the space domain for the corresponding depth level z. The ICD as CICD is also a recursive algorithm and will repeat for all depth levels.

4.2.9 Computational cost of ICD

Employing the big "O" notation I estimate the computational cost of the ICD algorithm.

operation	cost
PSDM	$O(MNlog(MN))$
PCCD	$O(M^2N^2 + 2MN + 5MNlogMN)$

Table 4.11: Workflow XIV: Computational cost of ICD is the sum of PSDM and PCCD. It takes $O(M^2N^2 + 2MN + 6MNlogMN)$ operations to ICD data.

Table 4.11 shows the computational cost of the ICD algorithm based on the two major operations PSDM and PCCD. ICD proves to be of optimal computation time and user independent. Hence, it can be used as an alternative tool in seismic difference analysis of time-lapse surveys.

4.3 Example I

4.3.1 Velocity models and synthetics

One of the data sets used is the EAGE/SEG salt velocity model (Aminzadeh et al., 1996). The model consists of complex salt structures with large velocity contrasts across the salt/sediment interface (Aminzadeh et al., 1996). This is a well known pre-stack depth migration (PSDM) testing data set publicly available, hence convenient to use in this work.

Note non-conventional difference amplitude spectra models are scaled to that of the conventional difference amplitude spectra. The scaling takes the maximum value

of both conventional and non-conventional difference spectra and defines a maximum value. Then, this value divides both conventional and non-conventional amplitude spectra. Hence, conventional and non-conventional models are now scaled for comparison. This process of scaling is kept consistent throughout the thesis work.

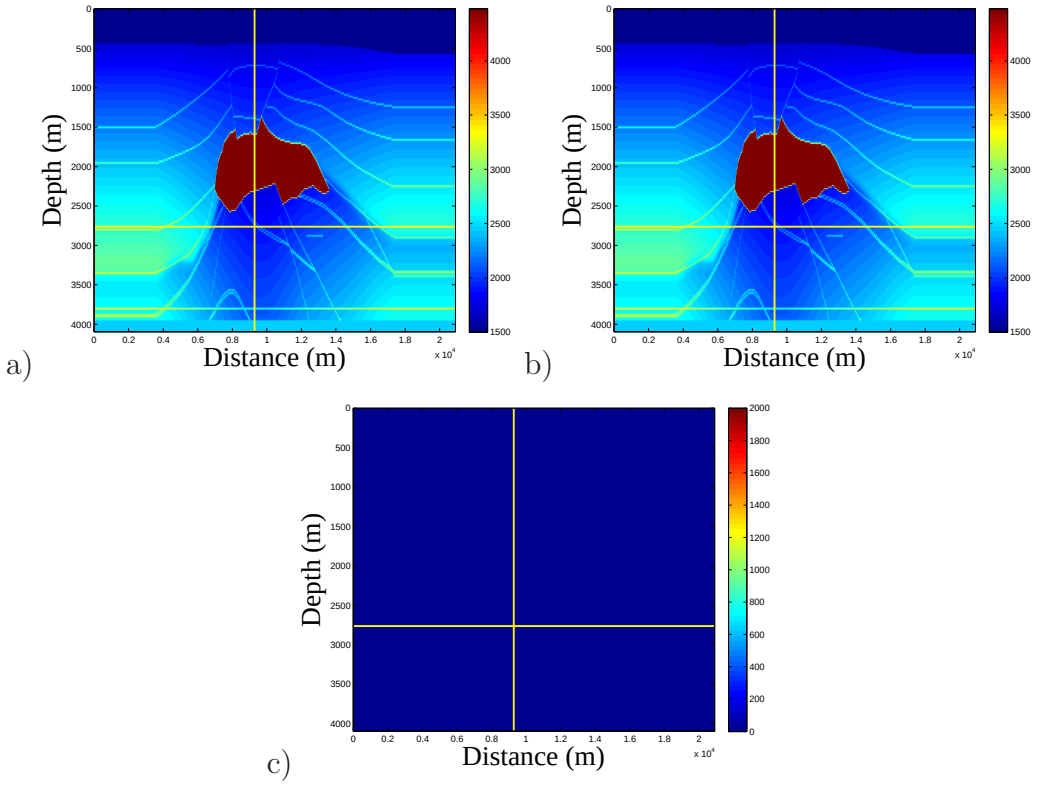


Figure 4.3: The EAGE/SEG salt velocity models: a) the original model, b) the author manipulated original model by inserting a small box in the sub-salt region to accommodate time-lapse analysis and c) the difference of gathers a) and b). The box will model change in the sub-salt region due to production. The color bar denotes velocity values in m/s.

In Figure 4.3 I present the EAGE/SEG velocity models. Figure 4.3(a) show the original EAGE/SEG model. Figure 4.3(b) shows the manipulated model. To accommodate time-lapse study and analysis I insert a small reflector in the sub-salt region of

the EAGE/SEG salt velocity model and assume it to mimic changes due to production. The box is of constant velocity and its location is indicated by the yellow cross arrows. I denote the original model baseline survey and the manipulated model monitor survey. Hence, I obtain two time-lapse steps. Figure 4.3(c) shows the conventional difference of Figures 4.3(a) and 4.3(b).

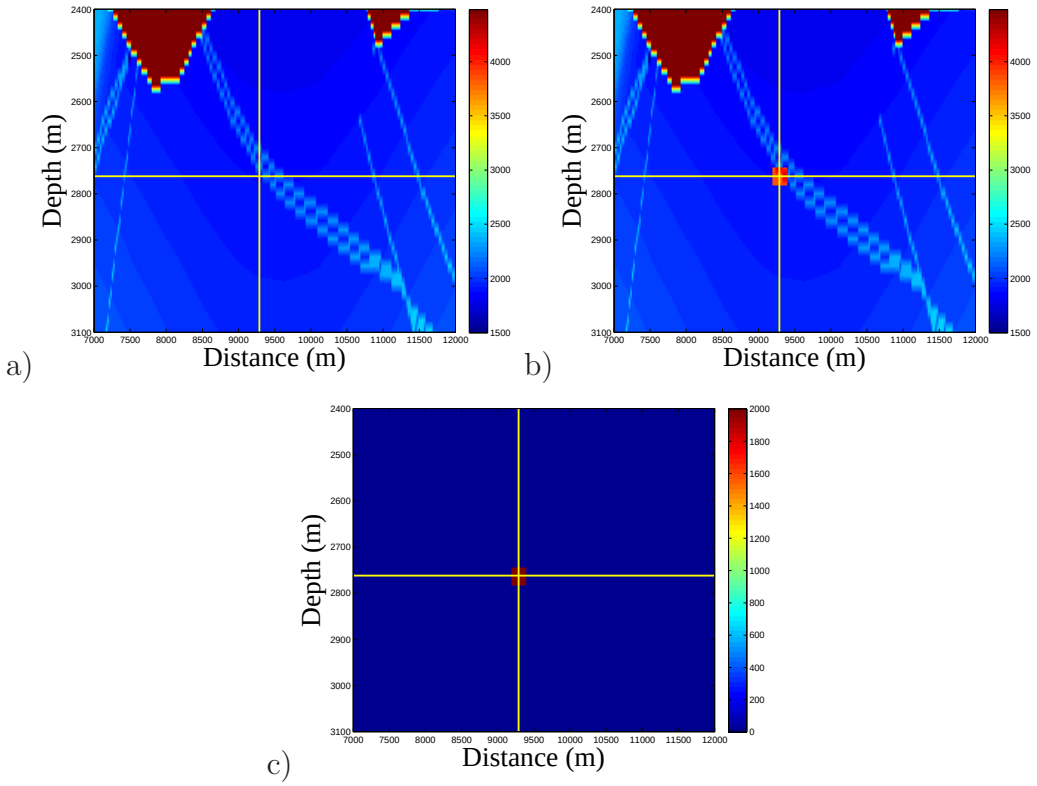


Figure 4.4: The EAGE/SEG salt zoomed in velocity models: a) the baseline survey, b) the monitor survey and c) the conventional difference of a) and b) models. Note the only difference between surveys is the inserted reflector. The color bar denotes velocity values in m/s.

Figure 4.4 is a zoomed in version of Figure 4.3. It focuses on the inserted reflector. Hence, the baseline to monitor survey difference is captured.

Now, using the velocity models synthetics are created employing $afd\_shotrec$, MAT-

LAB CREWES toolbox function created by Dr Gary Margrave.

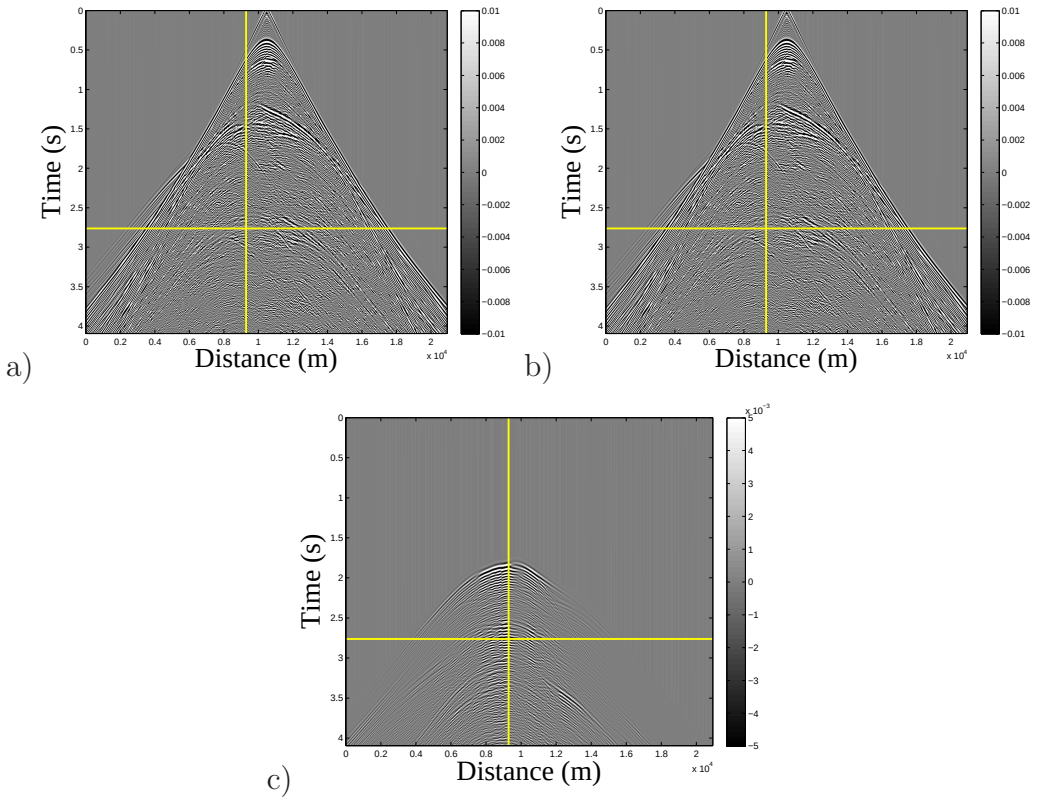


Figure 4.5: The EAGE/SEG salt shot gathers: a) the shot gather of the baseline survey, b) the shot gather of the monitor survey and c) the difference of gathers a) and b).

Figure 4.5 shows the shot gather of the baseline survey, the shot gather monitor survey and the difference of the two gathers.

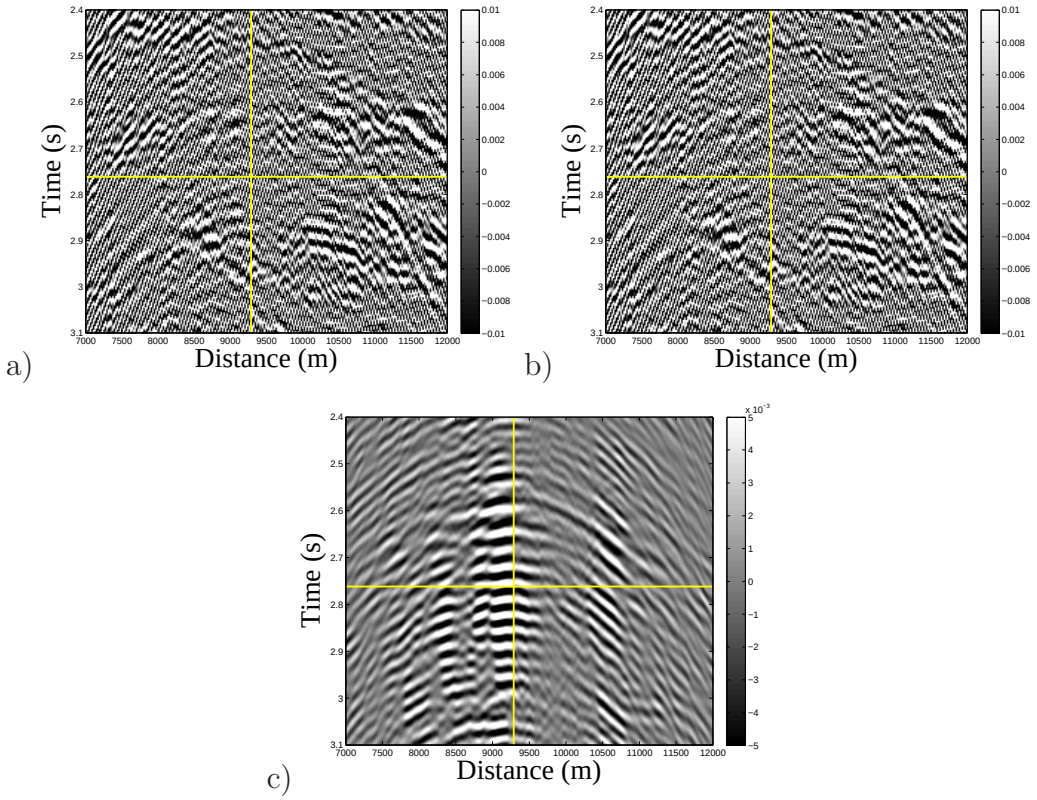


Figure 4.6: The zoomed in EAGE/SEG salt shot gathers: a) the shot gather of the baseline survey, b) the shot gather of the monitor survey and c) the difference of models a) and b).

Figure 4.6 shows the zoomed in models, focusing on the area around the inserted box. Observing closely Figure 4.6(c), it is not possible to determine the location of the box.

4.3.2 CD

The shot gathers are migrated employing a MATLAB function from the CREWES toolbox, *ss_salt_psdm_diff_script*. This function is based on the split-step Fourier migration (Stoffa et al., 1990) (explained in Chapter 3) and implemented by Dr Robert Ferguson.

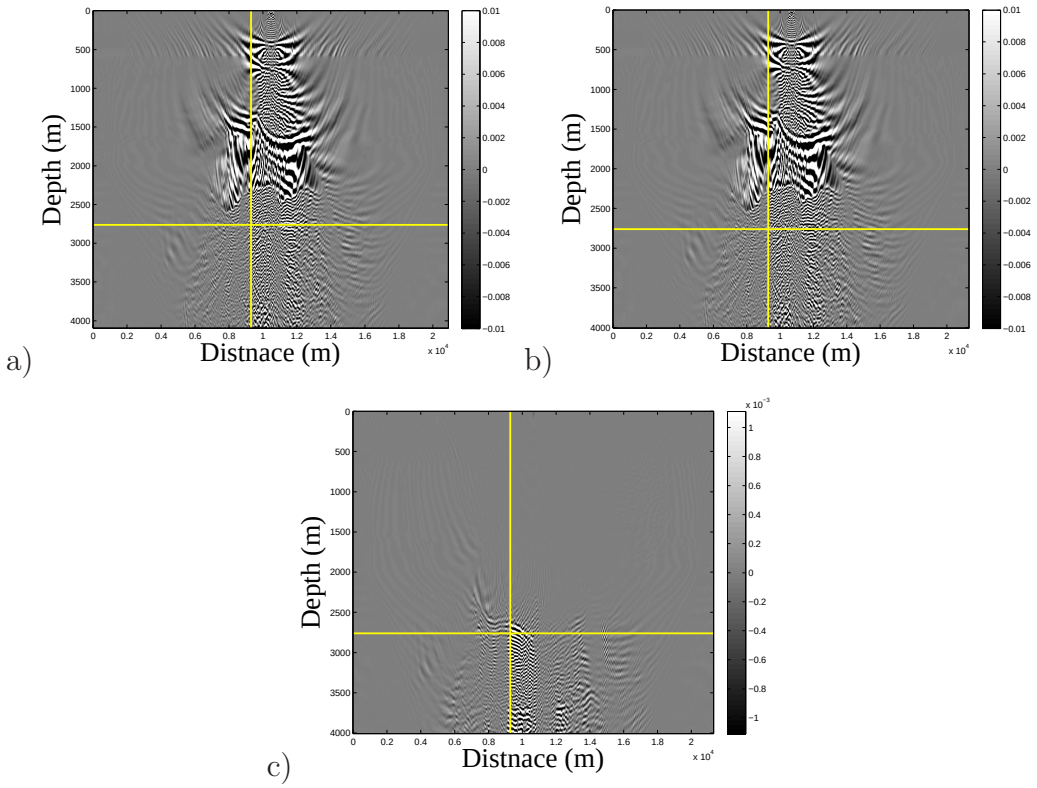


Figure 4.7: The EAGE/SEG salt shot gathers migrated: a) the migrated baseline shot gather, b) the migrated monitor shot gather and c) the difference of gathers a) and b).

Figure 4.7 shows the baseline survey migrated, the monitor survey migrated and the difference of the two.

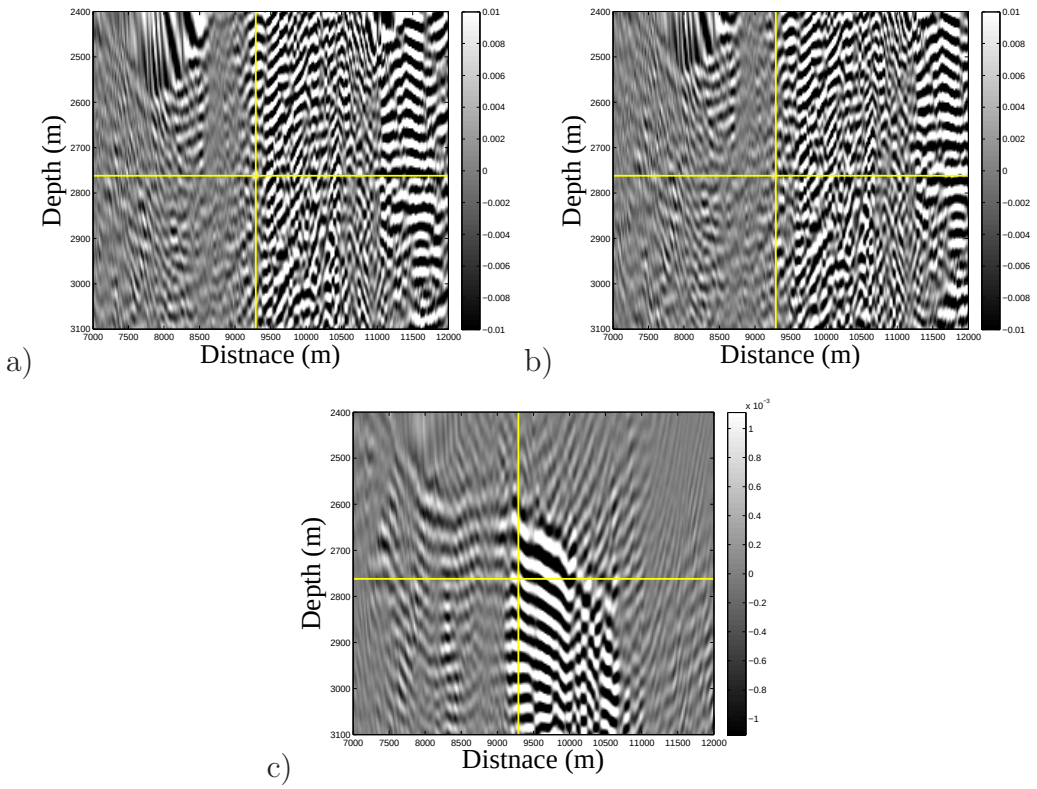
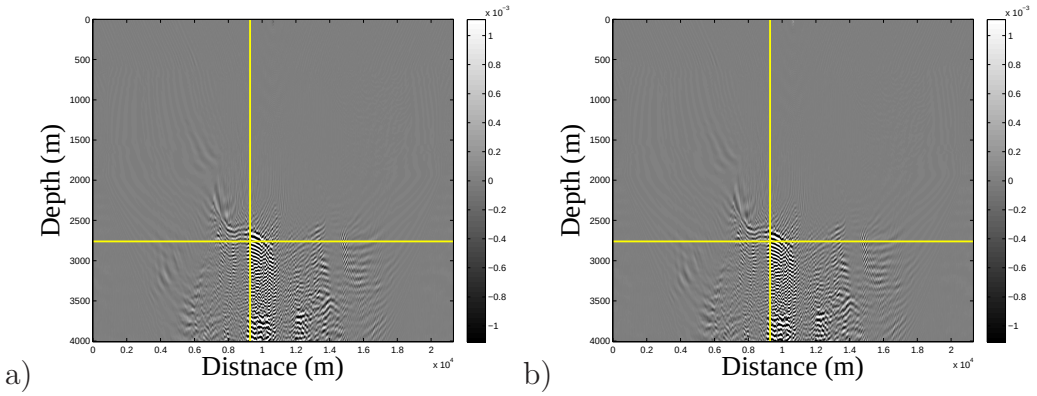


Figure 4.8: The zoomed in EAGE/SEG salt shot gathers migrated: a) the migrated baseline shot gather, b) the migrated monitor shot gather and c) the difference of a) and b).

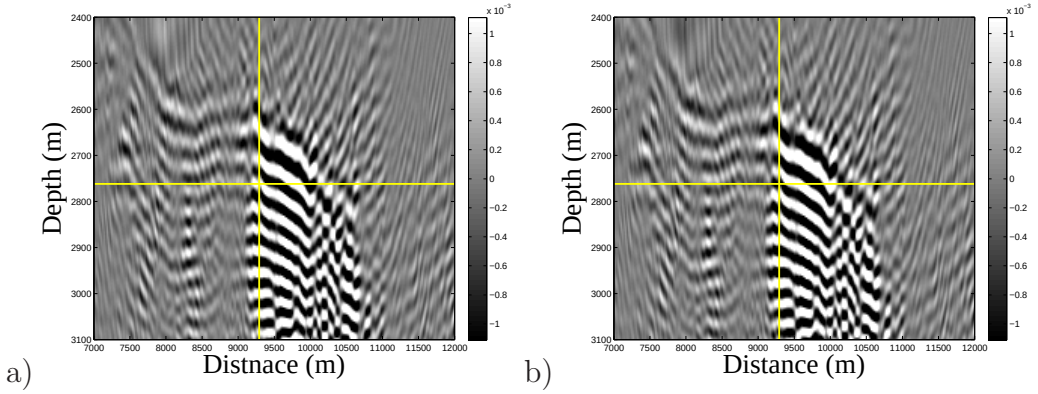
Figure 4.8 is a zoomed in version of Figure 4.7 around the area of interest. It is very difficult to determine the location of the inserted reflector if there were no cross arrows.



a) b)

Figure 4.9: The EAGE/SEG conventional difference models: a) the conventional difference of the migrated baseline and monitor surveys and b) the conventional difference of the baseline and monitor shot gathers migrated. As anticipated there is no difference between a) and b) except the machine precision as migration is linear.

Figure 4.9(a) captures the difference of migrated baseline and monitor surveys. Figure 4.9(b) captures the migration of difference of baseline and monitor surveys. As migration is linear there is no difference between Figures 4.9(a) and 4.9(b) except the machine precision.



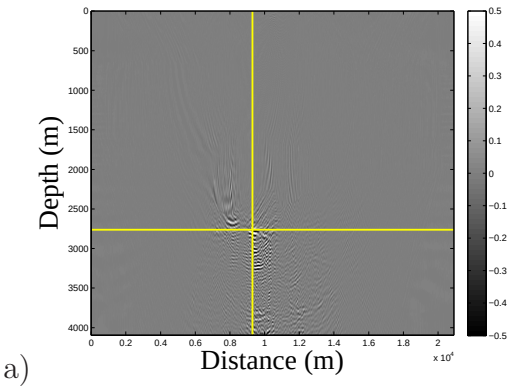
a) b)

Figure 4.10: The zoomed in EAGE/SEG conventional difference models: a) the conventional difference of the migrated baseline and monitor surveys and b) the conventional difference of the baseline and monitor shot gathers migrated. Again, note not much difference between a) and b) as migration is linear.

Figure 4.10 captures the zoomed in the models of Figure 4.9. Again, there is no significant difference, apart from machine precision, as expected between Figures 4.10(a) and 4.10(b).

4.3.3 CCD

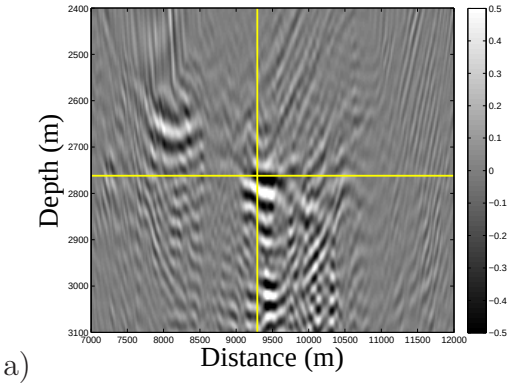
The baseline and monitor survey shot gathers are taken and filtered employing the cross-correlation method. Then, the filtered data is migrated using the split-step Fourier migration by the user.



a)

Figure 4.11: The CCD result.

Figure 4.11 captures the result of CCD and PSDM performed sequentially.



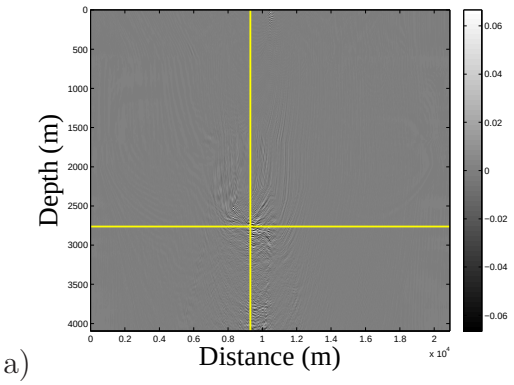
a)

Figure 4.12: The CCD result zoomed in.

Figure 4.12 is the zoomed in version of Figure 4.11. Although the yellow cross-arrows point to the location of the inserted reflector, its location is easy to spot without them. The algorithm achieves considerable improvement when compared to CD.

4.3.4 PCCD

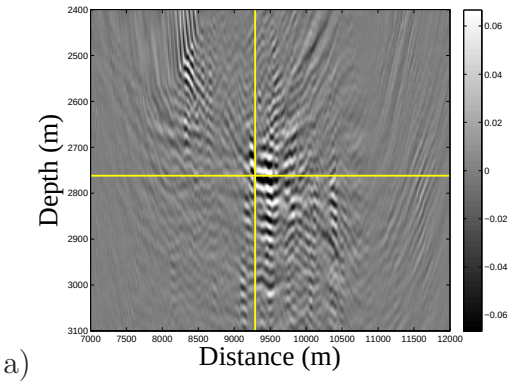
The baseline and monitor survey shot gathers are filtered employing the pseudo cross-correlation method. Then, the user migrates the output data invoking the split-step Fourier migration.



a)

Figure 4.13: The PCCD result.

Figure 4.13 captures the result of PCCD and PSDM performed one after the other.



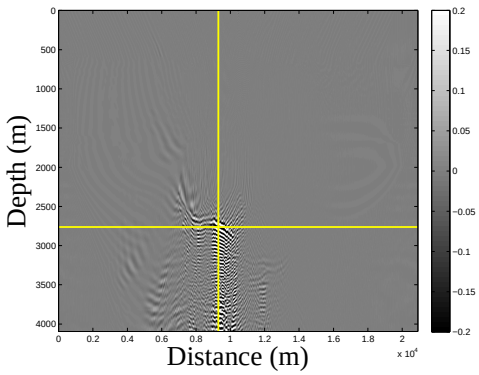
a)

Figure 4.14: The zoomed in PCCD result.

Figure 4.14 is the zoomed in version of Figure 4.13. PCCD shows considerable improvement to the result from CD. The location of the inserted reflector and its resolution are also improved when compared to the CCD result.

4.3.5 CICD

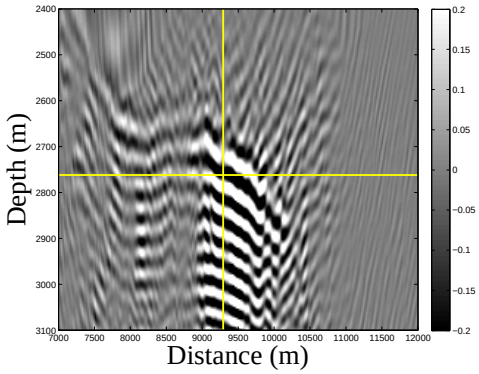
CICD is implemented as a pilot algorithm to combine CD and PSDM, which would otherwise involve user to manually execute each step sequentially.



a)

Figure 4.15: The CICD result.

Figure 4.15 captures the result of CICD algorithm.



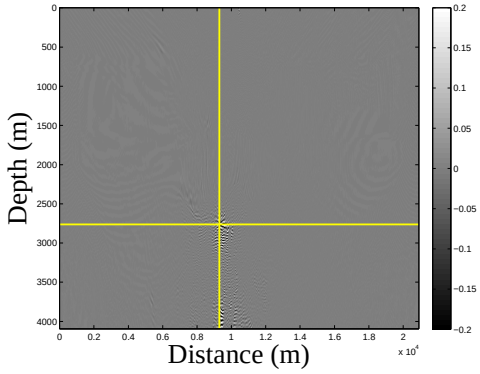
a)

Figure 4.16: The zoomed in CICD result.

Figure 4.16 is the zoomed in version of Figure 4.15. CICD shows no improvements when compared to CD and migration result as expected. It proves the workflow for combining PCCD and PSDM possible.

4.3.6 ICD

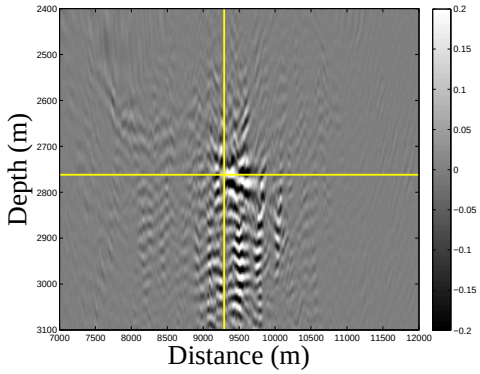
ICD is implemented as an algorithm to combine PCCD in PSDM.



a)

Figure 4.17: The ICD result.

Figure 4.17 captures the result of ICD algorithm.



a)

Figure 4.18: The zoomed in ICD result.

Figure 4.18 is the zoomed in version of Figure 4.17. The result shows improvements to the PCCD and the PSDM method as it eliminates dependence on the user to execute each step. The ICD result captures the inserted reflector location and cleans data around it when compared to the PCCD and the PSDM method executed sequentially.

4.4 Example II

4.4.1 Velocity models and synthetics

The second data set used to test the CCD and PCCD filtering is the 10^{th} SPE Comparative solution project. Here, I only employ difference filtering and omit the migration that is to be performed last. Recall the velocity and 2D synthetic seismic models generated from the Gassmann equations and the finite difference algorithm in acoustic medium presented in Chapter 2. The velocity models are shown in Figure 2.5 and the seismic synthetic models are shown in Figures 2.6(a), 2.6(b) and 2.6(c), respectively.

4.4.2 CCD

The seismic models are differenced employing CCD algorithm.

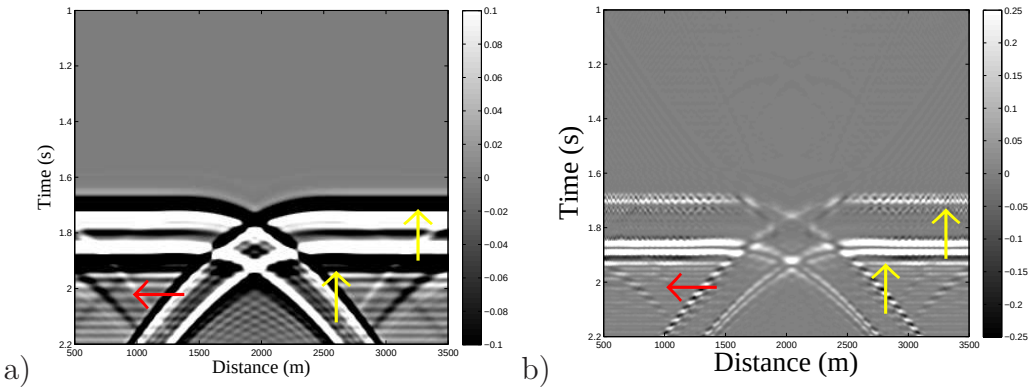


Figure 4.19: 2D seismic exploding reflector models after days 1 and 14 are differenced using a) CD and b) CCD algorithm. Note waterfronts and boundary effects denoted by yellow and red arrows, respectively.

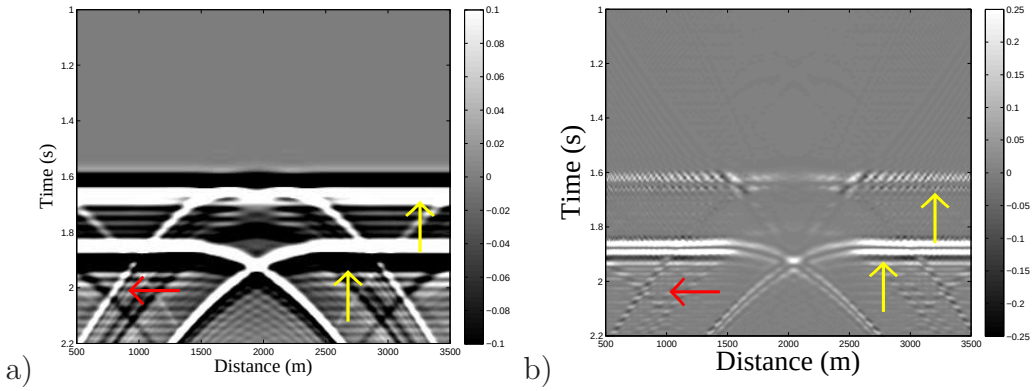


Figure 4.20: 2D seismic exploding reflector models after days 1 and 28 are differenced using a) CD and b) CCD algorithm. Note waterfronts and boundary effects denoted by yellow and red arrows, respectively.

Figures 4.19 and Figure 4.20 show results after the CD and CCD algorithm are ran for fluid flow change after days 14 and 28, respectively. Note waterfronts and boundary effects denoted by yellow and red arrows, respectively. When compared CD and CCD, fluid flow change is more pronounced on CCD plots and energy around them is minimized.

4.4.3 PCCD

Now, the seismic models are differenced employing PCCD algorithm.

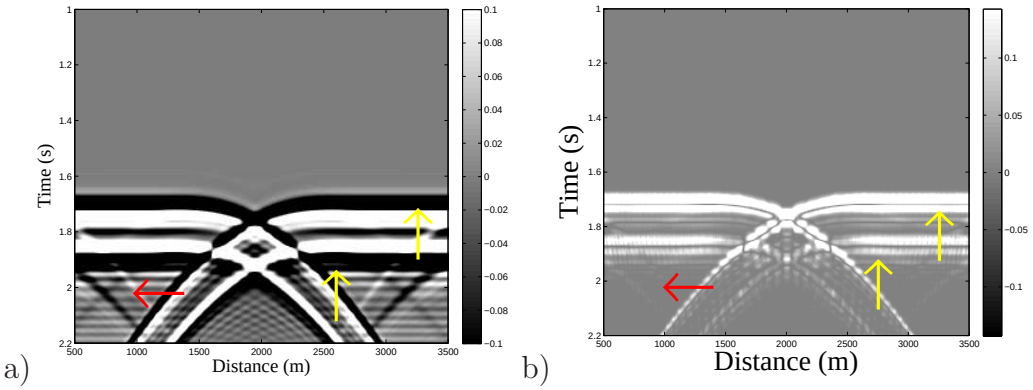


Figure 4.21: 2D seismic exploding reflector models after days 1 and 14 are differenced using a) CD and b) PCCD algorithm. Note waterfronts and boundary effects denoted by yellow and red arrows, respectively.

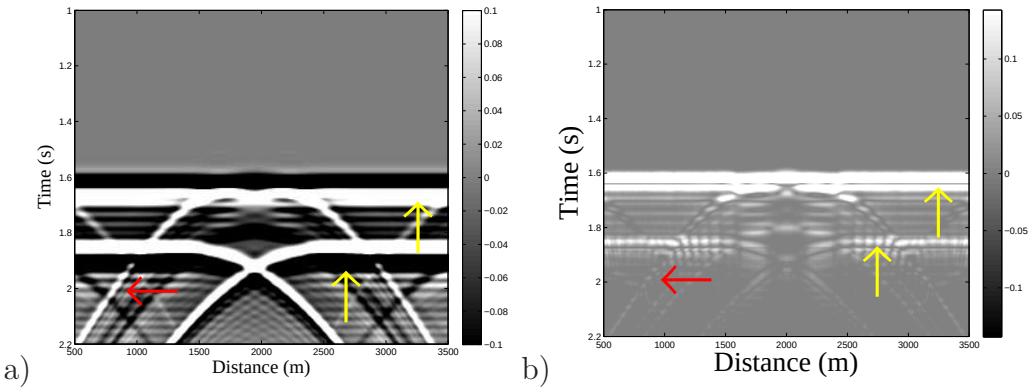


Figure 4.22: 2D seismic exploding reflector models after days 1 and 28 are differenced using a) CD and b) PCCD algorithm. Note waterfronts and boundary effects denoted by yellow and red arrows, respectively.

Figures 4.21 and Figure 4.22 show results after the CD and PCCD algorithm are ran for fluid flow change after days 14 and 28, respectively. Note waterfronts and boundary

effects denoted by yellow and red arrows as well, respectively. When compared CD and PCCD, fluid flow change is imaged better as there is less energy around them. Also, comparing CCD and PCCD plots, I observe improvement in main events to be better focused and scattered energy around them minimized.

4.5 Example III

4.5.1 Velocity models and synthetics

Recall the velocity and 3C-3D synthetic seismic models generated from the Gassmann equations and the finite difference algorithm in elastic medium presented in Chapter 2. The velocity models are shown in Figure 2.5 and the seismic synthetic models are shown in Figures 2.7, 2.8 and 2.9, respectively. Here, I only employ difference filtering and omit the migration that is to be performed last.

4.5.2 CCD

The synthetic seismic models are passed to the CCD algorithm.

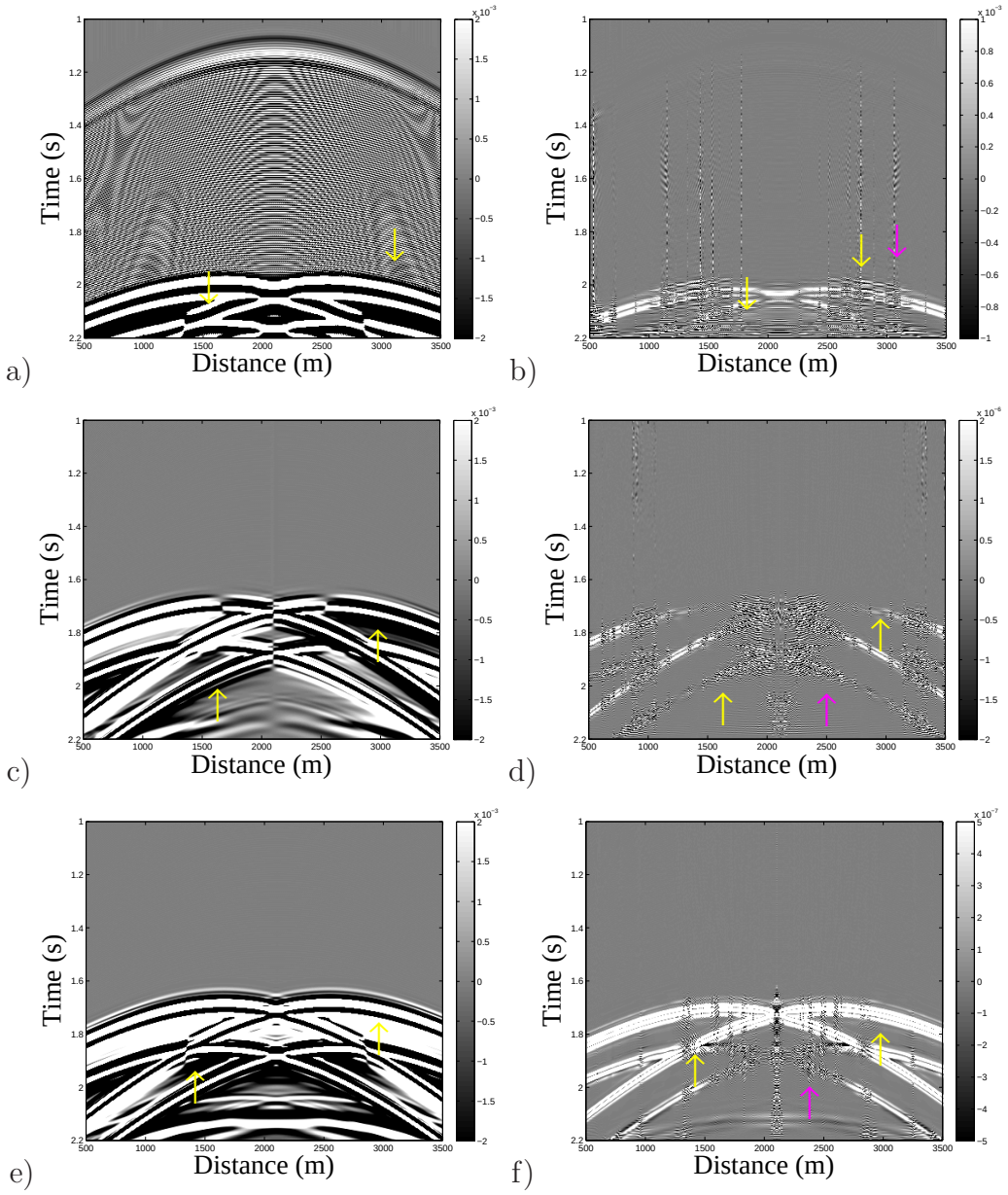


Figure 4.23: The CCD results on 3C-3D shot gather models: a) x-component CD of day 1 and 14, b) x-component CCD filtering of day 1 and 14, c) y-component CD of day 1 and 14, d) y-component CCD filtering of day 1 and 14, e) z-component CD of day 1 and 14, f) z-component CCD filtering of day 1 and 14. The yellow and magenta arrows denote waterfronts and numerical artifacts, respectively.

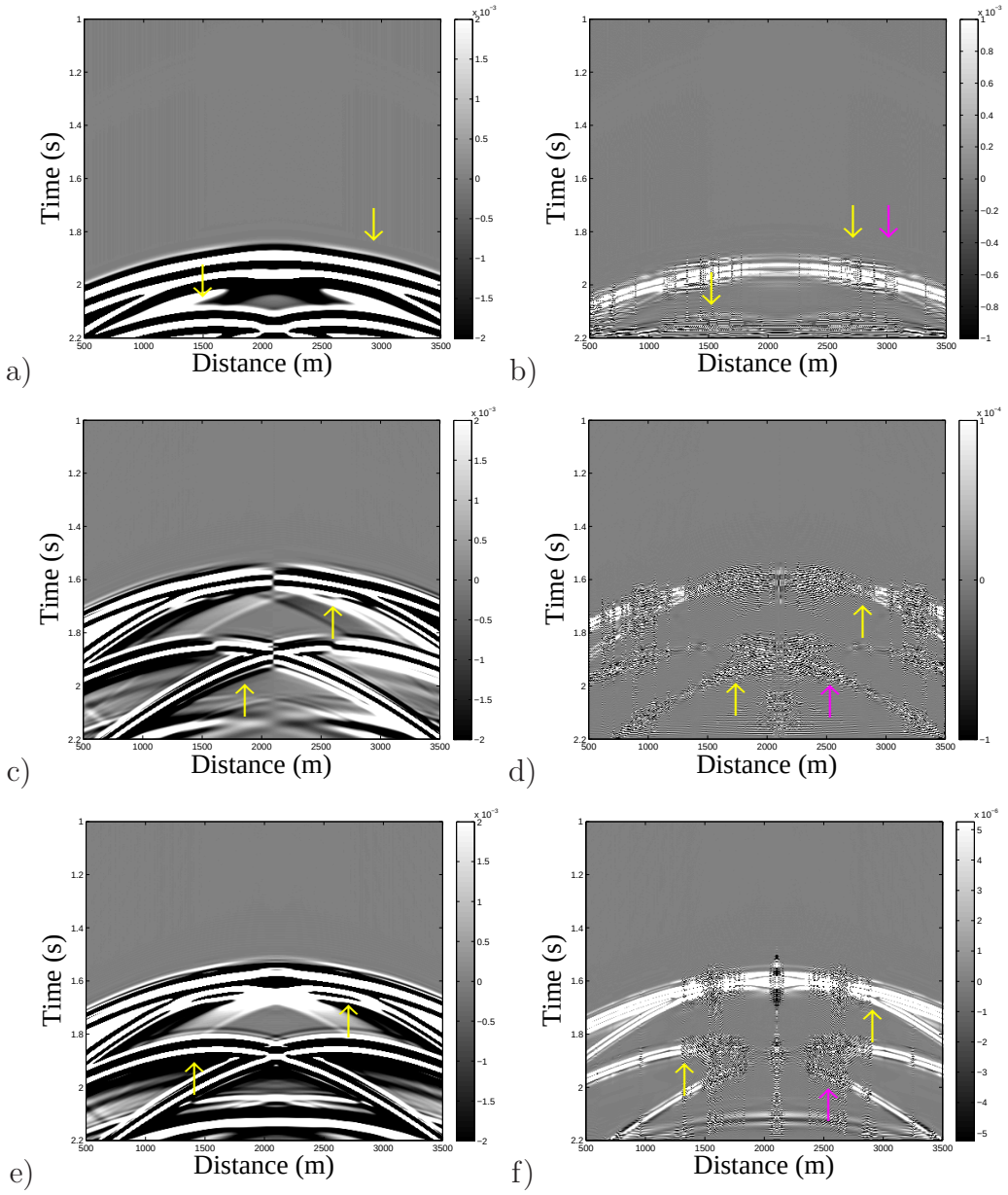


Figure 4.24: The CCD results on 3C-3D shot gather models: a) x-component CD of day 1 and 28, b) x-component CCD filtering of day 1 and 28, c) y-component CD of day 1 and 28, d) y-component CCD filtering of day 1 and 28, e) z-component CD of day 1 and 28, f) z-component CCD filtering of day 1 and 28. The yellow and magenta arrows denote waterfronts and numerical artifacts, respectively.

Figure 4.23 captures day 1 and day 14 differenced models. Models 4.23(a), 4.23(c) and 4.23(e) capture the CD filtering of days 1 and 14 of x, y and z components, respectively. Models 4.23(b), 4.23(d) and 4.23(f) capture the CCD filtering of days 1 and 14 of x, y and z components, respectively. The yellow arrows point to the location of the waterfronts after days 1 and 14. Figure 4.24 captures day 1 and day 28 differenced models. Models 4.24(a), 4.24(c) and 4.24(e) capture the CD filtering of days 1 and 28 of x, y and z components, respectively. Models 4.24(b), 4.24(d) and 4.24(f) capture the CCD filtering of days 1 and 28 of x, y and z components, respectively. The yellow arrows point to the location of the waterfronts after days 1 and 28. The magenta arrows point to the numerical artifacts produced by the algorithm. When compared the CCD filtered data is much easier to interpret as it eliminates amplitude ambiguities around events focusing only fluid flow changes.

4.5.3 PCCD

The same data set is used for testing the PCCD algorithm.

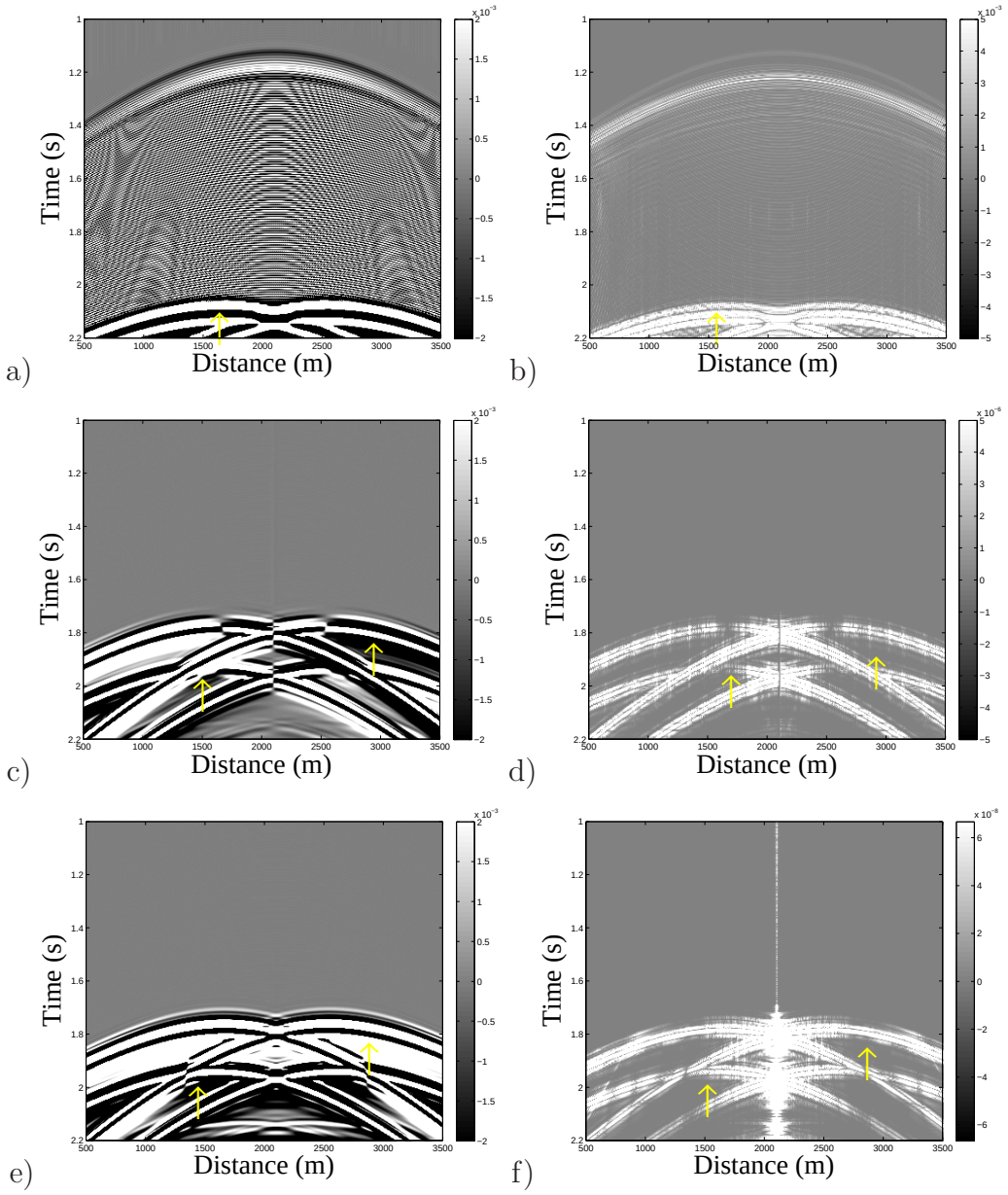


Figure 4.25: The PCCD results on 3C-3D shot gather models: a) x-component CD of day 1 and 14, b) x-component PCCD filtering of day 1 and 14, c) y-component CD of day 1 and 14, d) y-component PCCD filtering of day 1 and 14, e) z-component CD of day 1 and 14, f) z-component PCCD filtering of day 1 and 14. The yellow arrows denote waterfronts.

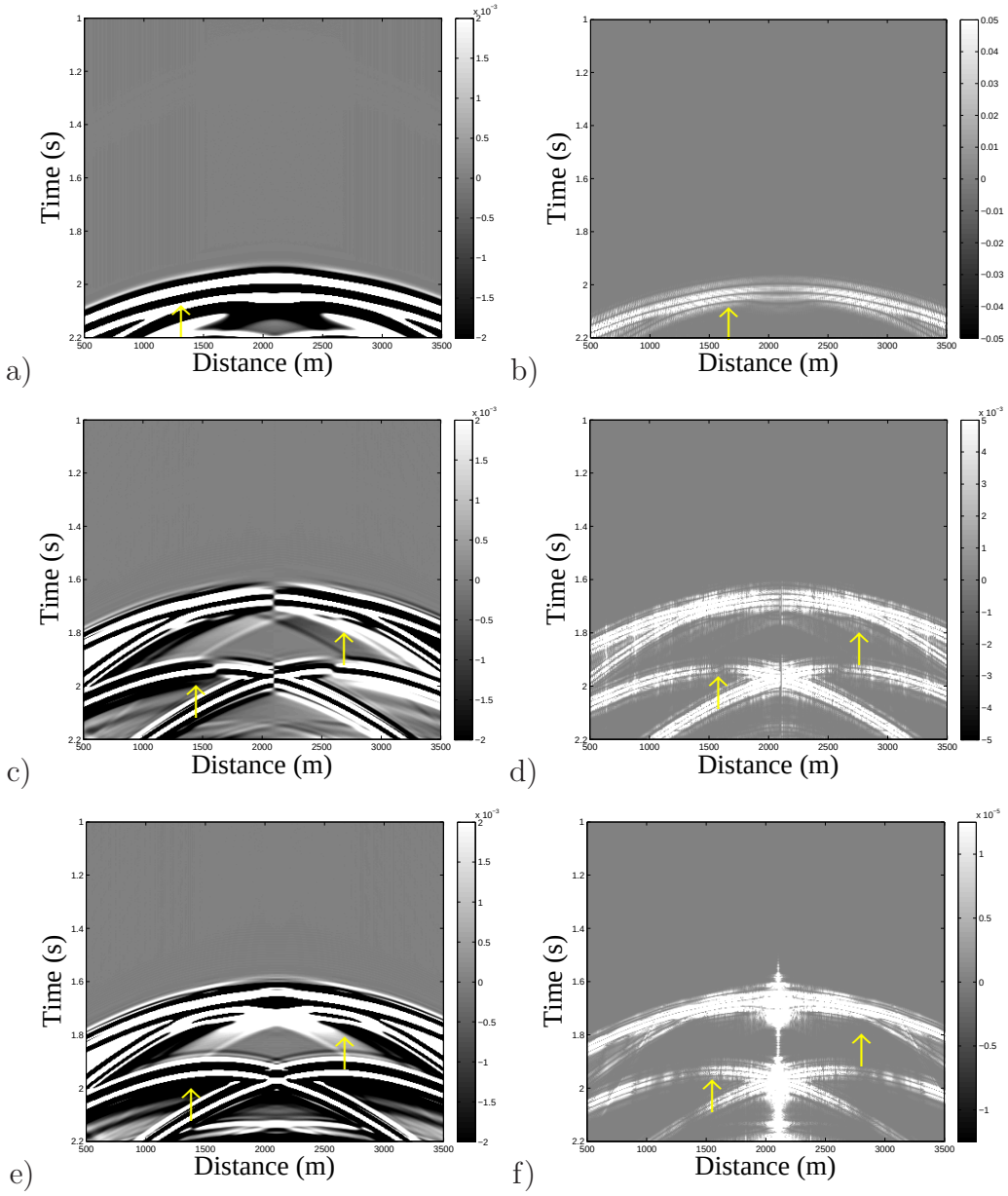


Figure 4.26: The PCCD results on 3C-3D shot gather models: a) x-component CD of day 1 and 28, b) x-component PCCD filtering of day 1 and 28, c) y-component CD of day 1 and 28, d) y-component PCCD filtering of day 1 and 28, e) z-component CD of day 1 and 28, f) z-component PCCD filtering of day 1 and 28. The yellow arrows denote waterfronts.

Figure 4.25 captures day 1 and day 14 differenced models. Models 4.25(a), 4.25(c) and 4.25(e) capture the CD filtering of days 1 and 14 of x, y and z components, respectively. Models 4.25(b), 4.25(d) and 4.25(f) capture the PCCD filtering of days 1 and 14 of x, y and z components, respectively. The yellow arrows point to the location of the waterfronts after days 1 and 14. Figure 4.26 captures day 1 and day 28 differenced models. Models 4.26(a), 4.26(c) and 4.26(e) capture the CD filtering of days 1 and 28 of x, y and z components, respectively. Models 4.26(b), 4.26(d) and 4.26(f) capture the PCCD filtering of days 1 and 28 of x, y and z components, respectively. The yellow arrows point to the location of the waterfronts after days 1 and 28. Note that numerical artifacts do not show on these plots. When compared the PCCD filtered data is much easier to interpret as it focuses fluid flow changes only. When CCD and PCCD are compared, PCCD shows favorable due to imaging no numerical artifacts and it is computationally economic.

4.6 Chapter summary

I implement four seismic differencing methods: 1) cross-correlation differencing (CCD), 2) pseudo cross-correlation differencing (PCCD), 3) conventional imaging condition differencing (CICD) and 4) imaging condition differencing (ICD). They are all based on the fact that seismic amplitude is less reliable to recording errors then the seismic phase. The CCD and PCCD algorithms are executed in the time and frequency domains, respectively. These algorithms perform cross-correlation, Gaussian filtering and inversion. The algorithms' results are passed to the PSDM. Although seismic difference imaging is improved, both algorithms depend on the user to manually move data from differencing to migration. The CICD algorithm is a combination of CD and PSDM. It is a pilot algorithm to combine the PCCD and PSDM. It proves to be efficient and robust when

compared to CD. The ICD method combines PCCD and PSDM in one algorithm, hence minimizes user's dependence and improves computational time and imaging.

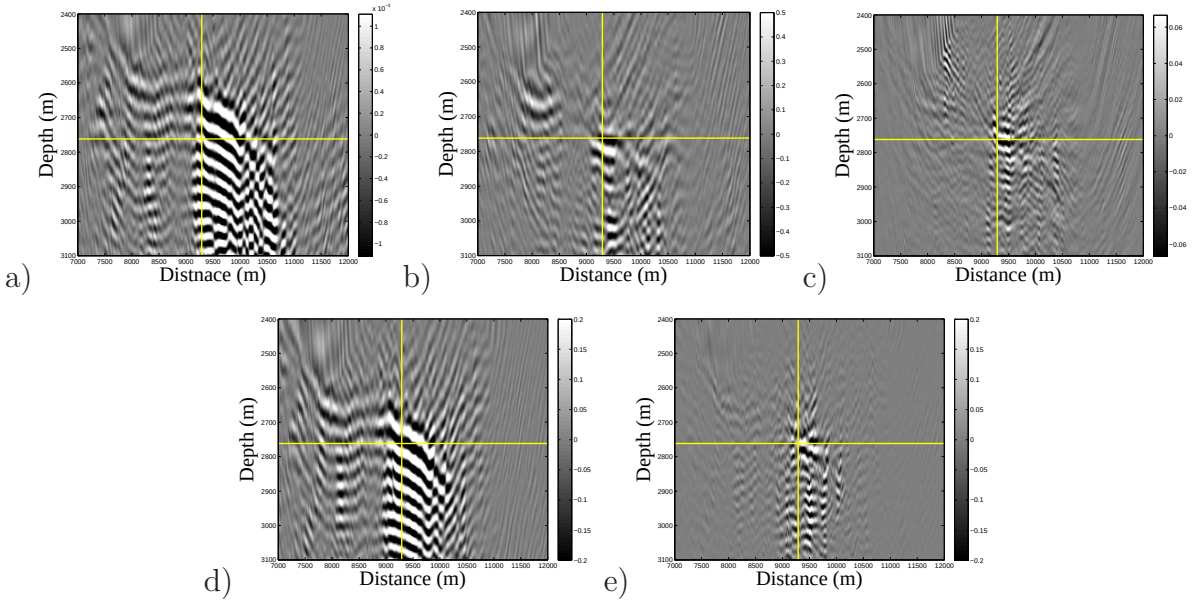


Figure 4.27: Summary of differencing methods: a) CD, b) CCD, c) PCCD, d) CICD and e) ICD.

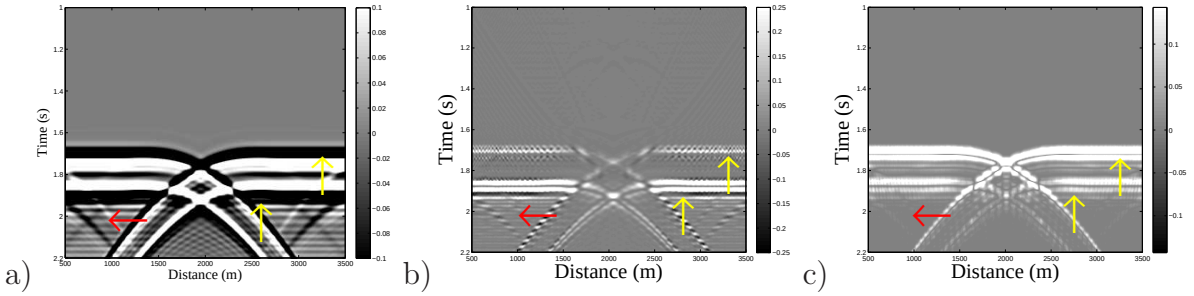


Figure 4.28: Comparison of 2D exploding reflector gather models after days 1 and 14 are differenced: a) CD, b) CCD and c) PCCD.

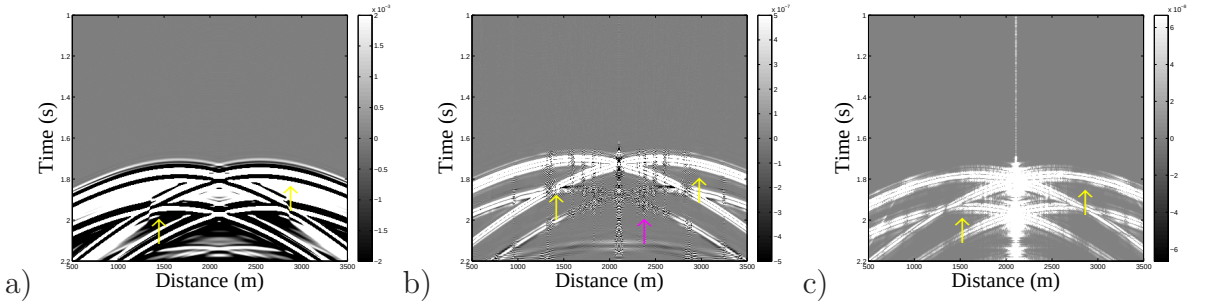


Figure 4.29: Comparison of 3C-3D shot gather models z-component after days 1 and 14 are differenced: a) CD, b) CCD and c) PCCD.

Figure 4.27 captures CD and four new seismic differencing algorithms run on the EGAE/SEG salt data set. Figures 4.29 and 4.28 capture CD and two new seismic differencing algorithms run on the waterflood data set.

In Figures 4.27, 4.28 and 4.29, CCD, PCCD and ICD highlight differences in time-lapse steps eliminating similarities by cross-correlation, filtering and inversion. The CCD, PCCD and ICD capture almost only fluid flow changes and eliminate almost all similarities on the differenced models.

The computational cost of non-conventional differencing methods varies. Assume the data is stored in an $M \times N$ matrix. The most expensive one is the CCD with $O(M^3N^3 + M^2N^2 + 3MN + MNlog(MN))$ operations to perform, where the ICD is the cheapest with $O(M^2N^2 + 2MN + 6MNlogMN)$ operations to execute. The examples prove to be significant improvement to difference imaging, hence can be used as an alternative tool in fluid flow monitoring, time-lapse studies and reservoir characterization.

Chapter 5

Conclusions and Future Work

5.1 Conclusions

This thesis work performs time-lapse analyses on two data sets. A time-lapse study takes place on a reservoir employing one producing and two injecting wells. The study performs the following: numerical simulation, analysis of rock physics and seismic modeling. The numerical simulation of fluid flow produces the saturation and pressure models. Then, the saturation models combined with rock properties deliver the velocity models as a result of the Gassmann equations. Further, the velocity models, through the finite-difference algorithms, generate 2D acoustic and 3C-3D elastic seismic models. The theoretical concepts are verified through numerical examples. There are subtle similarities and differences between acoustic and elastic models. The study proves both, acoustic and elastic models, to be assets in reservoir characterization.

The second data set used in testing is the EAGE/SEG salt velocity model. The original model is denoted as the baseline survey and then the identical model is modified by the author to accommodate its use in time-lapse study. The author inserts a small reflector in the sub-salt region to model changes due to production. The modified model is denoted as the monitor survey.

The synthetic seismic models of the waterflood scheme and the EAGE/SEG salt velocity models are migrated employing the split-step Fourier migration algorithm. Migrated sections are conventionally and non-conventionally differenced and compared.

The conventional seismic differencing is performed employing conventional matrix subtraction of two time-lapse matrices. It presents little value to reservoir characteriza-

tion and optimization as it captures amplitude ambiguity and masks fluid flow changes in a producing reservoir.

I observe that mentioned conventional seismic differencing carries various assumptions that can be distorted. It is assumed source/receiver coupling variations error to be negligible relative to the seismic response of fluid transport in a reservoir whose source/receiver positioning must be the same between time-lapse surveys. By conventional differencing, filtering followed by subtraction, we obtain a seismic image to interpret. Essentially, this image represents a change in fluid location superimposed upon certain background noise level. In practice noise can be large. It can produce differences in seismic amplitude, but there is often lesser effect on seismic phase.

Keeping these ideas in mind, non-conventional seismic differencing methods are implemented:

1. inverse data space differencing (IDSD),
2. cross-correlation differencing (CCD),
3. pseudo cross-correlation differencing (PCCD),
4. conventional imaging condition differencing (CICD) and
5. imaging condition differencing (ICD).

IDSD employs inverse data space theory to filter the migrated seismic models. This algorithm highlights differences, however, just dims similarities on the time-lapse difference models. CCD and PCCD are based on the cross-correlation operation, filtering and inversion. Both algorithms are user dependent to move from difference to migration and almost fully eliminate similarities and highlight differences. They are followed by the PSDM algorithm. CCD and PCCD algorithms execute in the time and frequency domains, respectively. In order to improve computational cost and eliminate dependence

on the user, I combine CD and PSDM to new algorithm named CICD. The resulting models of CICD are almost identical to CD as expected. The algorithm proves to be efficient and robust when compared to CD as the user dependence is eliminated. CICD algorithm is considered a pilot algorithm to combining PCCD and PSDM named ICD. ICD proves to focus and clear differenced models even more when compared to the CCD and PCCD results. It improves computational cost and robustness and eliminates user dependence.

The computational cost of non-conventional differencing methods differs. CCD is the most expensive, taking $O(M^3N^3 + M^2N^2 + 3MN + MNlog(MN))$ operations to complete and ICD is the cheapest taking $O(M^2N^2 + 2MN + 6MNlogMN)$ operations to complete.

The differenced waterflood time-lapse models capture waterfronts propagating upwards in time that are clearly identifiable when non-conventional differencing methods are used. Similarly, differenced EAGE/SEG salt time-lapse models capture inserted reflector when non-conventional differencing methods are employed.

All featured non-conventional algorithms except CICD detect significant imaging advances to fluid flow change, hence prove their use in geophysical interpretation, reservoir monitoring, characterization and time-lapse studies as an alternative tool.

5.2 Future work

It would be interesting to employ non-conventional differencing to the other time-lapse data sets to study their performance. For example, it would be worth evaluating them on channel models. Also, evaluation of non-conventional differencing algorithms on real time-lapse data sets would be valuable. These tests would give greater confidence to developed non-conventional algorithms. One of the most robust attributes assigned to

seismic data are the arrival traveltimes. It would seem that it would be worthwhile to compare results from amplitude differencing to results from differencing traveltime isochrones.

As noted in Chapters 2 and 4, there are some numerical artifacts on the seismic models, hence algorithm optimization should be attempted. Computation time of CCD still has a potential for improvement. CCD employs matrix inversion, therefore, improvement in computation cost will optimize the algorithm.

The Gaussian filter is used to eliminate zero-lag due to its robustness and easy implementation. It is worth considering other filtering methods to improve resolution.

Combining some of non-conventional algorithms would also be an interesting task in imaging improvement. For instance, combination of the IDS and ICD should be considered.

Bibliography

Aarnes, J. E., T. Gimse, and K. A. Lie, 2007, An introduction to the numerics of flow in porous media using MATLAB, *in* Numerical Simulation, and Optimization: Applied Mathematics at SINTEF: Springer, 265–306.

Agreste, S., and A. Ricciardello, 2011, Simulation of seismic wave propagation in 3d heterogeneous media: a parallel computing approach: Communications in Applied and Industrial Mathematics, Department of Mathematics, Messina University, Italy, 1–14.

Aminzadeh, F., N. Burkhard, J. Long, T. Kunz, and P. Duclos, 1996, Three dimensional SEG/EAGE models - an update: The Leading Edge, **15**, 131–134.

Beer, M. D., and J. W. Maina, 2008, Some fundamental definitions of the elastic parameters for homogeneous isotropic linear elastic materials in pavement design and analysis: the 27th Southern African Transport Conference.

Bentley, R., and Y. Zou, 2003, Time-lapse well log analysis, fluid substitution, and AVO: The Leading Edge, **22**, 550 – 554.

Berkhout, A. J., 2006, Seismic processing in the inverse data space: Geophysics, **71**, 29–33.

Berkhout, A. J., and D. J. Verschuur, 2005, Time lapse processing in the inverse data space: 75th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1522–1525.

Bertrand, A., S. M. Quaid, R. Bobokecki, and S. Leiknes, 2005, A high resolution workflow for 4D-friendly analysis: Application to gas-oil contact monitoring at troll west: 75th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 2422–2425.

Bini, D., 1995, Toeplitz matrices, algorithms and applications: ECRIM News Online Edition, **22**.

Charkraborty, S., 2007, An integrated geologic model of Valhall oil field for numerical simulation of fluid flow and seismic response: Master's thesis, University of Texas at Austin.

Cheng, A., L. Huang, and J. Rutledge, 2009, Time-lapse VSP data processing for monitoring CO_2 injection: 79th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 406–410.

Christie, M. A., and M. J. Blunt, 2001, Tenth SPE comparative solution project: A comparison of upscaling techniques: SPE Reservoir Engineering and Evaluation, Society of Petroleum Engineers, **4**, 308–317.

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

Cosse, R., 1993, Basics of reservoir engineering: Oil and Gas Field Development Techniques: Editions TECHNIP.

Du, Y., 2007, Prestack depth migration methods for isotropic and polar anisotropic media: PhD thesis, University of Calgary.

Ferguson, R. J., 1995, P-P and P-S inversion of 3-C seismic data: Blackfoot, Alberta: CREWES Research Report, **7**, 41.1–41.12.

———, 2009, CREWES 2D depth migration in MATLAB: A new release: CREWES Research Report, **21**, 1–9.

———, 2010, Theory of seismic imaging: Course notes, University of Calgary.

Ferguson, R. J., and G. F. Margrave, 1999, A practical implementation of depth migration by nonstationary phase shift: 69th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1370–1374.

———, 2005, Planned seismic imaging using explicit one-way operators: Geophysics, **70**, 101–109.

Hokstad, K., F. Maa, A. K. Nguyen, and R. Streich, 2007, 3D anisotropic elastic finite-

difference modelling in transversely isotropic media: SINTEF Petroleum Research, 1–45.

Holstein, E. H., 2007, Petroleum engineering handbook, volume v: Reservoir engineering and petrophysics: Society of Petroleum Engineers.

Huang, X., L. Meister, and R. Workman, 1998, Improving production history matching using time-lapse seismic data: The Leading Edge, **17**, 1430–1433.

Innanen, K., 2009, Inverting absorptive reflections: an inverse series tutorial: CREWES Research Report, **21**, 1–13.

———, 2010, Signal analysis: Course notes, University of Calgary.

Jin, L., and X. Chen, 2008, The combination of wavelet transform and nonlinear filtering for time-lapse seismic difference analysis: 78th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 3214–3218.

Kearey, P., M. Brooks, and I. Hill, 2002, An introduction to geophysical exploration: Blackwell Publishing.

Lines, L. R., and R. Newrick, 2004, Fundamentals of geophysical interpretation: SEG, monograph series.

Margrave, G. F., 1998, Theory of nonstationary linear filtering in the Fourier domain with application to time-variant filtering: Geophysics, **63**, 244 – 259.

———, 2008, Methods of seismic data processing: Course notes, University of Calgary and CREWES Project.

Mavko, G., T. Mukerji, and J. I. Dvorkin, 2009, The rock physics handbook: Cambridge University Press.

Mi, Y., 2002, Prestack depth imaging and velocity analysis for P-P and P-S data with nonstationary integral extrapolators: PhD thesis, University of Calgary.

Milicevic, V., and R. J. Ferguson, 2009, Numerical fluid flow modelling and its seismic response in time-lapse: CREWES Research Report, **21**, 1–13.

Rickett, J. E., and D. E. Lumley, 2010, Cross-equalization data processing for time-lapse seismic reservoir monitoring: A case study from the Gulf of Mexico: Geophysics, **66**, 1015–1025.

Schinelli, M., 2006, Using complex seismic attributes to improve 4D visibility of fluid contact movement: 66th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 3295–3298.

Schlumberger, 2011, Oilfield glossary: online resourse: http://www.glossary.oilfield.slb.com/.

Shearer, P. M., 1999, Introduction to seismology: Cambridge University Press.

Stoffa, P. L., J. T. Fokkema, R. M. de Luna Freire, and W. P. Kessinger, 1990, Split-step Fourier migration: Geophysics, **55**, 410–421.

Stoffa, P. L., L. Jin, M. K. Sen, and R. K. Seif, 2008, Time-lapse seismic attribute analysis for a water-flooded reservoir: Journal of Geophysics and Engineering, **5**, 210 – 220.

Vracar, V., and R. J. Ferguson, 2010, Conventional and non-conventional differencing in time-lapse: CREWES Research Report, **22**, 1–12.

Yao, Z., and G. F. Margrave, 1999, Elastic wavefield modelling in 3D by fourth-order staggered-grid finite difference technique: CREWES Research Report, **12**, 1–10.

Youzwishen, C. F., and G. F. Margrave, 1999, Finite difference modelling of acoustic waves in MATLAB: CREWES Research Report, **11**, 1–19.

Zou, Y., L. R. Bentley, and L. R. Lines, 2004, Integration of geophysical methods with reservoir simulation: 74th Ann. Internat. Mtg., Soc. Expl. Geophys., Expanded Abstracts, 1623–1626.