

**Problems with a localized nature in exploration
seismology**

by

Lucas Abraham (Bram) Willemsen

B.S., Earth Sciences, Utrecht University, The Netherlands (2009)

M.S., Mining and Geotechnical Engineering, LTU, Sweden (2011)

Submitted to the Department of Earth, Atmospheric and Planetary
Sciences

in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Geophysics

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2017

© Massachusetts Institute of Technology 2017. All rights reserved.

Author
Department of Earth, Atmospheric and Planetary Sciences
January 30, 2017

Certified by.....
Alison E. Malcolm
Associate Professor
Thesis Supervisor

Accepted by
Robert van der Hilst
Schlumberger Professor of Earth and Planetary Sciences
Head, Department of Earth, Atmospheric and Planetary Sciences

Dedicated
to my loving parents,
Paul and Karin Willemsen
and
my brothers
Rob and Maarten Willemsen

Problems with a localized nature in exploration seismology

by

Lucas Abraham (Bram) Willemsen

Submitted to the Department of Earth, Atmospheric and Planetary Sciences
on January 30, 2017, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Geophysics

Abstract

A wide range of problems in seismology have a localized nature. Examples include inversion for time-lapse change, salt boundary position and Amplitude or Phase Versus Angle (AVA/PVA). With increasing computational power, many inversion procedures have transitioned away from forward models with simplified physics and now simulate synthetic wavefields on a model representation of the Earth. These simulated wavefields form the basis for the model updates in the inversion. Usually these wavefields are simulated on the entire computational model, even though the region of interest and the associated updates of the computational model are localized.

In this thesis I explore ways of making use of the localized nature of the inversion by limiting the wavefield generation to the region of interest. This shrinking of the computational domain leads to faster solutions, reducing the time between the start and finish of the local inversion. This shorter turn-around time facilitates decision making and improves the interactivity of the inversion procedure. I first introduce a frequency domain local solver which can generate exactly the same wavefields as a full domain solver. I then apply this local solver to both a time-lapse problem and to salt boundary inversion. I then take a step back and look at time-lapse inversion outside the scope of local solvers. By looking at differences in the Full Waveform Inversion (FWI) gradient obtained from baseline and monitor datasets I compute a confidence map of locations likely to contain time-lapse change. This confidence map is then used to regularize a joint inversion for both the baseline and monitor datasets. I compare this methodology to other existing time-lapse algorithms on a synthetic North Sea model. In this study I notice the importance of postcritical reflections and become aware of their potential for reservoir characterization. Motivated by this observation I introduce a local solver for simulating elastic reflections in a region of interest. These elastic reflections can then be used for AVA and PVA to for instance invert for reservoir characteristics. The wavefield is efficiently propagated through the overburden using precomputed acoustic Green's functions obtained from a cheap acoustic solver.

Thesis Supervisor: Alison E. Malcolm

Title: Associate Professor

Acknowledgments

It is a strange sensation for me to realize that I have been at MIT for more than five years and that I'm about to finish my degree. I still very vividly remember arriving in Boston for the first time. When flying in I didn't know a single person in the USA. The sensation of emerging from the Kendall-MIT T-station for the first time and seeing the campus was both fantastic and intimidating at the same time. Like many incoming student I've been told, I felt quite some pressure to succeed in a place as competitive as MIT. I'm incredibly thankful to Chris Spiers, Hans Mattsson and Arie van den Berg who wrote letters of recommendation for me, which in my opinion made it possible to try this new challenge. Looking back at my time at MIT, it has proven itself to be an incredibly valuable experience for both my academic and personal development. I'm counting myself lucky to have had the opportunity to interact with so many great people since that moment. Brad Hager admitted me to MIT as advisor, but when I arrived I had never actually met him in person or talked with him about research. I was therefore very glad that in the process of getting to know him I learned that we could work well together. I'm thankful to Brad for the flexibility he displayed as advisor when I was getting ready for my qualifying exam. During this period I was not sure if I wanted to continue doing geomechanics. Brad was understanding and allowed me to explore and enroll in a wide range of classes. At that point seismology still seemed like an overwhelming topic to me. Almost every week during the weekly seminar hours there would be a discussion about seismology in one form or another. Usually I had no clue at all about the bigger picture and how these topics were related. While I was becoming convinced that seismology was a cornerstone of Geophysics, the mastery of this skill seemed reserved for people other than me. This slowly started to change when I took the class 12.510 'Introduction to Seismology', taught by Alison Malcolm.

Taking this class had an enormous impact on the direction of my research. Not only did I finally see the bigger picture in talks from others, but the way Alison taught seismology also made me enthusiastic about studying it myself. I am very thankful to

Alison for giving me the opportunity to do the second qualifying exam research project with her. During this time I got to know her as a patient and encouraging advisor with great knowledge about the mathematical aspects of seismology. I especially appreciate that she without exception always made time to talk. After the qualifying exam I decided that I wanted to focus on seismology for my thesis, but this would require switching main advisor from Brad to Alison. I'm very grateful to both of them for giving me the opportunity to agree to this change and follow this new academic path. I feel very fortunate to have had Alison as advisor, even with the distance issues that arose when Alison moved to Memorial University of Newfoundland and I stayed at MIT. While working with Alison I learned not only about seismology, but she also taught me very valuable life lessons which will stay with me long after my time at MIT. Another academically transformative experience was Laurent Demanet's class on 'Waves and Imaging'. This is the most rigorous class I've had on the fundamentals of Full Waveform Inversion. Without Laurent's deep insight on this topic, this thesis would have looked very differently. His recurring feedback on my thesis research was very valuable and I'm thankful for all these interactions. Around this time I also met Russell Hewett who created the open source seismic inversion code PySIT. From him I learned a lot about numerical methods, mathematical operators, version control and contributing to open source codes. All these methods have helped me tremendously during my Ph.D. Interactions with Marlies Vasmel, Filippo Broggin and Johan Robertsson from ETH in Zurich have also been very valuable throughout my Ph.D.

Quite early after deciding to focus on seismology I wanted to supplement the mathematical viewpoint at MIT with practical experience from people working in industry. My first experience working with industry professionals came quickly after my qualifying exams. Alison connected me with Paul Williamson at Total, but I had no high hopes since it was already the end of May and far too late for the regular recruitment season. But Paul managed to give me an opportunity that year anyway, working with Andreas Atle for Total in Houston. I'm very grateful to both of them and Total for giving me the opportunity to see seismology applied to real

world problems. It was such a good experience that I signed up for an internship every summer since then.

Subsequent years I interned with Schlumberger, ConocoPhillips and ExxonMobil. I am incredibly grateful in particular to both Schlumberger and ConocoPhillips for allowing me to work on thesis work during internships and the permission to integrate it into my thesis. At Schlumberger I got to work and interact with some of the most talented people I know. These people include but are not limited to Winston Lewis, Chirag Parekh, Pearl Flath, Richard Coates and Darrell Coles. At ConocoPhillips I want to thank Arcangelo Sena, Jun Cao and Baishali Roy for valuable research interactions and their support for this thesis. The summer internship at ExxonMobil was perhaps the internship I learned the most about seismology. Working in time-lapse group with David Johnston, Musa Maharramov, Ganglin Chen and Partha Routh under the supervision of Eric Wildermuth was a fantastic experience. During this internship, more like any other, I felt part of the team and had the opportunity to interact with and learn a lot from all these experts. I'm very grateful for this opportunity. I am also thankful to the administrators at EAPS, in particular Vicky McKenna and Roberta Allard. I must also thank the executive director at the Earth Resources Laboratory (ERL), Anna Shaughnessy, who was always ready to help on a variety of topics. When I was the president of the SEG student chapter I asked her so often for feedback or to release funds for our activities and projects, but she was always very enthusiastic and willing to help. Sue Turbak, Natalie Counts, Josh Kastorf and Terry Macloon also helped me a lot with all kinds of administrative questions in ERL. In order to study and do internships in the USA as an international student I had to navigate through sometimes complex regulations. I am therefore very thankful for the high quality help I received at MIT's international office, in particular from Janka Moss.

While MIT is a vast and fantastic place to learn, it can be overwhelming from time to time. During the years I have felt great support from my many people who have made the experience much more pleasant. The unconditional love of both my family and Eva was a great and steady support during the years. During my five

years in Cambridge I lived with several people. It has been really fun living with Stefanos Nikolaidis, Angel Sevil, Yichao Pan, Sharon Newman, Rebecca Smith, Josh Ehrlich, Stephen Messenger, Dustin Hayden and Deepak Cherian. I think fondly back to the great community the students have at the Earth Atmospheric and Planetary Sciences (EAPS) department. The most memorable social interactions happened at regular social gatherings like peer hour, Halloween and thanksgiving parties, the annual Long Pond retreat and the numerous house parties including the wine-tastings hosted by Ben Mandler. The friendship with other students at MIT, in particular Jaap Nienhuis, Natasha Maas, Elena Steponaitis, Erik Lindgren, Josimar da Silva and Thomas Wortmann was very valuable to me. I also enjoyed daily interactions with the other students at ERL, not limited to Arthur Olive, Ruel Jerry, Alan Richardsson, Sudhish Bakku, Yuval Tal, Jing Liu, Saleh Nasser, Haoyue Wang, Di Yang, Chen Gu, Ali Aljishi, Greg Eli, Jamie Potter, Harry Matchette-Downes, Anna Rogers, Fuxian Song, Beatrice Bienvenu, Di Yang, Ahmad Zamanian, Andrey Shabelansky, and Clarion Hess.

I want to finish by thanking for the financial support from MIT, ERL and the consortium members. While studying at MIT I was fortunate to receive generous fellowships from ConocoPhillips, Chevron (through the MIT Energy Initiative) and the Nafi Toksöz fellowship fund. Without their support I would not have been able to afford to study at MIT. Finally I want to thank Arie van den Berg who connected me with Rob van der Hilst. Rob first talked with me about applying to MIT and explained to me you do not need to be rich to apply. I'm very grateful for this, because without Rob I would not have known about this opportunity and would therefore not have made it to MIT. Therefore I am happy to see that Rob is also here as committee member now I am finishing my degree.

Contents

1	Introduction	27
1.1	Thesis Outline and Summary of Contributions	30
2	An efficient numerically exact local solver: Theory	35
2.1	Introduction	35
2.2	Theory	40
2.2.1	Numerically exact boundary equation for the scattered field	42
2.2.2	Computing the perturbed wavefield in the truncated domain	44
2.2.3	Full domain Green's function precomputation	48
2.2.4	Verification of numerical exactness of the local solver	49
2.3	Discussion	50
2.4	Conclusions	53
2.5	Acknowledgements	53
3	An efficient numerically exact local solver: Verification and applica- tion to problems with a localized nature	57
3.1	Introduction	58
3.2	Application to salt boundary inversion	58
3.2.1	Combining the local solver with the salt boundary inversion method	58
3.2.2	Application to salt boundary inversion	60
3.2.3	Multi domain inversion	62
3.2.4	Discussion of the local solver applied to salt inversion	66

3.3	Application to time-lapse inversion	70
3.3.1	Problem description	72
3.3.2	Time-lapse inversion	73
3.3.3	Discussion	74
3.4	Acknowledgements	75
3.5	Conclusion	76
4	Regularizing velocity differences in time-lapse FWI using gradient mismatch information	93
4.1	Introduction	94
4.2	Method	95
4.3	Numerical investigation	97
4.3.1	Setup of synthetic problem	97
4.3.2	Calculating the confidence map $\beta_g(\mathbf{x})$	98
4.3.3	Comparing GC-FWI and DDFWI	99
4.4	Discussion	100
4.5	Conclusion	101
4.6	Acknowledgements	101
5	Evaluating the performance of acoustic FWI time-lapse workflows on realistic synthetic data	109
5.1	Introduction	110
5.2	Methodology	114
5.2.1	Baseline velocity model	114
5.2.2	Time-lapse perturbation	115
5.2.3	Data generation	115
5.2.4	Baseline inversion algorithm	118
5.2.5	Time-lapse algorithms	119
5.3	Results	123
5.3.1	Parallel FWI	125
5.3.2	DDFWI	126

5.3.3	Joint FWI	127
5.3.4	Elastic data	127
5.3.5	Limiting the maximum offset	130
5.3.6	Role of the initial model	131
5.3.7	Gradient based confidence map	132
5.4	Discussion	134
5.5	Conclusions	137
5.6	Acknowledgements	137
6	An efficient coupled acoustic-elastic local solver applied to phase inversion	159
6.1	Introduction	159
6.2	Methods	165
6.2.1	Local solver	166
6.2.2	Standard elastic local solver	166
6.2.3	CDA coupling	171
6.2.4	New approximate injection approach for variable ρ/V_s along the boundary of the local domain	174
6.2.5	Exact boundary integral	176
6.2.6	Approximate boundary integral for variable ρ/V_s along the boundary of the local domain	177
6.2.7	Phase measuring	178
6.3	Results	179
6.3.1	Two layer example	179
6.3.2	Synthetic North Sea model	183
6.3.3	Inversion	190
6.4	Discussion	192
6.5	Conclusions	197
6.6	acknowledgements	198

7	Conclusions and Future Directions	225
7.1	Summary of main contributions	225
7.2	Directions for Future Work	227
A	Derivation of numerically matching scattered field	231
B	Numerically exact multi domain solver	239
C	Parallel FWI vs Joint FWI	243
D	Heuristic amplitude compensation factors	251
D.1	Injection compensation factor $V_s(\mathbf{x})$	251
D.2	Injection compensation factor $\rho(\mathbf{x})$	252
D.3	Remaining compensation factors	252
E	Wavefield amplitude scaling due to smooth density variation	257
F	The high amplitude artificial scatter event in Local C	261

List of Figures

2-1	The geometrical setup of the problem. Equation 2.1 defines the full computational domain Ω as the union of A , B , ∂C and C . The truncated domain is enclosed by the dashed red line. The nodes on the boundary of the truncated domain are in the set ∂C . The interior nodes of the truncated domain are in the set C . All the source and receiver nodes are in B . The size of zone B is not limited in any direction. It is normally much larger than the size of truncated domain C . The remainder of the computational domain, including the boundaries, is in set A . The nodes enclosed by the dotted black box are used later in this chapter to guide the derivation of the scattered field.	54
2-2	The 3480 by 600 nodes velocity model on which the wavefields are generated. The 758 by 133 nodes truncated domain is indicated by the box. Within this box the model is perturbed from the initial model on which the Green's functions are generated, and the wavefields are then compared in this domain.	55
2-3	Verification of the local solver using the Pluto model. (a) Real part of the wavefield from the full domain Helmholtz solver. (b) The result from the full domain solver is enlarged within the black box. (c) Real part of the wavefield in the black box from the truncated solver. An equally good fit is obtained for the imaginary part.	56

3-1	Salt boundary inversion results for the Pluto model where the true and initial model only differ within the interior of the truncated domain. (a) True truncated domain. (b) Initial truncated domain. (c) Inverted truncated domain.	78
3-2	(a) True 3201 by 1201 nodes Sigsbee2b velocity model with a 691 by 221 nodes truncated domain. (b) Initial velocity model with MVA sediment and an incorrect salt bottom everywhere, not just inside the truncated domain indicated by the black box. The top of the salt is correct in this example.	79
3-3	(a) True Sigsbee2b truncated velocity model from Figure 3-2a. (b) Initial truncated model from Figure 3-2b. (c) Inverted truncated model from local solver. Significant improvement can be observed.	80
3-4	Truncated domain inversion result embedded in the unchanged full domain initial model illustrating that abrupt changes can take place at the boundary of the truncated domain.	81
3-5	Migration in the true, initial and inverted models of Figure 3-3. The initial model uses an MVA sediment velocity and has a very incorrect salt bottom. The arrows show the locations of interesting features that appear after the salt bottom estimate has been improved.	82
3-6	The initial model for the multi-domain inversion. The top truncated domain is 347 by 91 nodes and the bottom domain is 347 by 111 nodes. The true model remains the same, shown in Figure 3-2. The white ray shows that a primary reflection on the salt bottom is influenced by the top domain twice. This demonstrates that it is important that the local solver correctly accounts for waves reentering the truncated domain. Waves reflecting multiple times benefit from this even more.	83
3-7	This figure shows the improvement in the top salt estimate. Inversion takes place simultaneously in both of the truncated domains in Figure 3-6. The initial and true model differ significantly everywhere. The white line shows the true salt boundary.	84

3-8	This figure shows the improvement in the bottom salt estimate. Inversion takes place simultaneously in both of the truncated domains in Figure 3-6. The initial and true model differ significantly everywhere. The white line shows the true salt boundary.	85
3-9	Migration artifacts around the salt bottom in the initial model still give an indication of the correct salt shape. Less distorted initial models have been investigated where the correct salt bottom was not recognizable from migrated images, but simultaneous inversion still improved the salt bottom estimate. The large salt perturbations in Figure 3-7b and Figure 3-8b make it easier to visualize improvements in the velocity model and the migrated image. Arrows show interesting features. The location of these arrows is slightly perturbed from those in the true model in Figure 3-5. But they show the same geological features.	86
3-10	Schematic cost comparison for running 5 simultaneous inversions using different inversion parameters and initial conditions. The length of the bars is to scale.	87
3-11	Figure 3-11a shows the true Marmousi II baseline model used in this study. Figure 3-11b zooms in on the white box of Figure 3-11a to show the true time-lapse change in the three reservoirs. The black boxes surrounding the reservoirs outline the multi-domain local solver. . . .	88
3-12	Figure 3-12a shows the initial baseline model. Figure 3-12b shows the inverted baseline obtained at the end of a frequency sweep. The sweep inverts for one frequency at a time, starting at 3.0 Hz and subsequently inverting for 4.0 Hz, 5.0 Hz, 6.5 Hz, 8.0 Hz and finally 10.0 Hz. The black boxes surrounding the reservoirs outline the multi-domain local solver and are identical to those in Figure 3-11a.	89

3-13	To test the local solver we compare the wavefields from the full domain solver and the local domain solver and plot the results in the white box of Figure 3-12b. Within the truncated domains (i.e. black boxes) the two wavefields are numerically identical. This means all interactions between the domains, including local model updates, and the exterior are included. This verifies that the multi-domain configuration of the local solver is setup correctly for this problem.	90
3-14	DDFWI results when sequentially inverting for the frequencies 5.0 Hz, 6.5 Hz, 8.0 Hz and 10.0 Hz. Compare the inversion results with the true time-lapse of Figure 3-11b. Figure 3-14a shows the result when inverting for time-lapse change in the full domain using a full domain solver with 16 iterations at each frequency. Figure 3-14b uses the setup, but reduces the number of iterations to 3 at each frequency. Finally Figure 3-14c also uses 3 iterations at each frequency but uses a local solver instead of a full domain solver and only updates the model within the black boxes.	91
4-1	The true baseline model.	102
4-2	The baseline model inverted from the baseline data using standard FWI for $r = 0.64$	103
4-3	The true time-lapse velocity change investigated in this study.	104
4-4	Confidence map $\beta_g(x)$ for GC-FWI for noise levels $r = 0.04$ and $r = 0.64$. The black and red arrows correspond to the preconditioned gradient evolution curves of the same color in Figure 4-5.	105
4-5	Evolution of the preconditioned gradients of both data terms of (4.1) in the inversion, at the locations indicated by arrows of corresponding color in Figure 4-4. True time-lapse regions correspond with gradients of large amplitude and different signs (black) whereas in regions without timelapse changes the gradients have small amplitude and often the same sign (red).	106

4-6	Comparison of GC-FWI using three different values of the model-regularization weighting term ϵ with DDFWI for noise level $r = 0.64$. Figure 4-3 shows the true model time-lapse change. At low regularization (top) GC-FWI gives results similar to those of DDFWI, but as ϵ is increased GC-FWI suppresses spurious changes and results in a more accurate recovery of the time-lapse changes.	107
5-1	True models with 2.0 m vertical spacing and 12.5 m horizontal spacing: (a) V_p (b) ρ (c) V_s . The black box marks the region on which the time-lapse study primarily focuses.	138
5-2	True time-lapse zoom corresponding to black box in Figure 5-1a: (a) V_p (b) ρ (c) V_s	139
5-3	Baseline pressure shot gathers for a) CDA, b) VDA and c) elastic for a shot at $x = 2.0$ km. Several events of interest are marked with numbers. 1) The head wave propagating along the bottom of the ocean. 2) Direct wave. 3) Top of reservoir reflection. These three events are visible in all three simulations. 4) Reflection on low density layer. 5) Slow events due to introduction shear velocity.	140
5-4	Baseline CDA inversion results starting from initial model (a) for different types of input data. (b) inverse crime data. (c) CDA data. (d) VDA data. (e) EL data. Figure 5-1a shows the true P-wave velocity.	141
5-5	Parallel FWI for P-wave velocity. (a) and (b) use inverse crime data and (c) and (d) use CDA data from SeisSpace: (a) and (c) start from the smooth initial model of Figure 5-4a, (b) starts from Figure 5-4b and (d) starts from Figure 5-4c. True reference is displayed in Figure 5-2a.	142
5-6	Parallel FWI depth profiles with labels referring to Figures 5-5a - 5-5d.	143
5-7	DDFWI results for the inverse crime, and the three SeisSpace datasets (CDA, VDA and EL) using the corresponding baseline models from Figures 5-4b - 5-4e.	144

5-8	DDFWI depth profiles with labels referring to Figures 5-7a - 5-7d. . .	145
5-9	Joint FWI for elastic data.	146
5-10	Depth profiles comparing the elastic DDFWI and Joint FWI time-lapse estimates along a vertical profile.	147
5-11	Zoom in on large-offset top of reservoir reflection for earlier pressure shot gathers in Figures 5-3b and 5-3c. In Figures 5-11a and 5-11b we see that these events have large differences in phase in the VDA and EL data. We also see that the time-lapse shot gathers δd_{VDA} and δd_{EL} in Figures 5-11c and 5-11d are slightly shifted in time and have slightly different shape.	148
5-12	Each source-receiver combination contributes to the DDFWI objective function of (5.9) with the EL data. This Figure shows the results on the initial (i.e. inverted baseline) model where the residual is δd_{EL} . Minor deviations from diagonal symmetry in the Figure are the result of the source spacing being coarser than the receiver spacing.	149
5-13	Figure 5-12 showed the EL DDFWI objective value contributions of each source-receiver combination. Here we show the relative change of the objective value contribution for each source-receiver combination at the end of the inversion, where the time-lapse estimate is displayed in Figure 5-7d.	150
5-14	Figure 5-12 showed the EL DDFWI objective value contributions of each source-receiver combination. Here we show the relative change of the objective value contribution for each source-receiver combination at the end of the inversion when we limit the maximum source-receiver offset to 2.0 km. The source-receiver combinations used in the EL DDFWI inversion fall within the two black diagonal lines. The source-receiver combinations outside these two black lines have offset exceeding 2.0 km. They are plotted here to inspect how their objective function contribution changes, even though they are not used in the inversion.	151

5-15	DDFWI with the maximum source-receiver offset limited to 2.0 km. The ellipse highlights an artifact that was less pronounced when all offsets were used in Figure 5-7d.	152
5-16	DDFWI with the maximum source-receiver offset limited to 2.0 km starting on the CDA baseline velocity model of Figure 5-4c.	153
5-17	Zoomed in section of the gradient based confidence map. The four stars correspond to four points at which we plot the illumination compensated gradient evolution in Figure 5-18. The blue star ($\beta_g = 0.92$) corresponds to the reservoir time-lapse area. The red ($\beta_g = 0.08$) and magenta ($\beta_g = 0.70$) stars are in the left and right elliptical overburden perturbations respectively. The yellow star ($\beta_g = 0.01$) is in the region of Figures 5-15 and 5-16 where and artificial time-lapse reflector is seen.	154
5-18	The squares and circles represent the illumination compensated baseline and monitor gradients respectively at four pixels of interest. These pixels are indicated by stars of matching color in Figure 5-17.	155
5-19	Same setting as Figure 5-16. DDFWI with the maximum source-receiver offset limited to 2.0 km starting on the CDA baseline velocity model of Figure 5-4c. We use the confidence map from 5-17 to regularize the model difference. The ellipse corresponds to the incorrect event which was visible in Figures 5-15 and 5-16, but is attenuated significantly.	156
5-20	Comparing Figures 5-15,5-16 and 5-19 along a depth profile.	157
6-1	Schematic drawing of a local solver (red box) within a full domain solver (black box). The local solver used in this study is a modification of the local solver introduced by Robertsson and Chapman (2000). It discards waves scattering on local model updates $\Delta m(x)$, which then exit and subsequently reenter the local domain, like the gray event. .	199

6-2	Staggered elastic grid around the top-right boundary of the elastic local solver. The various colored lines are used at different points in the chapter, where their purposes are also described.	200
6-3	Schematic of the local solver. The gray lines represent the required background CDA Green's functions between the sources, receivers and boundary of the local solver. The model update within the local solver is elastic, with the region between the red and the blue line representing a ρ and V_s taper. The blue box contains exactly the part of the elastic model we want to model.	201
6-4	True elastic model, with the source at $x = 1.0$ km and $z = 0.1$ km. Density and S-velocity models are tapered from fully elastic inside the blue box to CDA at the red box in Figure 6-5. Material properties of the two layers are summarized in Table 6.1.	203
6-5	Tapered density and shear velocity models from Figure 6-4. Inside the blue box the models are fully elastic and the same as in 6-4. Between the two boxes the model is tapered towards CDA material properties, which implies the V_s taper value is 0.0 m/s. We choose the constant density taper value $\rho = 2000.0$ kg/m ³ , which is equal to the density of the top layer in the full elastic model of Figure 6-4.	204
6-6	Gathers. The number 1 marks the direct wave. The number 2 marks the reflection. The number 3 marks the location where the headwave peels off and where we clearly enter the postcritical regime (to the right, larger offset). The number 4 marks the reflection of the incident pressure wavefield on the top taper boundary. This event is not present in the true simulation because there is no taper. The same amplitude clipping is applied to all three simulations so the color scales correspond.	206

6-7	Traces at $x = 3672.5$ m for a shot at $x = 1.0$ km. Applying the described single scalar amplitude correction to the full domain taper and local shot gathers. Notice the excellent agreement between the different models. Only at the intersection of events 3 and 4 (between 1.50 s and 1.65 s) do we see some differences. The simulation on the true model does not have the taper influence.	207
6-8	Phase shift with respect to incidence angle at the reflector. The dashed lines are obtained by subtracting the zero offset reflection phase (which is approximately shifted by -45 degrees because we are using a 2D model). The phase error between the estimated phase in the true and tapered/local result is four degrees at most.	208
6-9	True V_p , ρ and V_s for the North Sea model. The image of V_p is the background velocity model on which the CDA Green's functions are computed. The outer red box and inner blue box correspond with those in Figure 6-3. The lines connecting the outer red box and inner blue box are used in subsequent tests where parts of the taper are eliminated.	209
6-10	Shot gather for a source at $x = 2.5$ km with the red box indicating the zoomed region shown for the local solvers in Figure 6-11. This gather is generated on the true elastic model. The blue line represents the window later used to measure the phase of the reflection.	210
6-11	Sections of shot gathers for a source at $x = 2.5$ km in four different scenarios, within the region marked with the red box in Figure 6-10. The horizontal axis is the receiver coordinate. Three different locations of interest are marked. 1: The top of reservoir reflection. 2: The head-wave separating from this reflection, marking the transition to the post-critical regime. 3: Location where the traces are plotted in Figure 6-12. The phase has obviously rotated with respect the nearer offset reflections before the critical angle. The amplitude scale is the same in all Figures.	212

6-12	Traces corresponding to the four scenarios in Figures 6-11a - 6-11d in the same order at $x = 8.0$ km marked by the number 3 in the first panel. The red line almost perfectly matches the blue line.	213
6-13	This figure plots the wavefield simulations for configurations Local B, Local C and Local D respectively. Similar to Figure 6-11 the source is placed at $x = 2.5$ km. The red arrow in Figure 6-13b points to an artificial high amplitude scatter event. The amplitude scale is the same in all Figures.	214
6-14	The black trace is the same true trace as in Figure 6-12. The other three traces correspond to the solver configurations in Figure 6-13. The red and green line nearly overly one another.	215
6-15	The extracted gathers within the window indicated by blue lines in Figure 6-10. Plot (a) shows the true gather and plots (b)-(e) show the four local results. The amplitude scale is the same in all Figures. . .	216
6-16	Correlation values obtained when rotating each windowed trace and then correlating it with its envelope for a source at $x = 2.5$ km. This is done at each receiver coordinate for the windowed gathers in Figure 6-15. The peak of this correlation is plotted with the red line and this is used as the phase of the reflection. At some receiver coordinates we observe overall lower correlation values	217
6-17	a): Comparing the selected phases from Figure 6-16. Intersecting events cause abrupt deviations from a smooth phase transition. We therefore choose to mute the gray regions. b):Plotting the difference with the 'true' phase for all simulations in Figure 6-17a. The phase difference is corrected for phase wrapping. Average misfits in the non-muted regions: Local A: 10.07° , Local B: 6.69° , Local C: 9.84° and Local D: 2.67°	218
6-18	Normalized model evolution during inversion. '1' refers to the potentially hydrocarbon bearing anomaly. '2' refers to the bulk chalk. . .	220

6-19	Phase for source at $x = 2.5$ km for selected inversion iterations. The grayed out region is muted and the phase does not influence the inversion. Figure a) shows phase and b) shows the difference (i.e. residual).	222
6-20	Log plot of objective value summing contributions from residual curves like Figures 6-19b for all sources to get objective value (6.16). The black dashed line is the objective value from the true model using Local D.	223
A-1	Zoomed in section of the black dotted box in Figure 2-1.	237
A-2	(a) The summation of the first line in (A.10) results in weighted normal derivatives indicated by orange arrows. The left blue box represents a column not intersecting C . Summation over this column results in weighted normal derivatives at the top and bottom of the full mesh. The two smaller blue boxes represent subcolumns whose summation also result in weighted normal derivatives at the endpoints. (b) Additional summation over the second line adds the horizontal normal derivatives to the vertical ones of a . The contribution on the outer boundary adds up to zero and can therefore be neglected.	238
C-1	Velocity update directions corresponding to the squared slowness gradients on the CDA data with m_0 corresponding to Figure 5-4c. Figures C-1a, C-1b and C-1c correspond to the gradients g_0 , g_1 and δg defined in (C.1), (C.2) and (C.6) respectively.	248
C-2	Velocity steps taken in Parallel inversion and Joint inversion with velocity update directions corresponding to g_0 , g_1 and δg given in Figures C-1a, C-1b and C-1c respectively.	249

D-1	Workflow of the proposed algorithm. Step one computes CDA Green’s functions, which is the only time we use a full domain solver. Step two uses injection scaling factor (D.2) and (D.3). Step four uses (D.4) and (D.5) for propagating the wavefield to the receivers. Step five scales the background wavefield to compensate for differences in V_s and ρ between the source and receiver location, as defined in (D.6). Finally step six scales the wavefield for the shear velocity at the source location with $l_{V_s}^{inj}(\mathbf{x}_s)$	255
F-1	Boundary integrand of (6.15) for Local C for a source at $x = 2.5$ km and receiver at $x = 8.0$ km. From left to right, the three vertical dashed blue lines represent the transition from the top→right, right→bottom and bottom→left boundary. Node numbering is also clockwise. The red box is a zoom region.	264
F-2	(a) shows a zoom in on the red box in Figure F-1. The dashed blue lines are the same. The two vertical arrows indicate locations where the boundary integral passes the reflector, which is fully elastic in Local C since there is no taper. Notice the spiky features at the vertical arrows. When summing over all nodes along the horizontal magenta line we obtain the scattered field at $t = 4.85$ s. (b) is similar to (a), but now for Local D. The boundary node numbers are slightly larger because a horizontal taper is now used. When the boundary integral crosses the top of the reflector at the left arrow we observe that the transition in the integrand is very smooth. Because of this no scatter artifact is generated like we saw in Local C.	265

List of Tables

3.1	This table compares the computational cost of using the local solver and the full-domain solver. We use a speed-up factor of 13, which is what we observe in the example shown here, i.e. local solves are assumed to cost 1/13 of a full-domain solve. The computational cost of the local FWI includes both the pre-computation of (approximately 1225) full-domain Green's functions (done once) and subsequent local solves (assuming 3.5 wave solves per source including the line search, for 60 sources this gives 210 wave solves per iteration). We assume that the same number of iterations are used for each algorithm. The third column shows the ratio of the time for the local version of FWI to perform the inversion to that of the full-domain solve.	92
6.1	The elastic material parameters for the two-layer example.	202
6.2	All scaling factors used in the experiment defined by Figures 6-4 and 6-5. For this particular experiment they reduce to constants.	205
6.3	Table summarizing the taper properties for the four local solver configurations. Entries for ρ and V_s indicate the taper end value at the boundary. The entry 'N/A' means that the particular taper is not present in this configuration. Local A and Local B taper to the outer red boundary in Figure 6-9. Local C directly injects on the blue box with $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$. Local D only retains the horizontal tapers. Graphically this corresponds to the thin red box connected extending sideways from the blue box in Figure 6-9.	211

6.4	Average phase error (degrees) for each of the four local solver modeling cases for each of the selected sources, not including muted regions due to interacting events. Errors for each source are influenced by the picking of the window and mute ranges. The same windows and mutes for each source are used by all the local solver configurations. The average in the bottom row is just the arithmetic average of the numbers for the individual shots. This does not take into account that not every source has the same number of non-muted phase measurements.	219
6.5	The P-velocity, density and S-velocity for the potentially hydrocarbon bearing anomaly and the bulk chalk. The initial guess is improved significantly during this phase-only inversion.	221

Chapter 1

Introduction

Seismic inversion attempts to minimize the misfit between the observed data and the simulated data on a model representation of the Earth. A classical example of such an approach is ray-based travel time inversion. Multiple implementations are available, targeting for instance direct or diving waves (Lytle and Dines, 1980) or reflected waves (Bishop et al., 1985). What these methods have in common is that they use only a subset of the data, in this case the arrival times of the selected events. In addition, these methods are also based on the ray approximation. This approximation is based on the so-called 'infinite frequency' assumption, which shrinks the first Fresnel zone to an infinitely thin ray. However, the observed data for which the methods invert are finite-frequency and the picked arrival times depend on the velocity away from the raypath as well as that along it. Therefore ray-based inversion methods inherently come with inaccuracies, especially when the true Earth is heterogeneous on the scale of the wave-length.

More accurate inversion methods simulate entire seismic wavefields on a model representation of the Earth. A notable example of such an inversion method is Full Waveform Inversion (FWI), which was introduced in the early 80's by Lailly (1983) and Tarantola (1984). FWI is the procedure of iteratively finding better Earth models m , with synthetic wavefields $u(m)$ which progressively match the observed seismogram d better and reduce objective function χ . Often the least-squares objective function

is used, including all combinations of source s and receiver r .

$$\chi(m) = \frac{1}{2} \sum_{s,r} (d^{s,r} - u^{s,r}(m))^2 \quad (1.1)$$

The advantage of FWI over ray-based tomography methods is two-fold. First, the entire recorded trace is used as data and not just arrival times of a limited number of events. Second, the wavefield simulations remove the infinite frequency assumption with its associated limitations. This allows FWI to invert for more complicated Earth models. After the initial introduction of FWI, further development and applications were slowed due mainly to the high computational costs associated with the wavefield simulations. With increasing computational power, the cost of FWI became manageable on 2D models with a notable example given by Pratt (1999). Another decade later 3D simulations on field data became feasible, see for instance the demonstrations by Plessix (2009) and Vigh et al. (2010). The computational cost of FWI is primarily determined by the efficiency of the wavefield simulations of the forward model. Even though the 3D examples demonstrate the feasibility of FWI, the wavefield simulations remain very expensive and as a result FWI requires large expensive computational clusters.

Due to the high computational cost of (1.1) global optimization methods are not possible and local optimization is necessary. The computational cost of FWI can be broken up into the following components. In order to evaluate the objective function (1.1) for the current model estimate m_i a number of n_s wavefield simulations is required, with n_s representing the number of seismic sources. The next step involves computation of the gradient of (1.1), evaluated at current model estimate m_i . This step requires an additional n_s wavefield simulations to compute the so-called adjoint-wavefields required for the gradient. Depending on the non-linear optimization scheme, the search direction is computed by either directly using the negative gradient or by using it as input to nonlinear-CG or L-BFGS (Nocedal and Wright, 2006a). The next step is to evaluate the objective function (1.1) along this search direction to find a new model m_{i+1} with a lower objective function value. Each of these

objective evaluations along the search direction require a further n_s wavefield simulations. In the most favorable outcome, when only one evaluation along the search direction is required, the cost is $3n_s$ wavefield simulations for each FWI iteration. Then the model is updated and the procedure repeats for the next iteration. A more detailed description of FWI is found in Virieux and Operto (2009) and references therein.

From the analysis above it is clear that the computational cost scales directly with the number of sources and can therefore become quite large, especially in 3D. In several problems, like for instance salt boundary inversion and time-lapse inversion, the model updates are often restricted to a small region of the entire Earth model. In order to form the FWI gradient and update the model in the small region of interest we just need the forward and adjoint wavefields there. Propagation through the unchanged part of the Earth model therefore remains the same. This raises the question if we can design a wavefield solver which makes use of this observation to reduce computational cost. In this thesis we design a local solver which generates exactly the same wavefield as a full domain solver would within the small region of interest, which we refer to as the 'truncated domain'. We then apply this solver to salt boundary inversion and time-lapse inversion. Having established the utility of the local solver for these problems, we then investigate the time-lapse problem and propose a new regularization method.

While studying the impact of the acoustic approximation on the time-lapse problem, we observe the significant phase shift of the elastic post-critical reflections in our study. These phase shifts introduce artifacts in the acoustic time-lapse estimate. We therefore choose to remove the post-critical reflections in the inversion. Even though this significantly improves the acoustic time-lapse estimate, we remove useful information since phase shift depends on the elastic material contrast. An elastic inversion can be used to match the observed phase shifts, but this is computationally very expensive. We therefore investigate a local solver which limits the expensive elastic computations to a small region around the reflectors, where the post-critical phase shifts are generated. This solver is used to efficiently generate local elastic

wavefields at a reflector of interest for Phase VS Angle (PVA) inversion.

1.1 Thesis Outline and Summary of Contributions

The chapters in this thesis are laid out in such a way that they form relatively independent units. Chapters 2-3 form a self-contained unit, so do Chapters 4-5 and Chapter 6 is the final unit. By laying out the thesis in this way the reader can skip ahead to the section of interest with hopefully limited problems. As a result of this layout the reader may find some of the background sections to be redundant.

We first introduce the theory of a numerically exact acoustic local solver. Afterwards we demonstrate its functionality by applying it to the local problems of salt inversion and time-lapse inversion. We then investigate the time-lapse problem in more detail and notice the impact of the elastic phase-shift of post-critical reflections on acoustic time-lapse inversion. The straight-forward approach of muting these post-critical reflections results in a loss of useful information. For this reason we introduce a local elastic solver to efficiently model and invert for these phase shifts. Below we give a brief summary of each chapter of the thesis and the main contributions.

Chapter 2: An efficient numerically exact local solver

In Chapter 2 we introduce an exact acoustic local solver. The local solver shares many similarities with the local solver of van Manen et al. (2007). We significantly improve computational feasibility by implementing the local solver in the frequency domain instead of the time domain. Local solvers usually generate the wavefield in the truncated domain, potentially on a perturbed local model. They then use a boundary integral to propagate the scattered field (i.e. wavefield update) to the receiver locations where they are added to the unchanged background wavefield to form the perturbed wavefield in the locally perturbed model. We choose a finite difference implementation of the local solver, which raises the question of how to discretize the boundary integral. If the goal is to compute exactly the same wavefield as a full

domain simulation would on the locally perturbed model, this choice of discretization is important lest we introduce numerical artifacts by using the local solver. It is not immediately clear how to discretize the corners of the boundary integral and what normal derivative stencil should be used. We naturally answer these questions by deriving a discrete analogue of the boundary integral by starting from the Helmholtz equation and using summation by parts. The local solver generates local wavefields which are up to numerical precision the same as the wavefields generated by a full domain solver and extracted in the truncated domain. By using the properly discretized boundary integral we can propagate wavefield updates to the receiver locations, evaluate FWI residuals and then generate numerically exact adjoint wavefields within the local domain. Since numerically exact forward and adjoint wavefields are now available within the local domain we can construct exactly the same FWI gradient within the local domain as would have been generated with a full domain local solver. The local solver requires computation of a collection of full domain Green's functions before starting the local inversion. After these Green's functions have been generated no further full domain simulations are needed during any part of the inversion. Computation of local wavefields is significantly more efficient than generating full domain wavefields. This significantly reduces the computational time of each individual inversion, which makes the procedure more interactive.

Chapter 3: Application of the numerically exact local solver to problems with a localized nature

In this Chapter we apply the local solver of Chapter 2 to two local inverse problems. In the first problem we invert for the salt boundary using the algorithm proposed by Lewis et al. (2012). It is notoriously hard to image below salt due to its rugosity and high velocity contrast with surrounding sediments. Despite its difficulty, accurate recovery of the salt body is a very important step for sub-salt imaging. We use the local solver to invert for the salt shape within a region of interest. We then demonstrate improvements to sub-salt imaging.

In the second problem we apply the local solver to time-lapse inversion. The hydrocarbon reservoir normally occupies only a small part of the full computational model. When geomechanical changes are limited we expect most time-lapse changes to be restricted to the reservoir area, caused by fluid substitution effects during production. By using the local solver we can limit computation of FWI gradients to the reservoir areas and speed up the inversion accordingly.

Chapter 4: Regularizing velocity differences in time-lapse FWI using gradient mismatch information

In Chapter 4 we continue to investigate time-lapse inversion by introducing a new algorithm for inverting for time-lapse changes. This method belongs to the category of confidence map based time-lapse inversions, pioneered by Yang (2014). Confidence map methods have in common that they perform a preliminary inversion using the baseline and monitor data and determine which regions are more likely to contain time-lapse change. This results in a so-called confidence map, which is used to regularize the model difference in a subsequent joint inversion for the baseline and monitor data. The method we introduce computes the confidence map by performing a preliminary inversion for an intermediate model which matches both the baseline and the monitor data. This intermediate model cannot match baseline and monitor data exactly if there is a physical time-lapse change between two states. Intuitively the method is based on our expectation that the baseline and monitor gradient try to perturb the intermediate model in opposite directions at locations of time-lapse change. The confidence map is constructed by looking at the gradient sign and (illumination compensated) gradient amplitude at each pixel. Locations where the baseline and monitor gradients consistently have opposite sign and the gradient amplitudes are large are considered to have a higher likelihood of being regions of true time-lapse change. The resulting confidence map is used to regularize a subsequent joint inversion. We refer to the method as Gradient Confidence FWI (GC-FWI).

Chapter 5: Evaluating the performance of acoustic FWI time-lapse workflows on realistic synthetic data

Wave propagation through the real Earth is governed by complex physics, including poro-elasticity and anisotropy for instance. These complicated physics leave an imprint on the seismic data recorded in the field. While not complete, an elastic inversion can capture some of this complexity. The elastic simulations required in such an inversion are computationally expensive. As a result a further simplification is often made, the so-called acoustic approximation. In this Chapter we study the effect of the acoustic approximation on both regular FWI and time-lapse FWI. We study this effect by generating three different 'true' datasets with increasingly complex physics, starting from constant density acoustic to variable density acoustic and finally fully elastic. We then invert for these three increasingly complex true datasets using the acoustic approximation. For the time-lapse part of the study we investigate parallel FWI, joint FWI and double difference FWI. We demonstrate on our North Sea synthetic model that the acoustic approximation can cause significant deterioration of the time-lapse inversion when using elastic true data. We demonstrate that the inversion result can be improved by removing the post-critical reflections, whose phase shift we cannot model accurately using an acoustic forward model. We demonstrate that the time-lapse recovery could be improved further with a better baseline model. Finally we also include the confidence map from Chapter 4 to regularize the time-lapse inversion.

Chapter 6: An efficient coupled acoustic-elastic local solver applied to phase inversion

In Chapter 5 we observed that the phase shift of post-critical reflections negatively impacts time-lapse inversion when making the acoustic approximation. We therefore chose to remove them from the inversion. The phase shift of post-critical reflections depends on the elastic material contrast at the reflector. Instead of discarding this

PVA) information it is more desirable to use it in an inversion. Zhu and McMechan (2012) demonstrate that phase has the desirable property that it does not change significantly while the wavefield propagates through the overburden. This contrasts with AVA since the amplitude of a propagating wavefield partitions its energy at each reflector. The most natural forward model for PVA inversion is an elastic wavefield simulator. But since the reflector occupies only a small portion of the full computational domain a local solver can be effective. Robertsson and Chapman (2000) introduce a local elastic solver which accurately models primary reflections within a truncated domain. Their method requires a significant number of elastic full domain simulations to compute the required Green's functions. In this chapter we make use of the observation of Zhu and McMechan (2012) that propagation of a wavefield does not significantly impacts its phase. Instead of using elastic Green's functions we use constant density acoustic Green's functions instead since they are much cheaper to compute. We couple these constant density acoustic Green's functions to a local elastic solver to model the elastic phase shift along a reflector of interest. We then introduce several ways of coupling these different physics together at the boundary of the elastic local domain and show the respective phase errors of the reflections. Finally we invert for material parameters at a reflector by minimizing the misfit between the phase shift of the simulated phase, computed by the new local solver, and the phase shift of the true data.

Chapter 7: Conclusions and Future Directions

In this Chapter we summarize the major contributions of our work and suggest some avenues for future investigation.

Chapter 2

An efficient numerically exact local solver: Theory

Abstract

In a set of problems ranging from 4D seismic to salt boundary estimation, updates to the velocity model often have a highly localized nature. Numerical techniques for these applications such as full-waveform inversion (FWI) require an estimate of the wavefield to compute the model updates. When dealing with localized problems it is wasteful to compute these updates in the global domain, when we only need them in our region of interest. This chapter introduces a local solver that generates forward and adjoint wavefields which are, to machine precision, identical to those generated by a full-domain solver evaluated within the region of interest. This means that the local solver computes all interactions between model updates within the region of interest and the inhomogeneities in the background model outside. Because no approximations are made in the calculation of the forward and adjoint wavefields, the local solver can compute the identical gradient in the region of interest as would be computed by the more expensive full-domain solver.

2.1 Introduction

Historically, first arrivals and primary reflections have been used to construct increasingly complex velocity models. A stratified velocity model can be obtained by

This chapter is based on Willemsen et al. (2016b)

applying Dix’s equation (Dix, 1955) to stacking velocities obtained from common midpoint gathers. Reflection tomography has the potential to give higher resolution velocity models (Stork and Clayton, 1991) from prestack reflection arrivals. Migration Velocity Analysis (MVA) (Brandsberg-Dahl et al., 1999) and Wave Equation Migration Velocity Analysis (WEMVA) (Sava and Biondo, 2004) extract additional information from the reflections by extending the migrated image in the subsurface offset or angle domains.

Although the evolution of these reflection-based methods has resulted in velocity models with increasing resolution, they use only a subset of the recorded seismic data. FWI, on the other hand, views the entire trace as data. FWI has the potential to invert for material properties such as the pressure-wave (P-wave) speed, shear-wave (S-wave) speed and density (Fichtner, 2010). Distributions of these parameters are useful in resource exploration as well as in whole earth imaging. Monitoring changes of these parameters over time is useful in reservoir monitoring, where changing fluid distributions influence the bulk properties of the rock (Arts et al., 2009). The development of the theoretical framework of FWI is often attributed to the work of Lailly (1983) and Tarantola (1984), but the large amount of data contained in the seismic traces initially made application of the method infeasible. At the end of the 90s some small-scale 2D tests were performed, most notably those done by Pratt (1999). With the continuous increase in computational power and algorithmic efficiency, FWI is now often used in the field. An extensive collection of case studies is available, with some recent 3D examples given by Plessix et al. (2010), Prioux et al. (2013), and Bansal et al. (2013). A good overview of FWI, including references to more field applications, is given by Virieux and Operto (2009).

Additional increases in computational power and further improvements in algorithmic efficiency will continuously enable field data applications of FWI using higher frequencies and more realistic physics. The algorithmic improvements of FWI can be grouped into several categories. Some algorithmic studies focus on the inverse problem to improve the convergence speed (Métivier et al., 2013) or on the objective function to make the problem less sensitive to incorrect initial models (van Leeuwen

and Mulder, 2007; Jugnon and Demanet, 2013). Other approaches focus on reducing the number of wavefield simulations required (Krebs et al., 2009; Herrmann et al., 2013). Efficient computation of the wavefields is also studied extensively; for instance, for the frequency domain see the work of Wang et al. (2011) and Zepeda-Núñez and Demanet (2016). van Manen et al. (2005, 2006) show how interferometric methods can be used to compute wavefields efficiently. When there is interest in enhancing the model in a subset of the entire domain, the associated reduction in model size can also be exploited to increase the efficiency. There are at least two different methodologies for this. One is redatuming, where the seismic survey is propagated to the region of interest. This reduces the model space and typically also the amount of data used, leading to improved speeds. The other approach attempts to generate accurate local wavefields for velocity updates (scatterers) in the region of interest. These wavefields can then be used to update the velocity model. The work introduced in this chapter falls into the latter category. In addition, an interesting procedure for doing local inversion in many small domains is introduced by Haffinger (2012). He poses the FWI problem in a novel way that makes possible the subdivision of the global problem into many small local problems. All of the local domains can communicate with each other, but the method can only start from a homogenous initial model.

The idea of seismic redatuming is described by Berryhill (1984). It estimates the measured response between a virtual source and receiver at a datum by trying to remove the effects of the overburden in the data. Schuster and Zhou (2006) give an overview of model- and correlation-based redatuming methods. A model-based redatuming method requires an estimate of the Green's function. An example is given by Dong et al. (2009), who use a subsalt datum to improve a Kirchhoff-migrated image locally. In another example, Yang et al. (2012) use modeled Green's functions between the acquisition surface and a datum to localize a 4D seismic survey. In a different approach, using a modeled direct wave, Brogini et al. (2014) create virtual sources inside the earth at locations where no source or receiver is present. They then use a multidimensional deconvolution approach to approximate the redatumed reflection response. By contrast, correlation-based methods do not require an estimate of the

Green's function (Schuster, 2009). This is especially useful when the overburden above the region of interest is complex. An example of a correlation-based re-datuming approach was given by Byun et al. (2010), in which they cross-correlate recorded data from surface shots at two horizontal wells. In this way, the receivers in one well are replaced by virtual sources with receivers in the other well. The virtual data for waves traveling between the horizontal wells are then used to invert for velocity changes resulting from CO₂ sequestration for instance. The analytical expression for redatuming the sources and receivers requires an integral over the real source and receiver positions (Wiggins, 1984). Real surveys have finite spacing between sources and between receivers, so the integrals turn into summations (Bevc, 1995). The finite sampling of the integrals and the limited source and receiver coverage introduce artifacts in the redatumed data, as was demonstrated by Haffinger (2012). Mulder (2005) expressed the redatuming as an inverse problem where he tried to correct these artifacts while successfully removing most of the effects of the stripped over-burden. Tang and Biondi (2011) generated a local dataset for target-oriented MVA by using a generalized Born modeling approach, which shares similarities with redatuming. What all of these methods have in common is the fact that they modify the surface data recordings. In doing so they introduce varying degrees of artifacts. An ideal approach for local problems would be to compute the wavefield exactly within the region of interest; this wavefield can then be used to update the velocity model locally. This is the problem addressed in this chapter.

Several authors have investigated the computation of such local wavefields for many different model configurations. A special configuration is a model inhomogeneity embedded within an infinite homogeneous medium. By using an exact non-reflecting boundary condition, Teng (2003) generated the wavefield in a subdomain containing a collection of scatterers. His approach was limited to localized scatterers embedded in an infinite homogeneous space where waves exiting the scattering region do not reenter it. Gillman et al. (2013) also investigated the case of a homogeneous medium with inhomogeneities restricted to a bounded subdomain. They computed the wavefield by partitioning the inhomogeneous subdomain into a collection of small

volumes. Impedance-to-impedance maps connect the volumes to their neighbors. The volumes are merged recursively in a tree-like fashion to generate an impedance-to-impedance map for the entire subdomain. This impedance-to-impedance map then connects the subdomain to the homogeneous exterior. Grote and Sim (2011) introduce an efficient method for solving for the interaction between multiple inhomogeneous subdomains embedded in a homogeneous infinite space. The non-reflecting nature of the homogeneous medium gives exact knowledge of the arrival times for waves leaving one scattering domain and arriving at another. This locality in time makes an efficient time domain implementation possible when truncating the computational domain.

The problem with these aforementioned local wavefield generators is that realistic geological media are not homogeneous nor infinite and the background medium will also scatter. Robertsson and Chapman (2000) approximated the wavefield with high accuracy within a locally perturbed subdomain of the full domain and then propagated this to the receivers in the inhomogeneous background model. But their formulation discarded the perturbed wavefield scattering outside of the region of interest and subsequently reentering the subdomain. A time reversal mirror introduced by Masson et al. (2013) made the same approximation and therefore did not generate the exact local wavefield. van Manen et al. (2007) introduced an exact domain truncation method in the time domain that will work for inhomogeneous background models. The key property of this method is that it includes waves that leave the subdomain where the background model can be perturbed, and then scatter an arbitrary number of times in the background media before reentering the subdomain. Including all of these interactions between the subdomain, with its model perturbations, and the inhomogeneous background model, unfortunately results in a loss of locality in time. The computational domain is restricted to the subdomain by enclosing it with a time-dependent Dirichlet boundary condition that exactly matches incoming and outgoing waves. Each node on this time-dependent Dirichlet boundary depends on the wavefield at all the subdomain boundary nodes at all past times. This adds a significant cost to the implementation of the exact nonreflecting boundary condition

in the time domain. Including these multiple interactions between the subdomain and the full domain is essential for physical applications (Vasmel et al., 2013) where waves are expected to exit and re-enter the embedded domain multiple times.

The method that is introduced in this chapter also generates the wavefield exactly in a locally perturbed subdomain. The approach shares similarities with the work of van Manen et al. (2007), but implements a boundary condition around the subdomain in the frequency domain. By using a frequency domain formulation, application of the boundary conditions is much less computationally intensive than the approach of van Manen et al. (2007). The boundary condition matches incoming and outgoing waves exactly and conserves all orders of scattering. Once the local solver is set up, local wavefield generation and inversion is much faster than doing full domain solves. The wavefield in the subdomain is, up to machine precision, the same as the wavefield that is generated by the full domain solver. This makes postprocessing of a full domain FWI procedure intuitive. Identical wavefields and FWI gradients are generated within the subdomain as those that would have been generated by the full domain solver. In this chapter we introduce the theory of the numerically exact local solver. Then in chapter 3 we demonstrate its functionality by applying it to the local inverse problems of salt-boundary inversion and time-lapse seismology.

2.2 Theory

In order to minimize the FWI objective function we need to compute the gradient on the current model estimate. This gradient is computed using full domain forward and adjoint wavefields. In local inverse problems there is a particular interest in updating a subdomain of the full Earth model. From now on we refer to this inversion subdomain as the truncated domain. The most straight-forward way to compute local model updates in the truncated domain is to extract the local gradient from the full domain gradient. This local gradient can then be used to update the model in the truncated domain using a nonlinear optimization algorithm of choice. A computationally more efficient approach of computing the local gradient generates the forward and adjoint

wavefields only in the truncated domain. As long as the local forward and adjoint wavefields are exactly the same in the truncated domain as those generated by a full domain solver, then the local gradient is exact without using expensive full domain simulations. In this section we introduce a local Helmholtz solver that is capable of computing the local forward and adjoint wavefields required for the local gradient. This local Helmholtz solver requires a number of full domain Green's functions in the initial model. The local solver will then generate the exact wavefields when the model changes are restricted to the interior of the truncated domain. This section will explain how this is achieved.

Figure 2-1 illustrates the problem, with the truncated domain enclosed by a red dashed line. The full computational domain Ω is the union of all the domains:

$$\Omega = A \cup B \cup \partial C \cup C. \quad (2.1)$$

We split the computational grid outside of the truncated domain C into regions A and B . This choice will aid the derivations in Appendix A. As long as the boundary conditions are in region A our derivations will not be influenced by their characteristics. The model perturbations $\delta m(x)$ are restricted to C (i.e., the yellow nodes), which is the interior of the truncated domain, i.e.

$$\text{supp}(\delta m) \subset C, \quad (2.2)$$

where $\text{supp}()$ refers to the support of its argument. The perturbed model $m(x)$ is defined as:

$$m(x) = m_0(x) + \delta m(x), \quad (2.3)$$

where $m_0(x)$ is the background model, which is the initial guess or model in an inversion context. The model is the squared slowness. We define the scattered field as:

$$u_s(x) = u(x) - u_0(x), \quad (2.4)$$

where $u(x)$ is the wavefield in the perturbed model satisfying:

$$-m(x)\omega^2 u(x, \omega) - \Delta_h u(x, \omega) = f(x, \omega), \quad (2.5)$$

with Δ_h the discrete Laplacian with node spacing h . In this chapter a second-order accurate stencil is used. The wavefield in the background model u_0 satisfies:

$$-m_0(x)\omega^2 u_0(x, \omega) - \Delta_h u_0(x, \omega) = f(x, \omega). \quad (2.6)$$

2.2.1 Numerically exact boundary equation for the scattered field

An essential component of the method introduced in this chapter is a numerically exact expression for the scattered field $u_s(x)$ as defined in (2.4) when model perturbations are restricted to C (i.e., the interior of the truncated domain). With this expression we will later truncate the computational domain and efficiently propagate the scattered field $u_s(x)$ to the receiver pixels. For a continuous problem, the expression for $u_s(x)$ is derived from a seismic reciprocity theorem in the form of a boundary integral around the truncated domain (Fokkema and van den Berg, 2013). The boundary integral expresses the scattered field outside the truncated domain in terms of the perturbed field on its boundary and background Green's functions from the boundary to the point where the scattered field is evaluated. Directly discretizing this boundary integral would leave some open questions. Different choices for what to do at the boundary nodes and different normal derivative stencils would result in different $u_s(x)$. Using a discrete derivation and applying summation by parts (Guenther and Lee, 1988) we answer these questions naturally. By doing this the resulting boundary expression returns exactly the same $u_s(x)$ as the simple subtraction of wavefields $u(x)$ and $u_0(x)$ would in the discrete case. This numerical consistency is important because the ultimate goal is to obtain the same wavefield in the truncated domain and at the receivers as would have been obtained had we solved on the full domain. If in an inversion strategy a transition is made from a full domain solver to a

local domain solver, without this accuracy numerical artifacts in the wavefield would cause inversion artifacts. The numerically exact boundary expression for the scattered field prevents such possible complications. We derive an expression for $u_s(x)$, involving nodes on the boundary of the truncated domain, for model perturbations restricted to C . Appendix A gives the derivation of the boundary expression for the scattered field. The resulting expression is:

$$u_s(y, \omega) = - \sum_{x \in \partial C} \frac{1}{h^2} \left(u^{\partial C}(x, \omega) G_0^{\partial C+1}(x, y, \omega) - G_0^{\partial C}(x, y, \omega) u^{\partial C+1}(x, \omega) \right), \quad y \in B \cup \partial C. \quad (2.7)$$

where $u^{\partial C}(x, \omega)$ and $u^{\partial C+1}(x, \omega)$ are the perturbed wavefield on boundary node x and one node to the interior, respectively. The background Green's functions between the node y where the scattered field is evaluated and the node x on the boundary, and one node to the interior are $G_0^{\partial C}(x, y, \omega)$ and $G_0^{\partial C+1}(x, y, \omega)$, respectively. The scattered field at any point $y \in B \cup \partial C$ is expressed in terms of a summation along the entire boundary ∂C , excluding its corner nodes. The wavefield $u^{\partial C}$ on the nodes ∂C is multiplied by the background Green's function from one node to the interior to the node y where the scattered field is computed. The wavefield $u^{\partial C+1}$ one layer to the interior of ∂C is multiplied by the background Green's function from one node to the interior of ∂C to the node y where the scattered field is computed. Equation 2.7 is similar to a typical reciprocity theorem. Details of the derivation and interpretation can be found in Appendix A.

Equation 2.7 is used to assemble a system of equations that solves for the exact wavefield within the truncated domain for the perturbed model m . This means that all the orders of scattering between the inhomogeneities in m_0 outside of the truncated domain and the perturbations δm inside the truncated domain are computed correctly.

2.2.2 Computing the perturbed wavefield in the truncated domain

In this section, we set up a system of equations to compute the perturbed wavefield $u(x, \omega)$ in the truncated domain, $B \cup \partial C$. To do this, we first introduce our node numbering scheme. Then we introduce three concepts that relate the perturbed wavefield in the truncated domain to the perturbed model $m(x)$ and the scattered wavefield $u_s(x, \omega)$. Using the concepts that were introduced earlier we setup the system of equations needed to compute the perturbed wavefield in the truncated domain. We then explain how the solution of this system of equations, which is the perturbed wavefield in the truncated domain, is used to efficiently obtain data residuals at the receivers. Using these data residuals at the receivers and the previously set up system of equations, the FWI gradient is calculated within the truncated domain.

The size of the full domain Ω is n_x by n_z and that of the truncated domain is n'_x by n'_z . The number of boundary nodes n'_b in the truncated domain is, therefore:

$$n'_b = 2(n'_x + n'_z - 2). \quad (2.8)$$

The number of nodes in the truncated domain is n'_n :

$$n'_n = n'_x \times n'_z \quad (2.9)$$

A spiraling node numbering scheme is used in the truncated domain. The first node is the top left node on ∂C . Node numbering for the truncated domain then continues in a counter-clockwise inward spiraling fashion. This is illustrated in Figure 2-1. The scattered field in the truncated domain is found from the following three relations:

List 1.

1. For the nodes on the boundary ∂C , the difference between the wavefield in the perturbed model u and the wavefield in the background model u_0 is the scattered wavefield u_s , i.e. $u_s(x) = u(x) - u_0(x)$.

2. Also at each node on the boundary ∂C , the scattered wavefield $u_s(y, \omega)$ depends on the perturbed wavefield $u(x, \omega)$ at every node of ∂C , excluding the corner nodes. It also depends on $u^{\partial C+1}(x, \omega)$ one layer to the interior of ∂C . This relationship is expressed by Equation (2.7).
3. In the interior of the truncated domain, the Helmholtz equation in model $m(x)$ describes the relationship between the values of the wavefield at adjacent nodes.

Along with the node numbering scheme, the three relations in list 1 are used to assemble a system of equations:

$$\left(\begin{array}{ccc|ccc} -I & I & & & & 0 \\ I & G_0^{\partial C+1} & -G_0^{\partial C} & & & \\ \hline & & & & & \\ \mathbf{0} & & -\mathbf{m}(x)\omega^2 & -\Delta_h & & \end{array} \right) \begin{pmatrix} \mathbf{u}_s^{\partial C} \\ \mathbf{u}^{\partial C} \\ \mathbf{u}^{\partial C+1} \\ \mathbf{u}^{\partial C+2} \\ \vdots \end{pmatrix} = \begin{pmatrix} \mathbf{u}_0^{\partial C} \\ \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \\ \vdots \end{pmatrix}, \quad (2.10)$$

where the two horizontal lines on the right-hand side separate the three relations. The first and the second relation both result in n'_b rows. The third relation, which is the interior Helmholtz equation, results in $(n'_n - n'_b)$ rows, which is the number of interior nodes. The quantities n'_b and n'_n are defined in Equations (2.8) and (2.9) respectively. The unknowns are ordered in a layered fashion because of the spiraling node numbering scheme. The vector $\mathbf{u}_s^{\partial C}$ represents the scattered field for all of the nodes on the boundary ∂C . Similarly, the vectors $\mathbf{u}^{\partial C}$ and $\mathbf{u}^{\partial C+1}$ represent the wavefield in the perturbed model m on ∂C and one layer to the interior, respectively. The superscript $\partial C+2$ represents the nodes two layers to the interior, etc. The matrix in Equation (2.10) is composed of block matrices. The block I is the identity matrix and the blocks $G_0^{\partial C}$ and $G_0^{\partial C+1}$ represent the background Green's functions, from ∂C and one layer to the interior, respectively, to the node in ∂C where the scattered field is evaluated. The block $G_0^{\partial C+1}$ contains the functions $G_0^{\partial C+1}(x, y, \omega)$ from Equation (2.7) and the block $G_0^{\partial C}$ contains the functions $G_0^{\partial C}(x, y, \omega)$, where y is now the boundary node where the scattered field is computed. These Green's function block

matrices are the only dense blocks. They are several times larger than the size of the interior Helmholtz operator in computer memory. $G_0^{\partial C+1}$ is of shape n'_b by n'_b and $G_0^{\partial C}$ is n'_b by $(n'_b - 8)$.

The square matrix of size $n'_n + n'_b$ in Equation 2.10 is much smaller than the square full domain Helmholtz matrix of size $n_n = n_x \times n_z$. There are far fewer unknowns to solve for, which makes obtaining the truncated wavefield very fast. For the cases investigated in this chapter, the speedup factor is between 10 and 30, but the exact number depends entirely on the size of the truncated domain compared to the size of the full domain. More details on the cost and benefits can be found in the Discussion section.

The wavefield u in the truncated domain is obtained by solving the much smaller system of equations (i.e., Equation (2.10)). This wavefield matches exactly what would have been generated by the full domain Helmholtz solver, as long as the model perturbations are restricted to the interior of the truncated domain. When we say exact, we mean that the relative difference between the two solutions is on the order of $\epsilon \sim 10^{-12} - 10^{-15}$ when double precision is used, independent of the shape and amplitude of the model perturbation. We define relative difference as

$$\epsilon = \frac{|u_f - u_t|_2}{|u_f|_2}, \quad (2.11)$$

where u_f is the wavefield generated by a full domain solver, u_t is the wavefield from the truncated domain solver generated by solving Equation (2.10) and $|\cdot|_2$ is the least squares norm. This shows that the only difference is caused by rounding errors. All the orders of scattering between local perturbations in the model and inhomogeneities outside of the truncated domain are included.

We have now described the computation of the perturbed wavefield in the truncated domain. To compute a model update, either for standard FWI or for the salt updating scheme discussed in this chapter, we need to compute the FWI gradient. To obtain the gradient in the truncated domain, the adjoint wavefield is required in addition to the perturbed wavefield (Virieux and Operto, 2009). The adjoint wave-

field requires evaluation of the data residuals in the perturbed model $m(x)$. Once the wavefield in the truncated domain has been solved for, Equation 2.7 is used to obtain the scattered field at the receivers. The cost of computing the scattered field at all the receiver locations is negligible compared to the cost of computing the truncated wavefield. This remains true for a streamer acquisition instead of the fixed spread acquisition investigated in this chapter. The precomputed background wavefield u_0 is added to u_s to obtain the perturbed wavefield u at the receiver locations, as defined in Equation (2.4). In this way the data residuals and the objective function for the locally perturbed model $m(x)$ can be evaluated exactly, using only local solves. With the numerically exact residuals available, the adjoint wavefield can be specified. For the least-squares misfit function

$$\chi = \frac{1}{2} \sum_{s,r} (u^{s,r} - d^{s,r}) \overline{(u^{s,r} - d^{s,r})}, \quad (2.12)$$

the adjoint wavefield in the frequency domain for a specific shot s is defined as:

$$\beta^s(x, \omega) = \sum_{i=1}^{n_r} \overline{G(x, x_r, \omega)} (u^{s,r} - d^{s,r}), \quad (2.13)$$

where the bar is complex conjugation and n_r the number of receivers. The superscripts s and r are source and receiver indexes. The wavefield u in the truncated domain is already computed through Equation (2.10) and it is desirable to use the same system of equations to solve for the adjoint wavefield as well. This is achieved by realizing that the complex conjugate of Equation (2.13) is formed by letting the complex conjugate of the residuals act as source terms, just like standard FWI. The Green's functions that were used to propagate the scattered field from the truncated domain to the receivers are now used to propagate the contribution of each residual source to the boundary ∂C at no additional cost. Each residual source adds a contribution to the right-hand side term $\mathbf{u}_0^{\partial C}$ in Equation (2.10). Solving the system of equations now gives exactly the same complex conjugated adjoint wavefield as would have been generated by a full domain solver. Both the forward and the adjoint wavefields

have now been efficiently computed in the truncated domain and are combined to get exactly the same FWI gradient as a full domain Helmholtz solver would have returned.

2.2.3 Full domain Green’s function precomputation

Before the small system of equations (Equation 2.10) can be solved at high speed, a set of full domain Green’s functions must be computed. The background Green’s functions that are required are:

List 2.

1. Between each source and its corresponding receivers.
2. Between each source and the nodes on ∂C . This gives $\mathbf{u}_0^{\partial C}$ in the right hand side of Equation (2.10) when multiplied by the source wavelet.
3. Between all the node combinations on ∂C . This builds the block $G_0^{\partial C}$
4. Between the nodes on ∂C and one layer to the interior. With this $G_0^{\partial C+1}$ is constructed.
5. Between the receivers and ∂C and one layer to the interior. With these background Green’s functions the scattered component of the solution of Equation (2.10) is propagated to the receiver locations and the adjoint wavefield is sent from the receivers to the truncated domain.

A greedy algorithm based on source-receiver reciprocity is used to compute the five sets of Green’s functions listed above. For every node, a table is set up that keeps track of how many Green’s function connections to other nodes are still left to be computed. The node with the most remaining connections is then used as the source in the next Green’s function calculation until all pairs have been computed. The total number of full domain simulations required is favorable compared to what is required for normal full domain inversions. A cost analysis is given in the Discussion section. Future studies should focus on reducing the required number of full domain solves to further improve efficiency.

2.2.4 Verification of numerical exactness of the local solver

We are now ready to test the numerical accuracy of the local solver. We implement the local solver algorithm in the Python Seismic Inversion Toolbox (PySIT) code (Hewett et al., 2013). Numerical accuracy is tested by generating wavefields on the extended Pluto velocity model in Figure 2-2, which is perturbed from the true model inside the truncated domain indicated by the black box. During a typical local inversion, the model within the truncated domain will change from the initial model on which the Green’s functions were generated. To simulate this, we investigate the general case where the model $m(x)$ in Figure 2-2 differs within the truncated domain from the initial model $m_0(x)$ on which the Green’s functions are generated. The perturbed wavefield in the truncated domain $u(x, \omega)$ is obtained by solving Equation (2.10). Because of the nonzero model perturbation, the solution vector contains a nonzero scattered field vector $\mathbf{u}_s^{\partial C}$.

Perfectly Matched Layers (PMLs) are placed on all four boundaries of Figure 2-2, although this is not required by the local solver. The exactness of the local solver would continue to hold with free surface or other boundary conditions on any of the boundaries. Figure 2-3 (a) shows the real part of the wavefield generated on the velocity model of Figure 2-2 using a full domain Helmholtz solver. Figure 2-3 (b) enlarges the wavefield within the black box, which represents the truncated domain. Figure 2-3 (c) shows the result obtained using the local solver. The wavefields within the truncated region match exactly. The relative difference as defined by Equation (2.11) is $\epsilon \sim 10^{-12} - 10^{-16}$. Even though the model is perturbed from the initial model within the truncated domain, the local solver still matches the full domain wave solve exactly. This means that all of the interactions between the locally perturbed model and the inhomogeneous, but unperturbed, exterior are correctly modeled.

The same accuracy levels are maintained when propagating the solved local wavefield to the receiver locations. Therefore, data residuals are the same as a full domain solver would have obtained up to numerical accuracy. These residuals form sources for the adjoint wavefield. Similar tests show that the adjoint wavefield generated

by the local solver also matches the full domain equivalent with a relative difference $\epsilon \sim 10^{-12} - 10^{-16}$. This shows that the local solver introduces no artifacts in the forward wavefield, objective function evaluation, and adjoint wavefield. Therefore, all the components for a local inversion are available, but now at increased speed.

2.3 Discussion

In this chapter, we have introduced the theory for a numerically exact frequency domain acoustic local solver. We also numerically verified that the local wavefields are computed without error. In Chapter 3 we demonstrate its functionality by applying it to several problems with a localized nature. The local solver requires precomputation of a number Green's functions, which require full domain simulations. All this work is performed off-line once. After these pre-computations are complete, no further full domain simulations are required. Each local simulation is significantly faster than a full domain simulation. Even if the local model is perturbed, the local solver generates exactly the same wavefields as a full domain solver within the truncated domain. All interactions between the local model updates and the inhomogeneous exterior are preserved. Since both the local forward and adjoint wavefields are exact, the local gradient is also exact and obtained with higher efficiency than when using a full domain solver. This local gradient is then used to update the model using a local optimization of choice. The speedup by using a local solver makes the inversion process more interactive. The time required by the inversion is reduced by the local solver speedup factor, compared to the conventional approach of using a full domain solver.

While we derive the fundamental theory using 2D geometries, the idea naturally extends to 3D. A straight-forward application of the idea on 3D geometries scales less favorably, as we now demonstrate. Call N the node dimension of the full domain (including PML nodes). In 2D the full domain is $(N \times N)$ and in 3D it is $(N \times N \times N)$. In 2D the local solver scales desirably. The number of nonzeros in the Helmholtz block of equation 2.10 is $5 \times M^2$ The number of Green's function entries in equation 2.10

is $2 \times (4 \times M)^2 = 2 \times 16M^2 = 32M^2$, where the multiplier '2' materializes because we need the Green's functions from the outer layer to outer layer $(4 \times M)^2$, and from outer to inner layer which is asymptotically the same number. In comparison to the number of nonzeros in the local matrix, the full domain helmholtz matrix has $(5 \times N)^2$ nonzero entries. So for the local solver, the number of matrix entries for local solver relative to the number of matrix entries of the full domain solver is $(37/5) \times (M/N)^2$. For $M/N = 10$ the local solver has 0.074 times the number of entries in the full domain matrix. The model space also reduced from N^2 to M^2 , a factor of 100 for $M/N = 10$.

In 3D the number of entries in the local solver scales in an undesirable way. The number of local solver helmholtz entries is $7 \times M^3$, while the number of local solver Green's function entries is: $2 \times (6 \times M^2)^2 = 72M^4$. Notice the power 4 now, because there local cube has 6 sides with M^2 nodes all interacting with each other. In comparison, the full domain helmholtz matrix has $7 \times N^3$ nonzero entries. So for the local solver, the number of matrix entries for local solver relative to the number of matrix entries of the full domain solver is $(72 \times M/7 + 1) \times (M/N)^3$. Notice that in front of the (M/N) term there is now a term proportional to M . For $M/N = 10$, $M = 100$ and $N = 1000$, the local solver now has 1.02 the number of elements of the full domain helmholtz matrix. The model space was reduced from N^3 to M^3 , which is factor of 1000 for $M/N = 10$. So even though the model space reduces enormously when the local solver is used in 3D, the relative number of matrix entries (local vs full) scales with a factor $(M^4)/(N^3)$, which is not desirable. For this M, N example the number of matrix entries is approximately equal to the full domain version. And the explicit storage of $72 \times M^4$ Green's functions would be 115GB (for $M = 100$) in double precision for each frequency. This quartic storage scaling M^4 is also not desirable.

Given the problems mentioned, two improvements are needed to apply the local solver to 3D. First the number of Green's function computations, each requiring a 3D Helmholtz simulation, needs to be reduced significantly from $72M^4$. Second the storage cost must be brought down. The small storage cost for the local solver in 2D

makes it possible to factorize the matrix (2.10) and use direct methods for efficient computation of multiple solutions. Direct solvers are significantly harder to apply to large matrices such as full domain Helmholtz matrices or the 3D local solver matrix. One potential avenue to reduce the number of Green’s function computations and storage cost is the concept of matrix probing (J.Chiu and L.Demanet, 2012). The idea of matrix probing starts with a suitably chosen collection of basis matrices. Then the matrix of interest, in this case the Green’s function blocks of (2.10), is expressed in this basis. The way this is done is by first computing the interaction of random vector(s) on the matrix of interest (i.e. probing the matrix). The next step is to find the coefficients for the basis matrices to represent the matrix of interest. This is done by setting up a linear system of equations that minimizes the difference between real (i.e. probed) matrix-vector products and those using the representation in the basis. In general the matrix of interest is not in the span of the basis matrices and a perfect representation is not possible.

Each randomized matrix-vector product (i.e. probing) of the Green’s function blocks is implemented with a single full domain simulation. The randomized vector is implemented as a corresponding random force vector on the nodes of the boundary of the local solver. Matrix probing is interesting if the probing approach requires significantly fewer than the $72M^4$ full domain simulations of the straight-forward approach. Whether this is feasible will depend strongly on an appropriate choice of basis matrices. In addition, if these basis matrices are compressible then the storage cost of this approach can also be reduced compared to the straight-forward approach of storing all Green’s functions.

In addition to the complexities extending the local solver to 3D, the local solver introduced in this chapter also evaluates the Laplacian with second order accuracy. In order to reduce dispersion effects, it is important to adapt the algorithm to higher order discretization schemes. Extensions of the local solver to more complicated physics also requires further investigation. We investigate a local elastic solver in chapter 6, but relax the requirement of numerical exactness to improve computational performance.

2.4 Conclusions

In this chapter we introduced a numerically exact local solver. This local solver generates exactly the same wavefield as a full domain solver in a computationally efficient manner. This property is preserved when the model is perturbed in the interior of the truncated domain. All interactions between model perturbations in the interior of the truncated domain and the exterior are included automatically. The local solver computes the local wavefield by solving a relatively small system of equations. The small size of the matrix before and after factorization may allow for a direct solver in situations where the full domain matrix cannot be factorized because of the size of the factors.

2.5 Acknowledgements

First I acknowledge the great collaboration with co-authors Winston Lewis and Alison Malcolm when writing the paper on which this chapter is based (Willemsen et al., 2016b). We acknowledge Marlies Vasmel from ETH Zurich and Laurent Demanet from MIT for insightful discussions. We are thankful to Schlumberger for giving permission to publish this research. In addition we want to thank Johan Robertsson and Paul Williamson for their careful review of- and helpful suggestions for the paper on which this chapter is based (Willemsen et al., 2016b). Finally we would also like to thank MIT and the ERL consortium members for funding and the support of the National Science Foundation under Grant No. DMS-1115406.

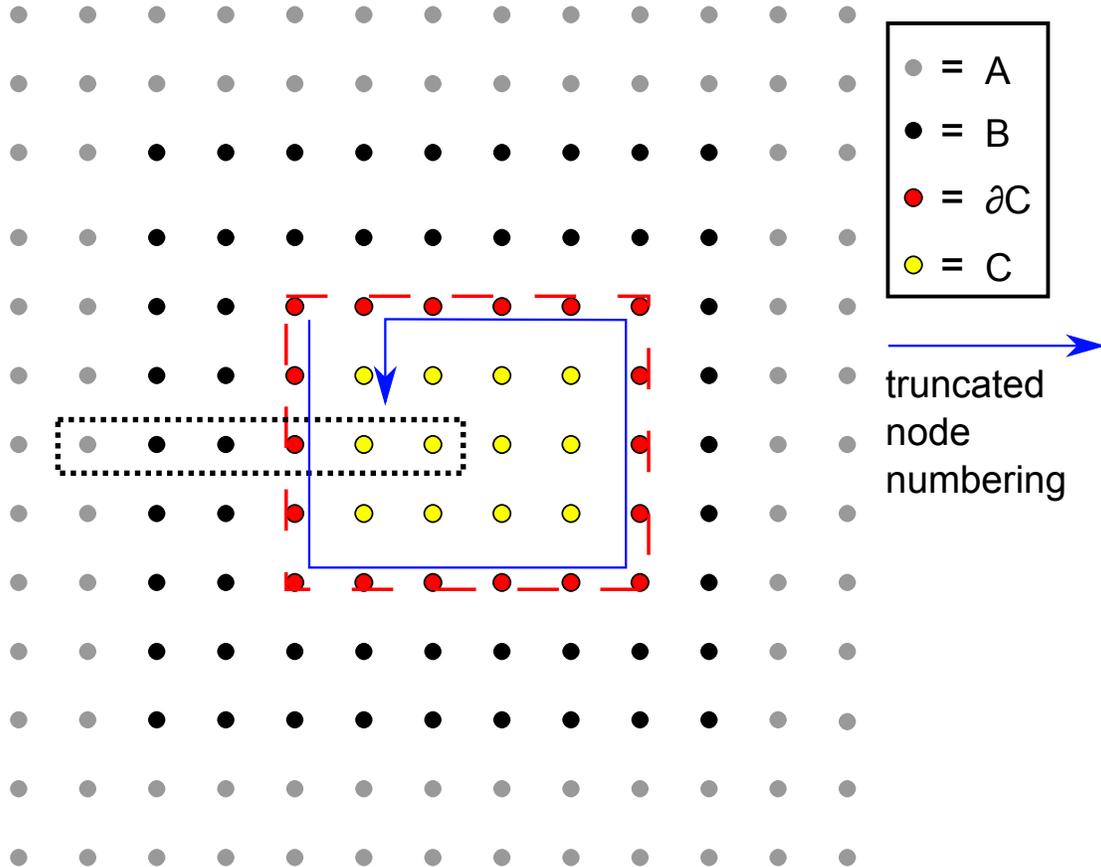


Figure 2-1: The geometrical setup of the problem. Equation 2.1 defines the full computational domain Ω as the union of A , B , ∂C and C . The truncated domain is enclosed by the dashed red line. The nodes on the boundary of the truncated domain are in the set ∂C . The interior nodes of the truncated domain are in the set C . All the source and receiver nodes are in B . The size of zone B is not limited in any direction. It is normally much larger than the size of truncated domain C . The remainder of the computational domain, including the boundaries, is in set A . The nodes enclosed by the dotted black box are used later in this chapter to guide the derivation of the scattered field.

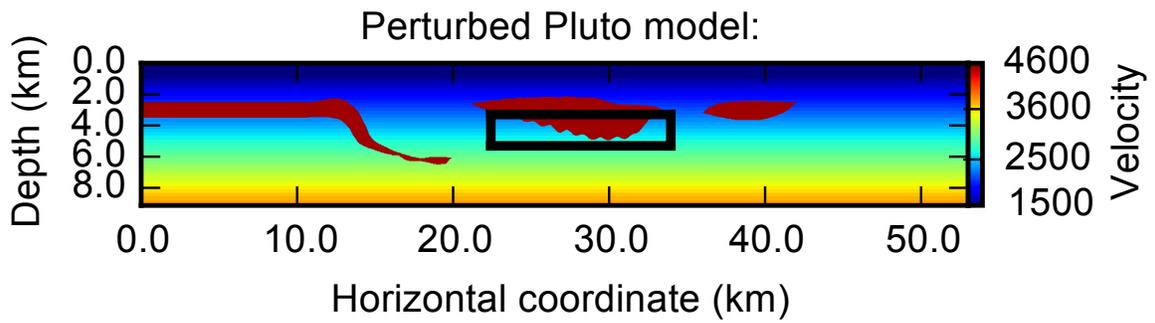
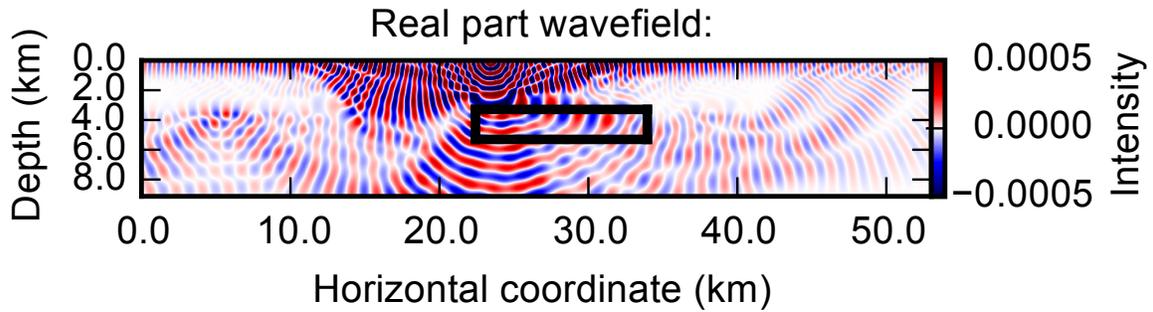
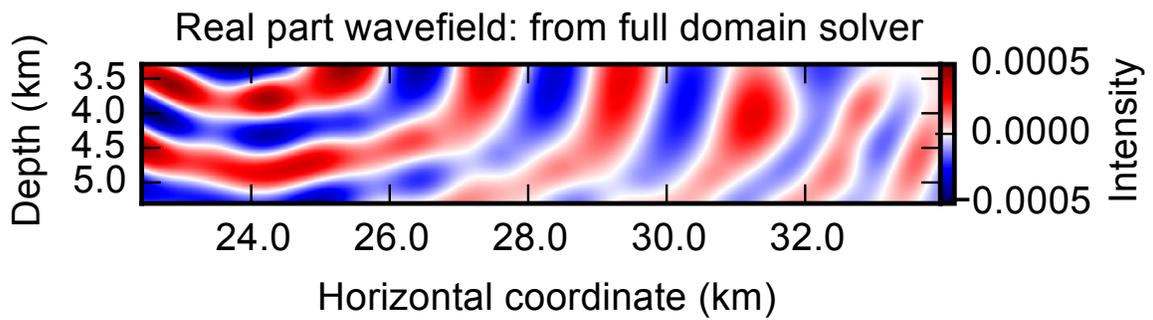


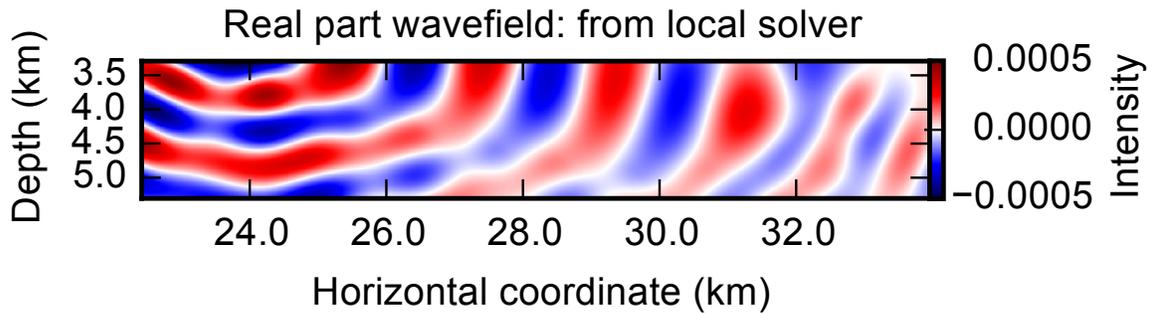
Figure 2-2: The 3480 by 600 nodes velocity model on which the wavefields are generated. The 758 by 133 nodes truncated domain is indicated by the box. Within this box the model is perturbed from the initial model on which the Green's functions are generated, and the wavefields are then compared in this domain.



(a)



(b)



(c)

Figure 2-3: Verification of the local solver using the Pluto model. (a) Real part of the wavefield from the full domain Helmholtz solver. (b) The result from the full domain solver is enlarged within the black box. (c) Real part of the wavefield in the black box from the truncated solver. An equally good fit is obtained for the imaginary part.

Chapter 3

An efficient numerically exact local solver: Verification and application to problems with a localized nature

Abstract

There are many imaging and inversion scenarios in which we have a particular interest in a localized region within the Earth. Two examples of such problems are salt boundary inversion and time-lapse inversion. One way to make use of the local nature of these problems is to use a local solver. We show that the computations can be significantly sped up by using the local solver introduced in the previous chapter. It is well known that an incorrect representation of a salt body in the velocity model significantly deteriorates a subsalt image. The salt body occupies only a subset of the full computational domain. In the first local problem we use the local solver to efficiently generate the FWI gradient at the boundary of a salt body. This gradient is then used in a level set method to automatically update the salt boundary within the truncated domain. We then demonstrate the improvement to the sub-salt image. The second application of the numerically exact local solver is time-lapse inversion. In the absence of significant geomechanical changes, the time-lapse model change is most significant in the reservoirs. We therefore use the numerically local solver to invert for the time-lapse model change in the reservoirs using Double-Difference FWI (DDFWI)

This chapter is based on Willemsen et al. (2015a, 2016b) and Malcolm and Willemsen (2016)

3.1 Introduction

In this chapter we implement the local solver of Chapter 2 into the Python Seismic Inversion Toolbox (PySIT) (Hewett et al., 2013). We apply the local solver to two local problems and therefore split this chapter into two large sections. We first demonstrate the functionality of the local solver by applying it to salt boundary inversion. In the second part of this chapter we apply the local solver the problem of recovering time-lapse velocity change.

3.2 Application to salt boundary inversion

In this section the local solver is applied to the problem of automatically updating the shape of a salt body in the velocity model. The salt body is parametrized in an implicit way using a level set method (Lewis et al., 2012). The work builds on that of Dorn et al. (2000) and references therein who previously used level set methods for shape estimation in geophysical inverse problems. In this method the FWI gradient at the boundary of the salt body is needed to determine the update to the salt shape. This intuitively makes sense because the gradient contains information about whether the velocity estimate needs to be increased or decreased, which corresponds to expanding or contracting the salt respectively. Instead of calculating the gradient in the entire computational domain, the local solver of chapter 2 is used to generate the gradient within a subdomain containing a part of the salt boundary. The salt body is then iteratively updated within this subdomain at high speed. We demonstrate that improving the salt body estimate locally results in a more accurate subsalt image.

3.2.1 Combining the local solver with the salt boundary inversion method

Estimating the shape of the salt body in a velocity model is labor intensive (Etgen et al., 2014) and generally involves the following steps. When an initial estimate is available, a migration is run on the velocity model. The focusing properties of

the migrated image contain information about the quality of the salt body estimate. Interpreters use this information to alter the shape of the salt body in the velocity model. New migrations are run and this process is repeated iteratively until an adequate image is obtained. This process can take months because of the manual guidance that is required and the computational cost of migration. This illustrates the desirability of an automatic method for determining the correct salt geometry. Lewis et al. (2012) introduced a method to do this based on a level set approach. The method requires the FWI gradient at the boundary of the salt as input. Intuitively, the sign of the gradient tells whether the salt boundary should move outward or inward. In addition to the sign of the gradient, the salt updating code also uses the amplitude of the gradient to obtain a search direction for salt body updates. When performing a line search with this search direction, the method returns a new salt body shape for every attempted step length. The salt shape resulting in the lowest value of the objective function is accepted and a new gradient and a new search direction are computed. By iterating over this procedure the salt body estimate is expected to converge toward the true shape.

The salt boundary inversion method requires the FWI gradient only at the boundary of the salt, which makes using a local solver desirable. Instead of computing the full domain gradient and discarding the gradient everywhere except for the salt boundary, we now have the option to compute the gradient exactly and efficiently for this desired region with the local solver. For the local solvers used in the salt updating examples of this section, the speedup factor is between 10 and 30. Larger truncated domains have a smaller speedup factor, while smaller truncated domains have higher speedup factors. The speedup from the local solver makes the inversion much more interactive. For example, instead of having to wait six days, the inversion finishes in approximately 7 hours on a single core. The short feedback time of the local inversions and the reduced computational resource requirements make it possible to investigate many more scenarios. For instance, we can try many different sets of inversion parameters and also different initial salt shapes to investigate the robustness of the inversion. In this study we assume that the sediment velocity is

known accurately enough to focus the gradient with the correct sign at the frequencies we invert for. Even though we invert only for the salt boundary, the gradient could simultaneously update the sediment velocity within the truncated domain as well.

In this section we apply the local solver to the automatic salt updating scheme. The complexity of the salt body inversion is raised incrementally to show its characteristics. A finite difference code is used to generate the data and the background Green's functions. No noise is added to the data.

3.2.2 Application to salt boundary inversion

When the initial model differs from the true model only within the truncated domain, a very good reconstruction is expected for the noiseless case. We investigate this scenario in the extended Pluto model which we introduced in chapter 2 to verify the exactness of the local solver. The initial model is equal to the true model everywhere except for the black box in Figure 2-2. A fixed spread acquisition geometry is simulated with 200 equally spaced sources and 500 equally spaced receivers 40 m below the surface. We perform a frequency sweep with inversion taking place at the frequencies 2.5, 3.0, 3.5, 4.5, 5.5, 7.0, 8.5 and 10.0 Hz. We use the same inversion frequencies in the other salt boundary inversion experiments of this section. For the Pluto example we perform five inversion iterations at each frequency. The FWI gradient is generated within the truncated domain, and this gradient is used by the salt updating code to generate a search direction. For each step length in the line search a different salt geometry is suggested. A very simple line search is used in the inversion in order to minimize the number of evaluations of the objective function. At the start of each inversion frequency, a step length on the higher end of the scale is used. If this results in an increase of the objective function, the step length is cut back drastically for the remaining iterations at that frequency. In this way the number of Helmholtz solutions in most iterations is only three times the number of shots. We do this because generating the gradient costs two times the number of shots; one forward simulation and one adjoint simulation for each shot. Only one step length normally has to be evaluated, at a cost proportional to the number of shots.

The true local velocity of the Pluto model is displayed in Figure 3-1 (a). Notice that the transition between the salt and sediment happens in a staircase manner. The salt updating code will only return models where the transition is a velocity gradient. Therefore, the inversion cannot give a perfect reconstruction of the true velocity model with the staircase transition. The initial velocity model in Figure 3-1 (b) contains an oscillatory perturbation of the salt boundary with an amplitude exceeding 100 m. No updates are performed on the boundary nodes. The local solver only returns the correct wavefield when model updates are limited to the interior nodes, so we set the initial velocity equal to the true velocity on the boundary of the truncated domain. The inversion result obtained after performing the frequency sweep is displayed in Figure 3-1 (c). Because the differences between the true and initial model are restricted to the interior of the truncated domain, a very good inversion result is obtained. The only difference is that the inverted result has a thin "halo" around the boundary of the salt instead of the pixelated staircase pattern of the true model.

In a realistic inversion the initial model will differ from the true model everywhere, and not just in the truncated domain. It is interesting to investigate if the local solver, in combination with the salt updating code, can enhance the salt boundary estimate locally, despite the presence of inaccuracies outside of the truncated domain. To investigate this we use the Sigsbee2b velocity model. The true velocity model is shown in Figure 3-2 (a). The initial velocity model in Figure 3-2 (b) has the true sediment velocity replaced by the Migration Velocity Analysis (MVA) sediment velocity. The top of the salt is correct, but the bottom of the salt is incorrect everywhere and not just within the truncated domain indicated by the black box. Just as in the Pluto model, 200 shots are used, but instead of using all 500 equally spaced receiver locations, only those within a 10 km offset range are used.

We perform three iterations at each inverted frequency. Figure 3-3 shows the true, initial, and inverted model for this scenario. An increased number of inversion iterations does not substantially change the resulting velocity model. Unlike the Pluto test, the initial velocity model now has an incorrect sediment velocity and salt bottom

perturbations with amplitude larger than 100 m outside of the truncated domain. Despite these inaccuracies in the initial model, inversion in the truncated domain still yields a more accurate local velocity model. A thin halo-shaped feature is again present on the inverted model, which is an artifact of the current implementation of the salt updating code. Because the salt boundary is only updated in the truncated domain, a model discontinuity may appear when the salt moves along the truncation boundary. This can be seen in Figure 3-4, where the truncated inverted model is embedded in the unchanged full domain initial model. Because of this it is desirable to let the salt boundary intersect the domain truncation boundary on the sides. This causes a vertical discontinuity that has a smaller effect on the data than a horizontal discontinuity would. A horizontal reflector would have been generated had the salt cut the bottom or top of the truncated domain. In practice the salt boundary should be tapered to avoid the discontinuity that is observed in Figure 3-4. This can be achieved naturally in the framework of the salt updating code by increasing the surface tension near the boundaries of the truncated domain. The current implementation of the code did not yet allow for spatially varying surface tension.

An improved salt bottom estimate has beneficial effects for sub-salt migration. Figure 3-5 shows sections of the migrated images in the region below the truncated domain on the true (a), initial (b), and inverted (c) velocity models, respectively. The migration in the initial velocity model severely distorts the migrated image. It is very hard to find any of the features that are present in the true model. The truncated domain inversion improves the salt bottom estimate locally, which improves the image so that many of the features in the true model can now be seen. The inverted image shows the horst-graben structure and most of the deep point diffractors.

3.2.3 Multi domain inversion

In the previous section the assumption was made that the top of the Sigsbee2b salt body was pixel-perfect in the velocity model. Even though it is much easier to estimate the top of the salt, there will always be slight inaccuracies. These inaccuracies make accurate reconstruction of the bottom salt a difficult task. Inaccuracies in both

the top and the bottom will distort the subsalt image. In a scenario where the migrated image needs to be enhanced in a subsalt region, it is desirable to get a good approximation of the salt body above it. For this purpose, we derive a numerically exact multi domain extension of Equation (2.10). Details can be found in Appendix B. This multi domain solver generates the identical wavefield as a full domain solver would have in multiple disjoint regions, as long as the model is only updated within those regions. This means that all interactions between (model updates in) the local domains and the homogeneous exterior are included. There are no restrictions on the number of domains or on their shape.

Using this multi domain solver, both the top and the bottom of the salt are inverted in order to create a "window" through the salt. The computational cost of the multi domain solver is slightly better than the cost of a single domain with the same number of boundary nodes. The reason for this is that for a fixed number of boundary nodes the single domain contains more nodes and, therefore, more unknowns. For this experiment the bottom is perturbed in a similar way to the previous example. In addition, the top is also perturbed, but less than the bottom to represent that it is easier to obtain the correct top of salt than it is to obtain the correct bottom of salt.

Figure 3-6 shows the initial geometry with the black boxes indicating the two simultaneous inversion domains. Individually, both domains are slightly smaller than the single domain in Figure 3-2 (b), so that the multi domain local solver has a similar speedup factor over the full domain Helmholtz solver as did the single domain local solver in Figure 3-2 (b). The two windows are moved eastwards by a small distance so that they have a relatively large depth separation. In this case the multi domain inversion has a significant advantage over using a big single domain enclosing both boxes on the top and the bottom. Such a big single domain would solve for many unnecessary unknowns, such as the wavefield in the interior of the salt, where no model updates take place.

We perform ten iterations at each inverted frequency. Inversion takes place simultaneously at the top and the bottom. Rather good inversion results are obtained after only five iterations, with slight improvements continually accruing up to ten itera-

tions in this noiseless investigation. Differences in illumination between the top and bottom of the salt can be compensated by applying preconditioning to the gradient (Shin et al., 2001). The modified gradient has similar amplitude above and below the salt, promoting salt boundary updates of similar magnitudes. The improvement in the top domain of Figure 3-6 is seen in Figure 3-7. Just as in the previous inversions, the halo-like feature around the inverted salt body is slightly too wide. Reinitializing the implicit function as it evolves periodically may alleviate this problem. Despite this artifact, the initial model has improved significantly. The inverted topography on the right half is much closer to the true topography. With the exception of the left corner, the salt flank on the left has also improved significantly.

The results of inversion in the bottom domain, which are computed simultaneously with the inversion in the top domain, are shown in Figure 3-8. The bottom domain initially has larger perturbations of the salt boundary than the top domain. The multi domain inversion improves the salt bottom estimate significantly even though the top of salt starts out incorrect everywhere, and is only improved right above the bottom of salt. The interior of the bottom domain sees the greatest improvement, although the sides also move in the correct direction.

Many waves corresponding to long offsets travel through the incorrect top of salt outside of the top window before reaching the bottom truncated domain. Especially at higher frequencies, their contributions to the gradient may be out of phase and will not add constructively. The narrow offsets for sources above the domains will travel through the top domain before reaching the bottom domain. Therefore, they will benefit more from the improved top of salt estimate and are more likely to add constructively to the gradient in the bottom domain. It is worth mentioning that the exact nature of the local solver is very important for this specific problem. Imagine a wave traveling from a source above the domains to the bottom domain and then finally to a receiver above the domains. This wavepath is represented by the ray in Figure 3-6. The wave will first enter the top domain where it experiences the updated velocity model. It then moves down to the bottom domain, where it is reflected. It then reenters the top domain before it moves to the receiver. This primary reflection

on the bottom of salt is influenced by model updates in the top of salt twice. Because no assumptions are made, these multiple interactions between the waves with the perturbations in the top domain are handled correctly. Waves can enter and exit any perturbed domain an arbitrary number of times. Multiples between the top and bottom of salt are influenced by perturbations as they pass through the salt even more times, and the local solver still handles these waves correctly.

Figures 3-9 (a) and 3-9 (b) show migration with the initial and the inverted velocity models. Because now the top of salt also starts out incorrectly and the truncated domains are narrower than in the “bottom only” scenario, the migrated image is less accurate. There is still great improvement, with sections of the layers and the graben structure becoming visible because of the window that is created in the salt by improving the salt boundary on the top and the bottom simultaneously.

Instead of using the multi domain implementation introduced in this section, we can also elect a sequential inversion approach. In this approach we first generate the Green’s functions for the top truncated domain in Figure 3-6. After using this single domain local solver to update the top salt, we would then generate the Green’s functions for the bottom domain in Figure 3-6. The bottom salt is then updated, changing the arrival time of the bottom salt primary reflection in Figure 3-6. If we would want to invert for the top salt again, now that the bottom salt primary reflection has changed significantly, we would have to recompute the Green’s functions used by the top single domain solver. The reason for this is that the salt bottom update changed the model outside of the top truncated domain. Generating wavefields using the single domain solver is slightly faster, because the matrix is smaller. But the advantage of the multi domain solver is that all the offline work of computing the Green’s functions can be done once, before the inversion starts. All orders of multiple interactions between model changes in the different subdomains can also be computed naturally without having to recompute Green’s functions each time a subdomain is changed.

3.2.4 Discussion of the local solver applied to salt inversion

Local inversion may not converge toward the true truncated model, even when the data are noiseless. To see this, assume that the initial model differs from the true model outside of the truncated domain. The model space is constrained to the truncated domain. Reconstruction of the true model within the truncated domain may not coincide with a minimum of the objective function. Interactions between the inverted true truncated model and the incorrect exterior will leave an imprint on the synthetic data that can not be removed with local updates. Other truncated models may match the noiseless data better. Despite this problem, the local inversion experiments on the Sigsbee2b velocity model show significant improvement.

All the inversion results shown started from an initial model that had oscillatory perturbations. The reason why oscillatory perturbations were chosen is that they are easy to visualize in plots. We have investigated other perturbations, such as uniform depth errors in the range of 100 m, and similar results were obtained. As with any least-squares inversion, there is a limit on how far the initial model can differ from the true model, as the inversion can get stuck in a local minimum. There are objective functions that avoid this problem, for example, those based on cross-correlation (Diaz and Sava, 2013).

The motivation for the combination of the local solver with the salt updating code is that inverting the salt shape currently requires much manual work. In some of the examples we tested, the salt shape of the initial model was so far from the correct model that the migrated images contained clear hints of the correct salt shape, as can be seen in Figure 3-5 (b). The migrated image shows semi-circular artifacts on the salt bottom. The perfect top salt in Figure 3-2 would realistically result in a rather accurate initial estimate for the salt bottom through salt-flooding. The reason why an unreasonably bad initial model was chosen was to demonstrate the inversion capabilities of the local salt inversion strategy. The initial model for multi domain inversion in Figure 3-6 is much more realistic. The incorrect top of salt makes it harder to estimate the initial salt bottom. The migrated image in the initial velocity

model in Figure 3-9 still contains hints about the correct salt bottom because the error is still large. However, salt flooding would not have resulted in a close to perfect salt-bottom estimate in this case. We have also performed multi domain inversion on initial models with smaller bottom of salt errors. In that scenario, the migrated image in the incorrect velocity model did not show clear hints about the true salt bottom shape. The multi domain inversion still improved the salt body estimate and the subsalt image.

Cost and benefit

The Green's functions that must be precomputed are described in the Theory of chapter 2. In both of the Sigsbee2b tests, the number of Green's functions that need to be computed at each frequency is approximately 2500, when using reciprocity and a greedy algorithm. These precomputations require an equal number (2500) of full domain Helmholtz solves. The precomputations for all the frequencies can be performed before the frequency sweep starts.

Figure 3-10 schematically compares the work of the local solver with that of using a full domain solver to generate the gradients in the regions of interest. The comparison is made for a single frequency, but the use of multiple frequencies would simply result in a multiplier on both bars. The precomputation for multiple frequencies can be done before the frequency sweep starts. The figure compares the cost of running five inversions with different inversion parameters and initial conditions. Each inversion performs five iterations, and each iteration requires the computation of the gradient and the evaluation of the objective function for at least one step length. Generating the gradient requires two times the number of shots (2 times 200), with the cost split over forward and adjoint simulations. Evaluating the objective function in the line search requires another 200 simulations. If all this work is done using a full domain solver, the total cost would be $5 \times 5 \times 600 = 15,000$ full domain simulations, which somewhat optimistically assumes that only one objective value had to be evaluated for each line search. The computational cost of these 15,000 full domain solves is represented by the large red bar for this conventional approach. The truncated domain

solvers require approximately 2500 full domain simulations to precompute the Green’s functions, represented by the smaller red bar. No full domain simulations are required after the initial setup.

For the Pluto model the local solver is more than 10 times faster than the full domain solver. The local solves in both the single domain and the multi domain Sigsbee2b scenarios are slightly less than 30 times faster than using a full domain solver. The difference in speedup factors is caused by differences in the relative size of the truncated domain(s) and the full domain. In the comparison in Figure 3-10, an intermediate speedup factor of 20 is used. Just like the full domain analysis, 15,000 simulations are performed. But these are now much faster. This is represented by the shorter yellow bar in Figure 3-10. In addition to the decrease in total computation, the local solver approach has the advantage that the time between the start and the end of the inversion is 10 to 30 times shorter. The memory requirements are also substantially lower because of the reduced model space, allowing for more simultaneous inversions on the same hardware. The full domain Helmholtz equation includes PMLs on the sides. It is solved using the direct solver package UMFPACK (Davis, 2004), which is feasible in 2D. The single truncated domain system of equations (Equation (2.10)) and the multi domain version (B.2) are also solved with UMFPACK. The memory required for factorizing the local solver matrix (2.10) is much smaller than for the full domain Helmholtz matrix. This could potentially enable the use of direct solvers in scenarios where they would otherwise fail due to lack of memory. The advantage of a direct solver is that the computational cost of computing the wavefield associated with each source is very small after the factors have been computed.

Similar to the local solver used in this chapter, the time domain local solver of van Manen et al. (2007) also computes the local wavefield exactly. But the computational complexity of the domain truncating boundary condition is less favorable in their implementation. Similar to the frequency domain implementation introduced in chapter 2, the wavefield at each boundary node depends on the wavefield at all the other n'_b boundary nodes. Both methods therefore scale quadratically with the number of boundary nodes. The major difference between the two methods is that the

time domain method also scales quadratically with the number of timesteps n_t . The reason for this scaling is that at each timestep the results of all the previous timesteps are required. This scaling can make the time domain local solver inefficient, because the computational cost of a full domain solver depends only linearly on the number of timesteps. The frequency domain implementation introduced in chapter 2 depends only linearly on the number of frequencies n_f used in the inversion, where n_f is usually also much smaller than n_t . The usual difference between time and frequency domain solvers applies to the computation of the wavefield in the interior of the truncated domain.

The truncated domains in this study have been rectangular, but this is by no means necessary. In fact, the domains can have arbitrary shape. The local solver could have been made even faster for this salt updating application had the salt boundary been followed more closely, reducing the number of unknowns. Figures 3-3 and 3-8 show that a substantial volume of sediment and salt interior are included in the truncated domain, while the salt boundary never moved to many of those locations during the inversion, so the wavefields and gradients were unnecessarily computed there.

For the experiments in this chapter a fixed grid spacing is used during the frequency sweep. This means that an unnecessarily high number of grid points per wavelength is used at low frequencies. If the grid density is decreased at lower frequencies, the number of Green's functions that must be precomputed decreases as well. This reduces the setup cost of the local solver and makes it more favorable.

The local solver introduced in chapter 2 can be applied to other types of local problems as well. In the next section we apply it to time-lapse inversion, where we make use of the observation that the reservoir occupies only a subset of the full Earth model.

3.3 Application to time-lapse inversion

The goal of time-lapse seismology is to recover the change in Earth parameters between an initial (baseline) state and one or more repeat (monitor) states. The conventional time-lapse approach involves comparing baseline and monitor migrated images. This is based on the concept that velocity changes in the Earth change the travel-time to reflectors. These time-shifts in the migrated images are then used to invert for velocity change. Additionally, changes in the amplitude of the reflector between the baseline and monitor state are also used in conventional time-lapse inversion. In the last decade there has been a growing interest in using FWI for the time-lapse problem. A good overview of the time-lapse FWI is presented by Yang (2014) in his Ph.D thesis. In chapter 5 I demonstrate various time-lapse algorithms on a realistic synthetic model. The algorithm we use in combination with the local solver in this chapter is Double Difference FWI (DDFWI).

The DDFWI algorithm is a two-step procedure. The first step is to invert for the baseline model from the baseline data. For this we use the standard FWI (Lailly, 1983; Tarantola, 1984) objective function

$$\chi_0(m_0) = \|\delta d(m_0)\|_2^2 \quad (3.1)$$

$$= \|d_0 - \mathbf{S}u(m_0)\|_2^2, \quad (3.2)$$

where we use the subscript 2 to show we are using the least-squares objective function of the residuals δd . The subscript 0 represents the baseline state of objective function χ_0 , model m_0 and true observed data d_0 . The operator \mathbf{S} samples the simulated data u at the receiver locations for each source. The baseline FWI inversion of equation 3.2 typically updates the initial model everywhere, which is why this step requires a conventional full domain wave solver. The second step of DDFWI is another pass of

FWI using a modified residual δd ,

$$\chi_1(m_1) = \|\delta d(m_1)\|_2^2 \quad (3.3)$$

$$= \|(d_1 - d_0) - (\mathbf{S}u(m_1) - \mathbf{S}u(m_0))\|_2^2 \quad (3.4)$$

$$= \|\tilde{d}_1 - \mathbf{S}u(m_1)\|_2^2. \quad (3.5)$$

The modified residual δd in (3.3) is the difference between the observed data difference and the simulated data difference, as is seen in (3.4). This double difference gives DDFWI its name. The modified residual δd is now a function of only the monitor model m_1 , with baseline model m_0 fixed at the baseline inversion result of (3.2). In equation (3.5) we group these constant terms into a modified monitor data \tilde{d}_1 . This makes clear that the second step of DDFWI is just another FWI inversion, now on the modified monitor data \tilde{d}_1 . The starting model for m_1 is the inverted baseline model m_0 . The time-lapse estimate (i.e. $m_1 - m_0$) is therefore zero at the start of the inversion and changes when we update m_1 to minimize the objective function of (3.5).

Production induced time-lapse change is usually caused by changes in pore fluid and fluid pressure. These fluid changes are typically confined to the reservoir, which occupies only a subset of the Earth model. Changes in pore pressure modify the effective stress and result in volume changes of the reservoir. These effects influence the stress state of the surrounding rocks and therefore its seismic properties (Tura et al., 2005) in what is referred to as geomechanical effects. Outside the reservoir, the Earth model can therefore also change due to production in the reservoir. If the geomechanical effects are known to be small in the reservoir under investigation, then we expect the time-lapse changes to be dominant within the reservoir. In this scenario, the most significant time-lapse change is restricted to the reservoir area and occupies only a subset of the full computational domain. This observation can be exploited to obtain computational savings by using a local solver. In this section we use a synthetic example of this type of time-lapse change to demonstrate the application of the numerically exact local solver to DDFWI.

3.3.1 Problem description

We demonstrate the local solver on the Marmousi II velocity model (Martin et al., 2002). Figure 3-11a shows the true baseline model of this study. Figure 3-11b shows the true time-lapse change, where the black boxes around the three time-lapse regions outline the boundary of the multi-domain local solver. The true monitor model is the true baseline model of Figure 3-11a with the true time-lapse change of Figure 3-11b added to it. The grid spacing of the models is 20.0 m in both the horizontal and vertical directions. With this discretization, the size of the models is 651 by 176 pixels. We first generate the true baseline and monitor data on the baseline and monitor models using a fixed spread acquisition with 60 sources and 651 receivers (i.e. one for each pixel). These sources and receivers cover the surface of the model with equal spacing. The first step of DDFWI is to invert for the baseline model by minimizing the FWI objective function of (3.2). For this we start from the heavily smoothed initial model of Figure 3-12a and do a frequency sweep using a full domain Helmholtz solver. We invert for one frequency at a time, starting at 3.0 Hz and then increasing to 4.0 Hz, 5.0 Hz, 6.5 Hz, 8.0 Hz and finally 10.0 Hz. The final inverted baseline model is plotted in Figure 3-12b. We see that the inverted baseline model of Figure 3-12b contains many of the features of the true baseline model of Figure 3-11a. The shallow part of the model is recovered particularly well. The deeper parts of the model, especially those below the high velocity layers are less accurate. For instance, notice the difference between the true and inverted baseline models in the bottom black box.

The second step of DDFWI uses this inverted baseline model as an initial model for the monitor model inversion which minimizes the DDFWI objective function (3.5). For this step we restrict the model updates to the black boxes of the inverted baseline model in Figure 3-12b. We use the exact multi-domain local solver to efficiently generate the exact local forward and adjoint wavefields required for the local gradient. For this we first compute the required full domain background Green's functions on the background (i.e. inverted baseline) model of Figure 3-12b. We then verify that

the local solver is set up correctly for this multi-domain configuration by comparing the full domain and local domain wavefields within these black boxes. The wavefields plotted in Figure 3-13 are computed on a model with an arbitrary simultaneous model updates in all of the three black boxes. The relative difference as defined in (2.11) between the full domain and the local domain wavefields is approximately 10^{-16} . This verifies that all interactions between the truncated domains (including model updates) and the exterior are computed correctly. The local adjoint wavefields are also numerically exact. With these tests concluded, we now use the local solver to simultaneously invert for time-lapse change within the three truncated domains.

3.3.2 Time-lapse inversion

We invert for the monitor model by minimizing (3.5) while using the inverted baseline model of Figure 3-12b as starting model. We sequentially invert for the frequencies 5.0 Hz, 6.5 Hz, 8.0 Hz and finally 10.0 Hz. The results of this inversion are plotted in Figure 3-14. We first invert for time-lapse change in the entire computational model, using a conventional full domain simulation with 16 iterations at each frequency. We plot the results of this procedure in Figure 3-14a. Even though we only plot the results within the zoom region, the full domain DDFWI inversion updates the model in the entire computational domain. Figure 3-14b shows the results when we follow the same procedure, but reduce the number of iterations to three at each frequency. Finally Figure 3-14c shows the results when we use the local solver and as a consequence also limit model updates to the interior of the black boxes.

First of all we see that both the inversions in the full-domain (Figure 3-14a) and in the truncated domain (Figure 3-14c) are able to recover the time-lapse perturbations (Figure 3-11b) reasonably well. The shape of the bottom perturbation is inaccurate, which we suspect is caused by imperfections in the background model. We conclude this because other parameters (shape, size, velocity contrast) are similar for the other perturbations, but comparing Figures 3-11a and 3-12b illustrates that there are clearly some errors in the deeper parts of our estimated baseline model.

In Figure 3-14 we also compare the number of iterations required for the two

methods to converge. In the full domain many more iterations are required to recover the magnitude of the time-lapse perturbations. Figure 3-14b and 3-14c show the results for three iterations. We observe that the full-domain solution has not correctly estimated the magnitude of the perturbations. In the full domain, we need 16 iterations (Figure 3-14a) to achieve results similar to those obtained in the truncated domain with only three iterations. Although we are not certain why this is the case, we speculate that the reduced model space in the local inversion allows the FWI algorithm to converge more rapidly, thus further reducing the cost of FWI (though a reduction in iterations may also be present when using a full-domain solver if model updates are restricted to a local domain). The benefits of restricting model updates to the reservoir are also reported by Zhang and Huang (2013), who observe improved recovery of the time-lapse change. Care must be taken in using the local solver that the changes across the boundaries are not too abrupt. In this case, if we continue beyond three iterations we will see artifacts caused by jumps in the time-lapse model update. This is easily mitigated using tapers and slightly larger domains.

3.3.3 Discussion

Our goal in using a local solver is to both improve the resulting time-lapse estimates and to reduce computational cost. We compare the cost of a full domain solver with the truncated solver bearing in mind that each method produces different products. The full-domain solver gives a model update throughout the domain while the local solver gives an update only within the truncated domain. To quantify the reduction in computational cost compared to a global solution over the full domain, we must consider two aspects of the process. First, there is the cost of computing the background Green's functions (approximately 1225 in this example). This is done once before the processing begins. Second, there is the cost of computing the forward and adjoint wavefields within the truncated domain. With the number of sources $n_s = 60$, we assume the number of wavefield simulations per iteration is $3.5n_s$. This includes n_s forward simulations, n_s adjoint simulations and an average of 1.5 linesearch trials, adding the final $1.5n_s$ wavefield simulations. Together, these two steps make up the

majority of the computational cost of the algorithm. Although it appears that fewer iterations are required when using the local solver, we choose to compare the costs of the two methods with the same number of iterations to obtain a conservative estimate of the speedup. We show this speedup in Table 3.1, looking at different numbers of inversions and iterations. We consider multiple inversions because this is done when refining the inversion parameters and processing multiple monitor data sets. Once the setup step is done, each iteration is about a factor of 10-15 times faster for the examples shown in this study (the numbers in Table 3.1 use a factor of 13).

For one inversion with one iteration (first row) the local solver takes about six times as long as the full domain solver. By contrast, for twenty inversions with twenty iterations each the local solver takes about nine percent of the time required by the full-domain solver. This difference accumulates because once the offline work generating the background Green’s functions is done, each single wave-solve executes much faster (here 13 times faster). The background work is done only once for the entire process, before the first inversion is initiated. This speedup could lead to more interactive workflows and allow the exploration of more inversion scenarios.

Localization in the model space allows us to consider more expensive time-lapse processing ideas for 4D, such as the alternating FWI suggested in Yang (2014), or cross updating as proposed by Maharramov et al. (2014). This opens up the possibility of truly estimating the uncertainty in 4D changes, which as noted in Asnaashari et al. (2015), is not currently computationally tractable.

3.4 Acknowledgements

The two local solver applications we present in this chapter are based on separate papers. In both studies I benefited from great collaboration with my co-author Alison Malcolm.

For the salt boundary inversion example, Alison and I worked together with co-author Winston Lewis. We appreciate his expertise and insight when writing the paper (Willemsen et al., 2016b). We acknowledge Marlies Vasmel from ETH Zurich and

Laurent Demanet from MIT for insightful discussions. We are thankful to Schlumberger for giving permission to publish this research. In addition we want to thank Johan Robertsson and Paul Williamson for their careful review of- and helpful suggestions for the paper on which the salt-inversion section of the chapter is based (Willemsen et al., 2016b). Finally we would also like to thank MIT and the ERL consortium members for funding and the support of the National Science Foundation under Grant No. DMS-1115406.

The time-lapse application of the numerically exact local solver is based on the paper of Malcolm and Willemsen (2016). Funding for this work comes from MIT as well as from the Natural Science and Engineering Research Council of Canada, the Research and Development Corporation of Newfoundland and Labrador and from Chevron Canada Ltd. We would like to thank our colleagues at MIT and MUN as well as Johan Robertsson and Filippo Brogгинi of ETH for useful discussion. We are also grateful for the work done by David Johnston and two anonymous reviewers of the paper of Malcolm and Willemsen (2016) on which the time-lapse section of the chapter is based on.

3.5 Conclusion

There are many applications in full-waveform inversion in which we attempt to determine Earth properties within a subset of the total region of the subsurface sampled by the data. In these cases, calculating wavefields outside the region of interest at each iteration of inversion is wasted effort. In this chapter, we show that the work required for FWI is reduced by locally computing the wavefield only within the domain of interest. We first combined the local solver with an automatic salt updating code. The locally improved salt boundary estimate resulted in significant improvements in the subsalt migrated image. We then applied the local solver to time-lapse inversion. For time-lapse inversion we observe that in addition to a cheaper forward model, we also require fewer inversion iterations to converge to an accurate time-lapse estimate when using the local solver. As a result, velocity estimates can be improved while

controlling the computational cost. This is of particular interest and importance for 4D seismic applications in which we are interested in an isolated region of the subsurface (the reservoir) and have good knowledge of the properties outside the reservoir, and do not expect these areas to change between surveys.

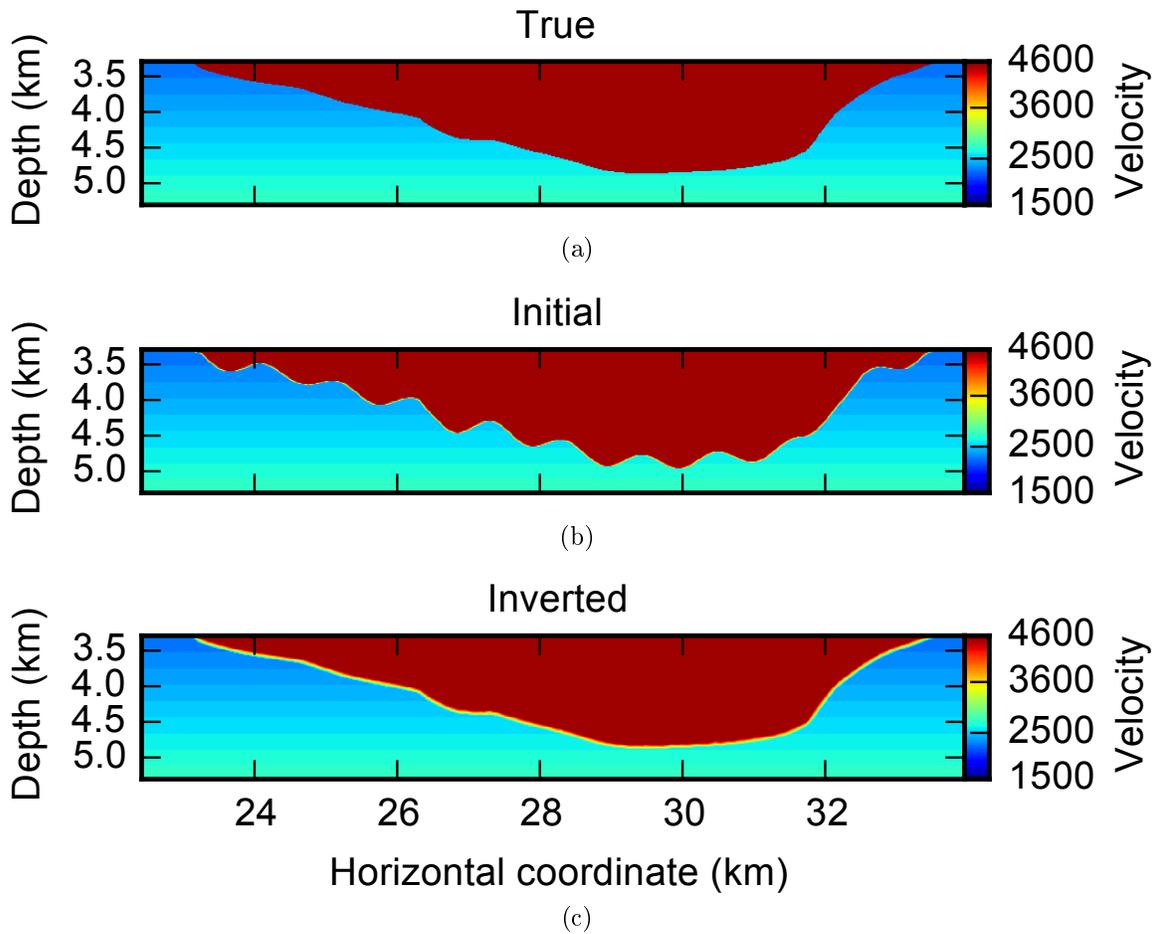


Figure 3-1: Salt boundary inversion results for the Pluto model where the true and initial model only differ within the interior of the truncated domain. (a) True truncated domain. (b) Initial truncated domain. (c) Inverted truncated domain.

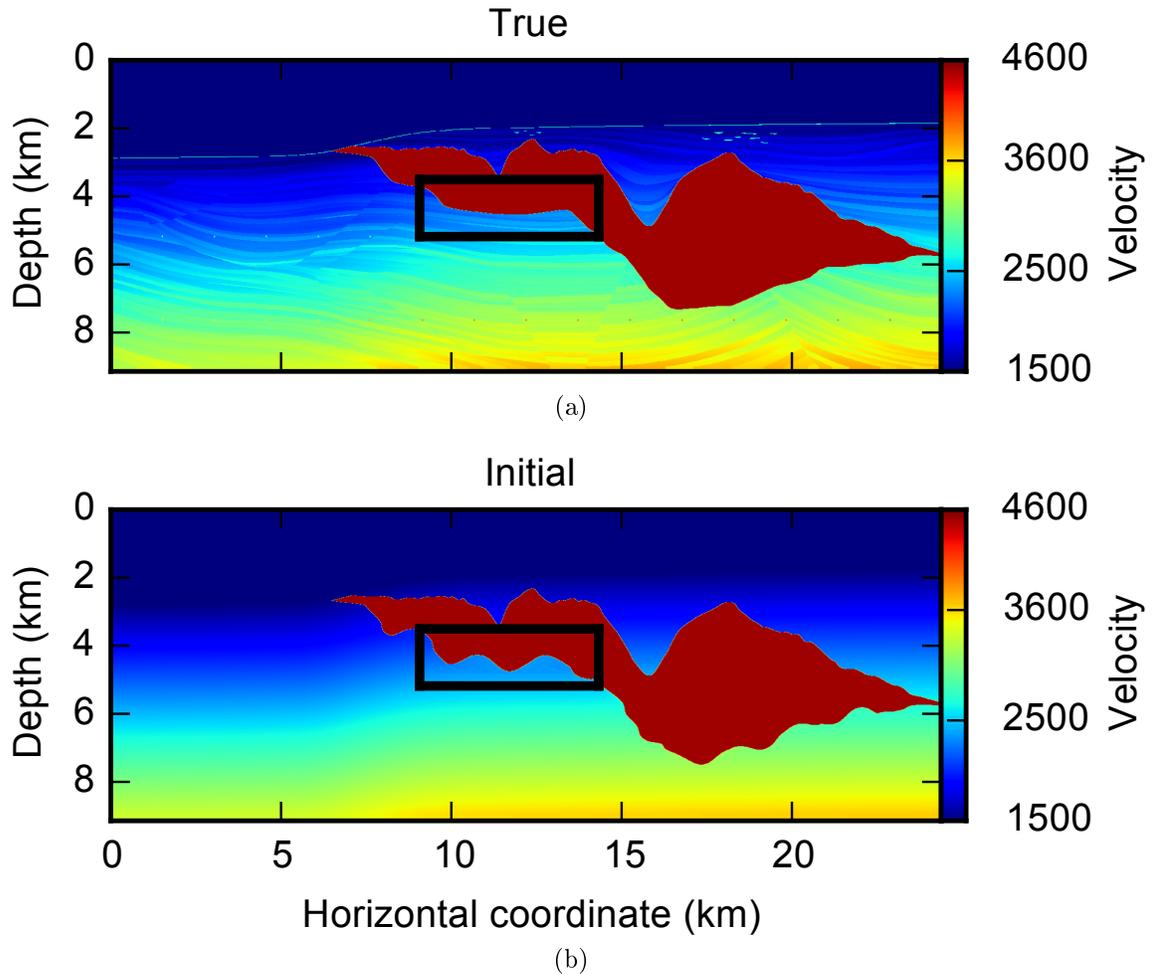


Figure 3-2: (a) True 3201 by 1201 nodes Sigsbee2b velocity model with a 691 by 221 nodes truncated domain. (b) Initial velocity model with MVA sediment and an incorrect salt bottom everywhere, not just inside the truncated domain indicated by the black box. The top of the salt is correct in this example.

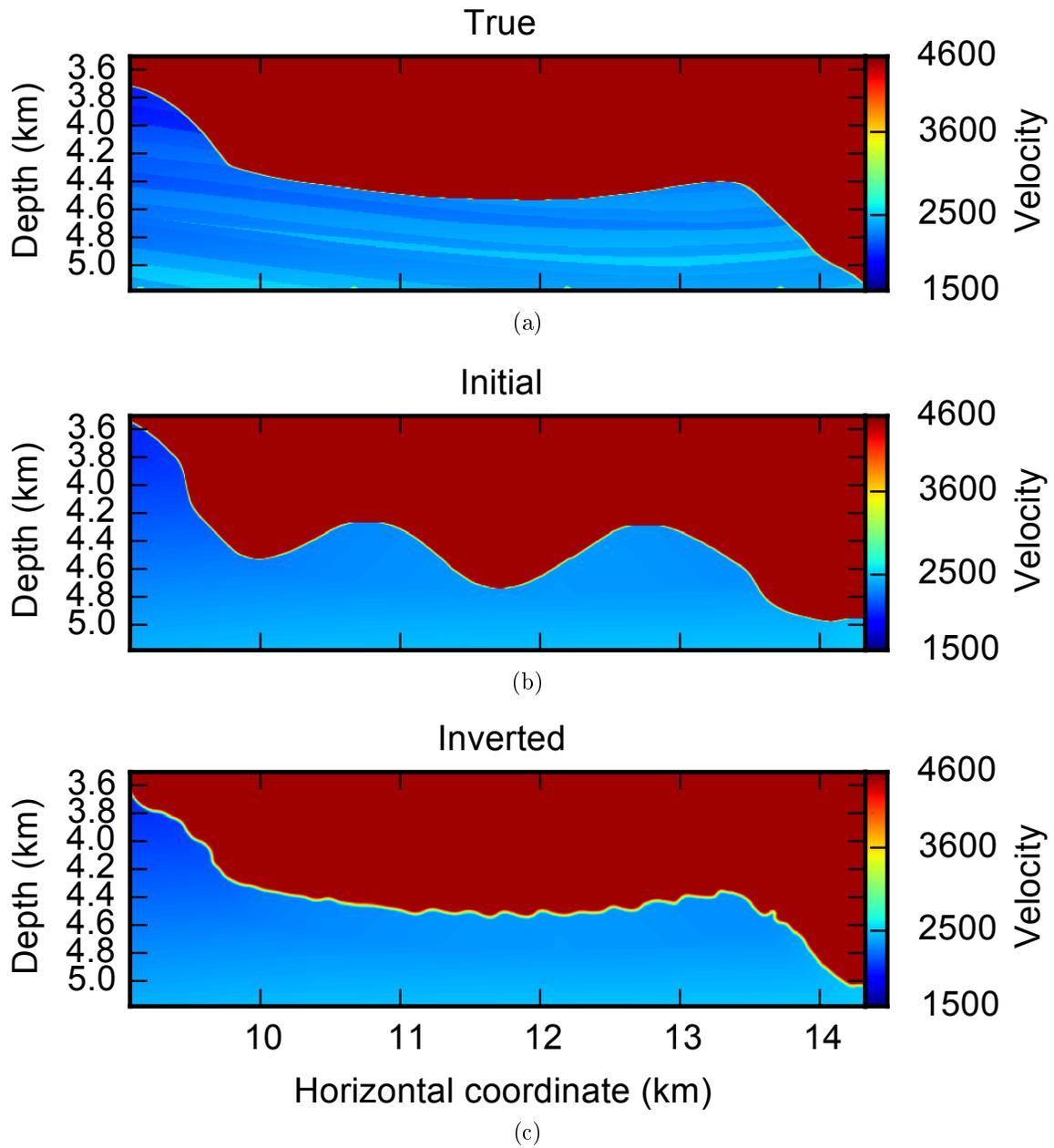


Figure 3-3: (a) True Sigsbee2b truncated velocity model from Figure 3-2a. (b) Initial truncated model from Figure 3-2b. (c) Inverted truncated model from local solver. Significant improvement can be observed.

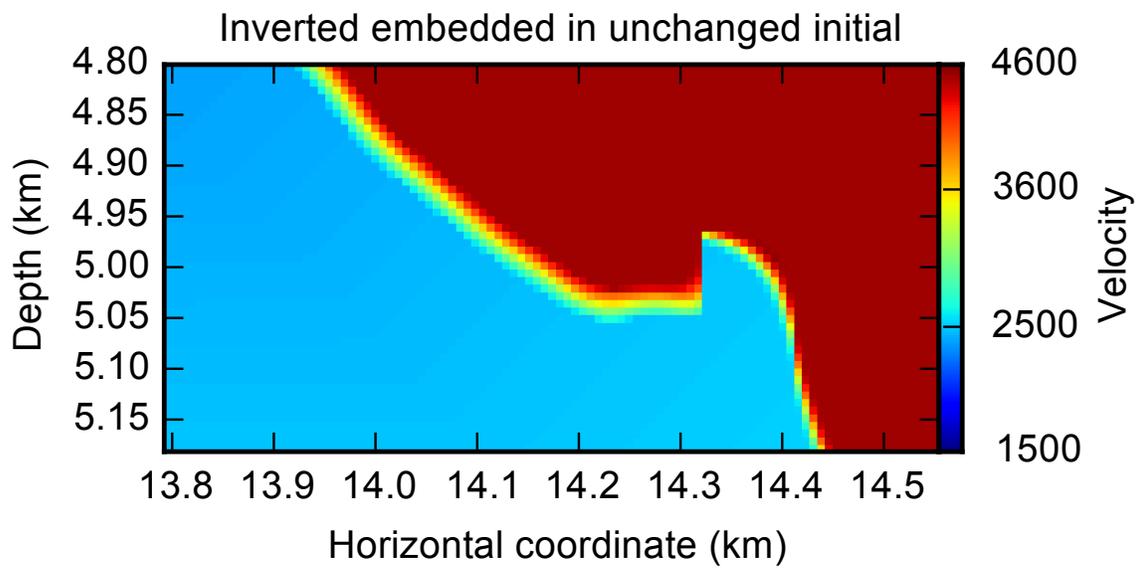


Figure 3-4: Truncated domain inversion result embedded in the unchanged full domain initial model illustrating that abrupt changes can take place at the boundary of the truncated domain.

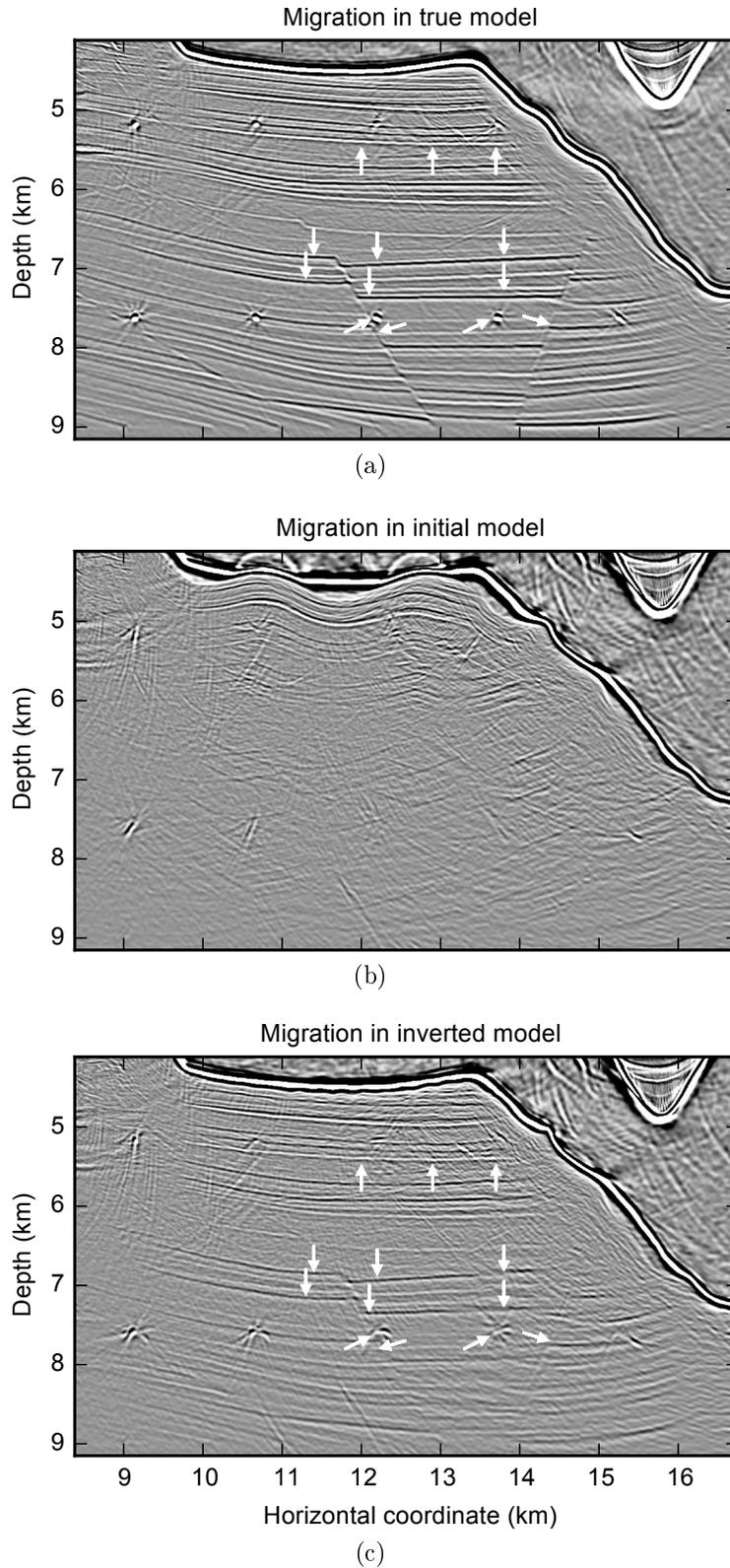


Figure 3-5: Migration in the true, initial and inverted models of Figure 3-3. The initial model uses an MVA sediment velocity and has a very incorrect salt bottom. The arrows show the locations of interesting features that appear after the salt bottom estimate has been improved.

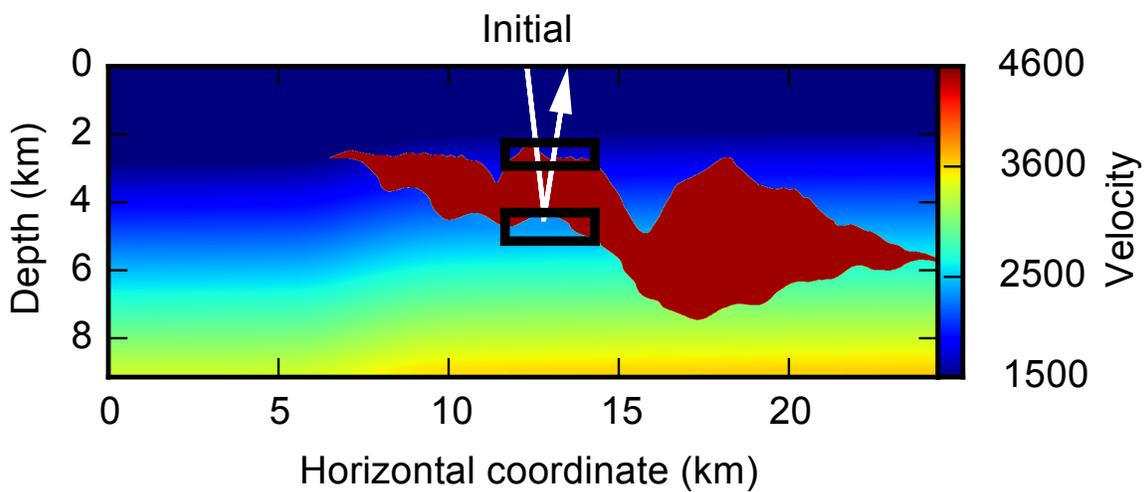


Figure 3-6: The initial model for the multi-domain inversion. The top truncated domain is 347 by 91 nodes and the bottom domain is 347 by 111 nodes. The true model remains the same, shown in Figure 3-2. The white ray shows that a primary reflection on the salt bottom is influenced by the top domain twice. This demonstrates that it is important that the local solver correctly accounts for waves reentering the truncated domain. Waves reflecting multiple times benefit from this even more.

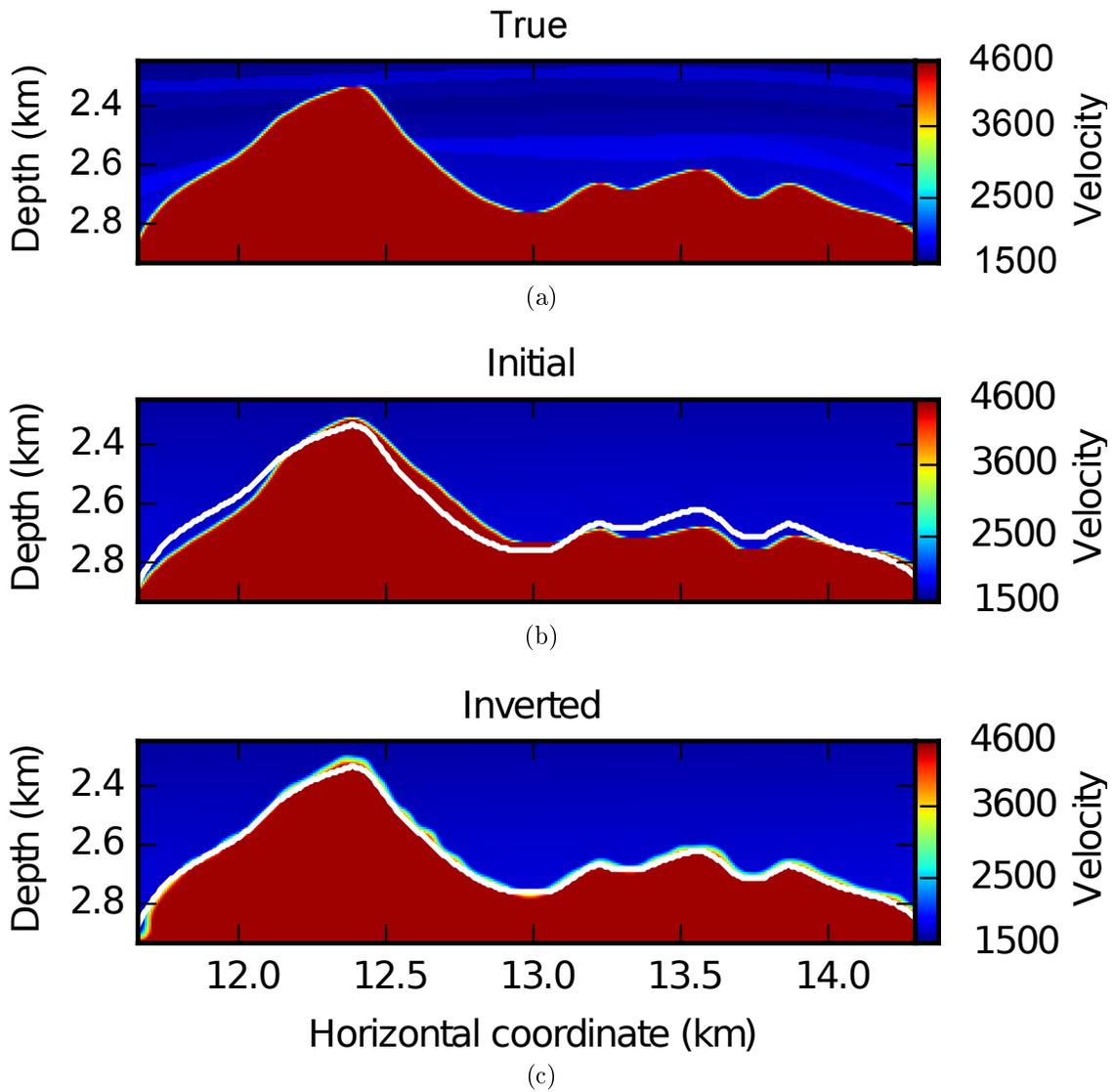


Figure 3-7: This figure shows the improvement in the top salt estimate. Inversion takes place simultaneously in both of the truncated domains in Figure 3-6. The initial and true model differ significantly everywhere. The white line shows the true salt boundary.

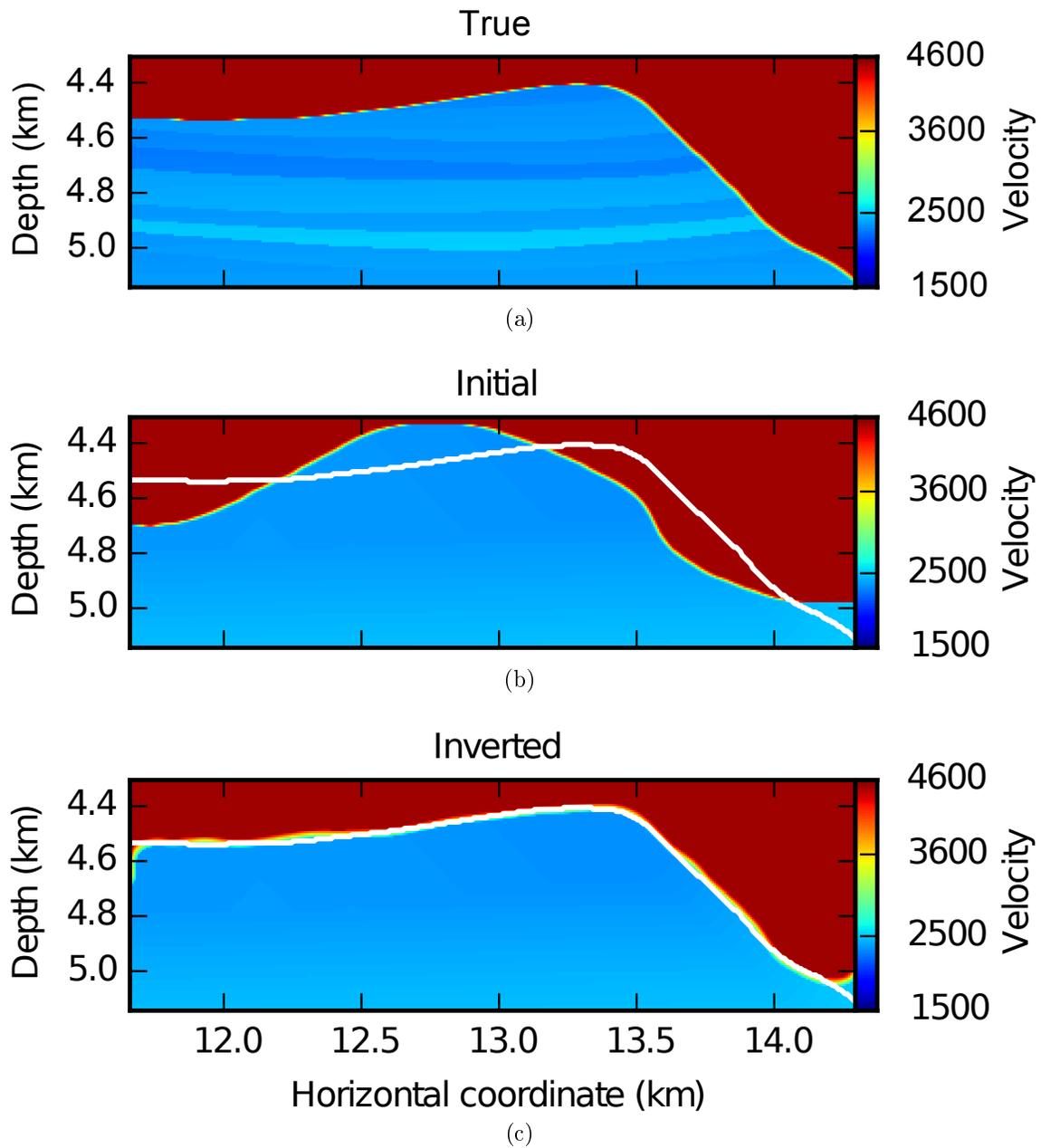


Figure 3-8: This figure shows the improvement in the bottom salt estimate. Inversion takes place simultaneously in both of the truncated domains in Figure 3-6. The initial and true model differ significantly everywhere. The white line shows the true salt boundary.

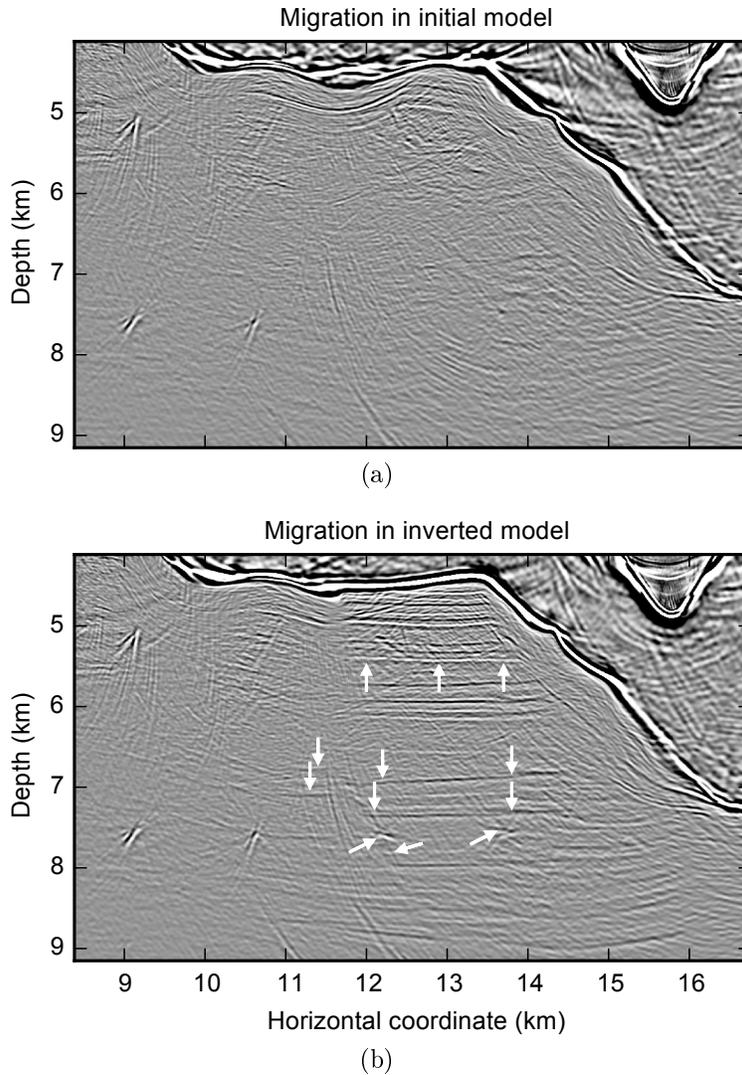


Figure 3-9: Migration artifacts around the salt bottom in the initial model still give an indication of the correct salt shape. Less distorted initial models have been investigated where the correct salt bottom was not recognizable from migrated images, but simultaneous inversion still improved the salt bottom estimate. The large salt perturbations in Figure 3-7b and Figure 3-8b make it easier to visualize improvements in the velocity model and the migrated image. Arrows show interesting features. The location of these arrows is slightly perturbed from those in the true model in Figure 3-5. But they show the same geological features.

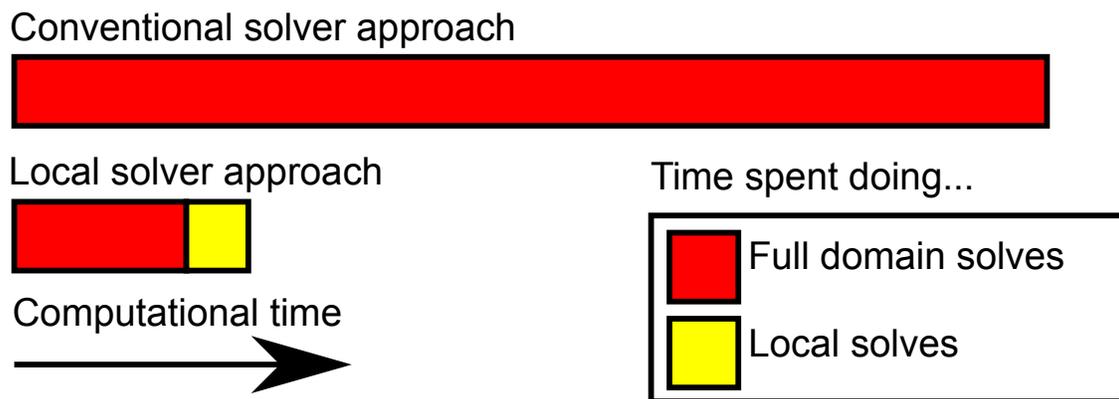
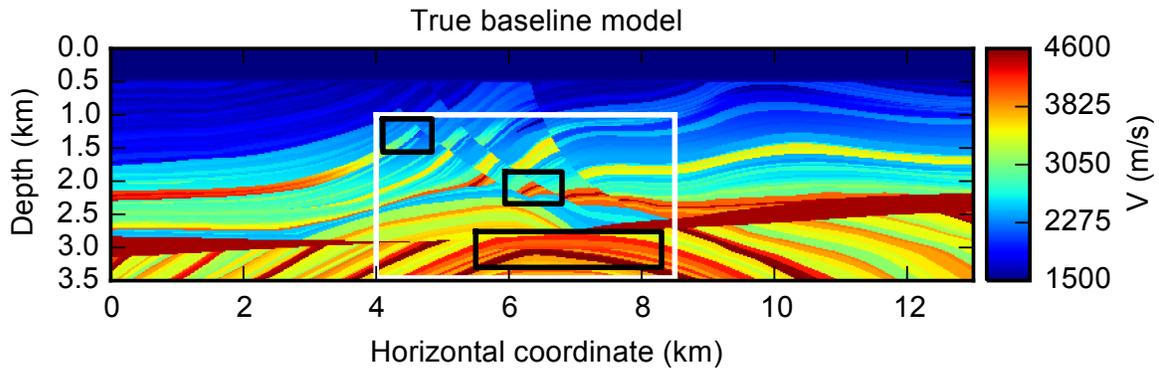
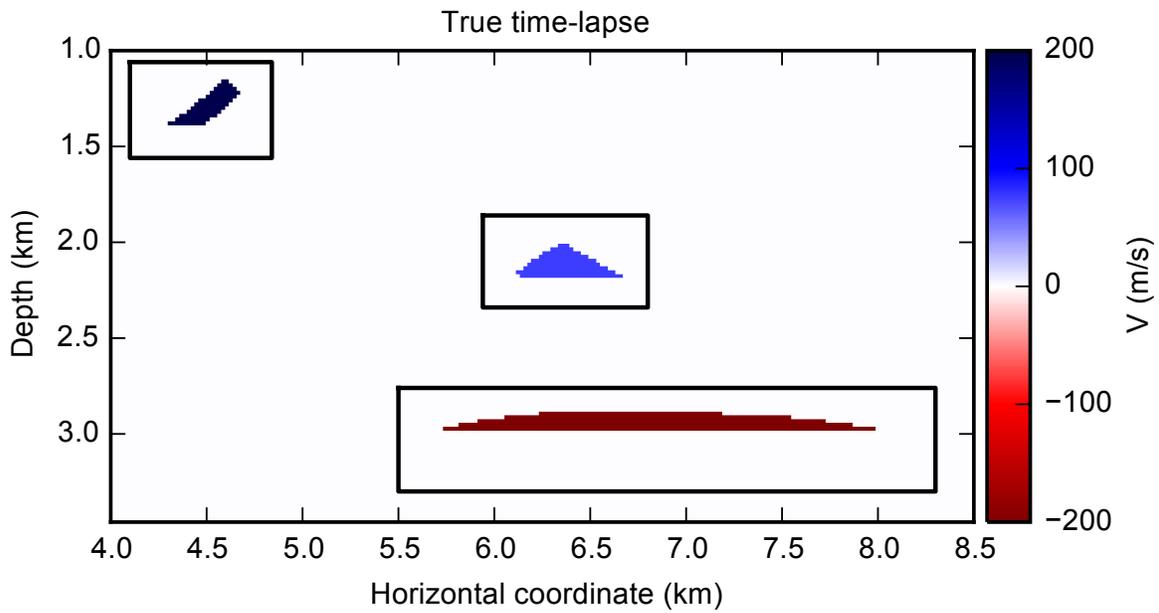


Figure 3-10: Schematic cost comparison for running 5 simultaneous inversions using different inversion parameters and initial conditions. The length of the bars is to scale.

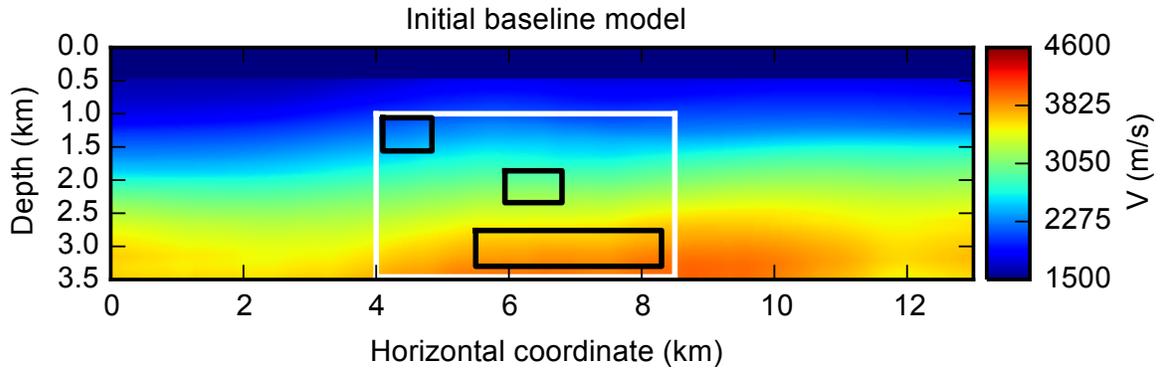


(a)

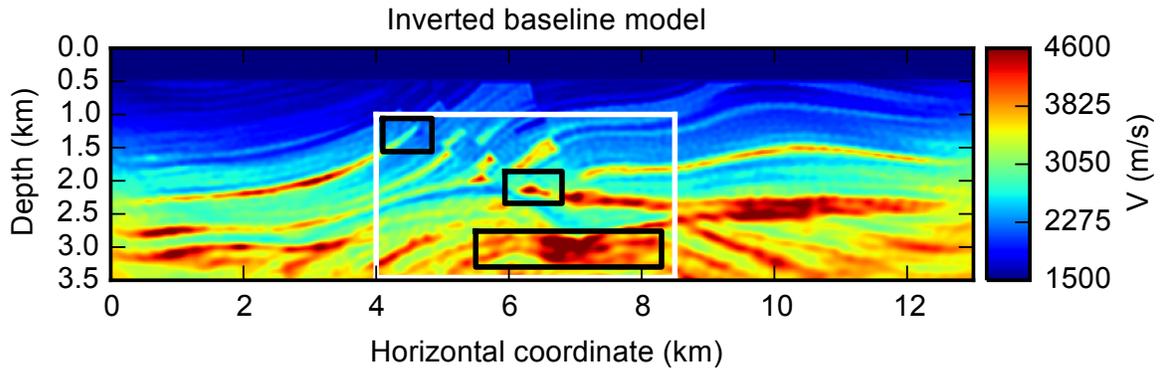


(b)

Figure 3-11: Figure 3-11a shows the true Marmousi II baseline model used in this study. Figure 3-11b zooms in on the white box of Figure 3-11a to show the true time-lapse change in the three reservoirs. The black boxes surrounding the reservoirs outline the multi-domain local solver.

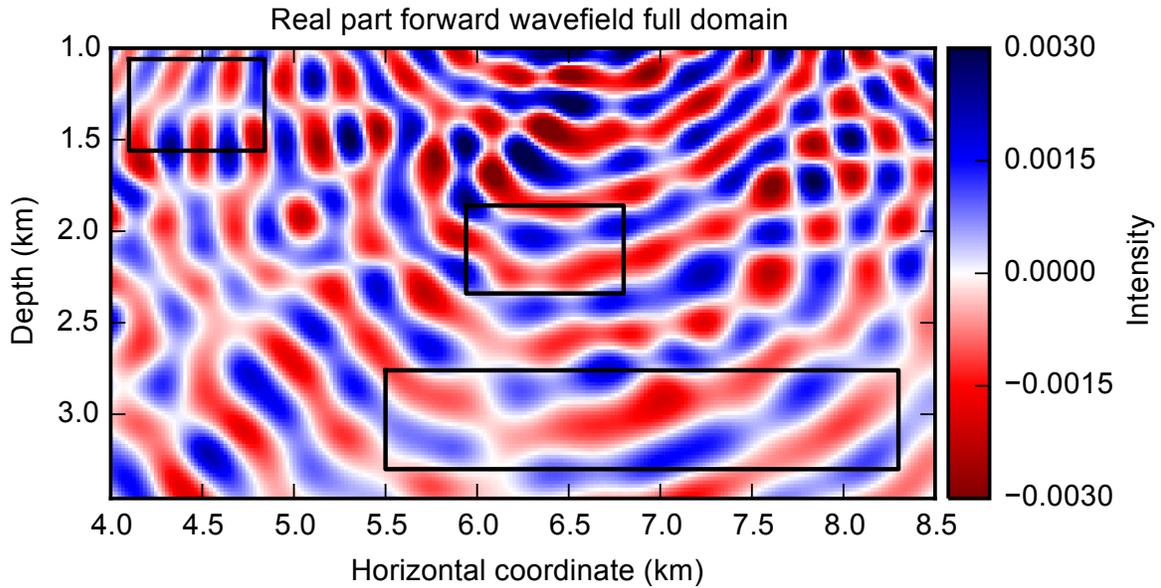


(a)

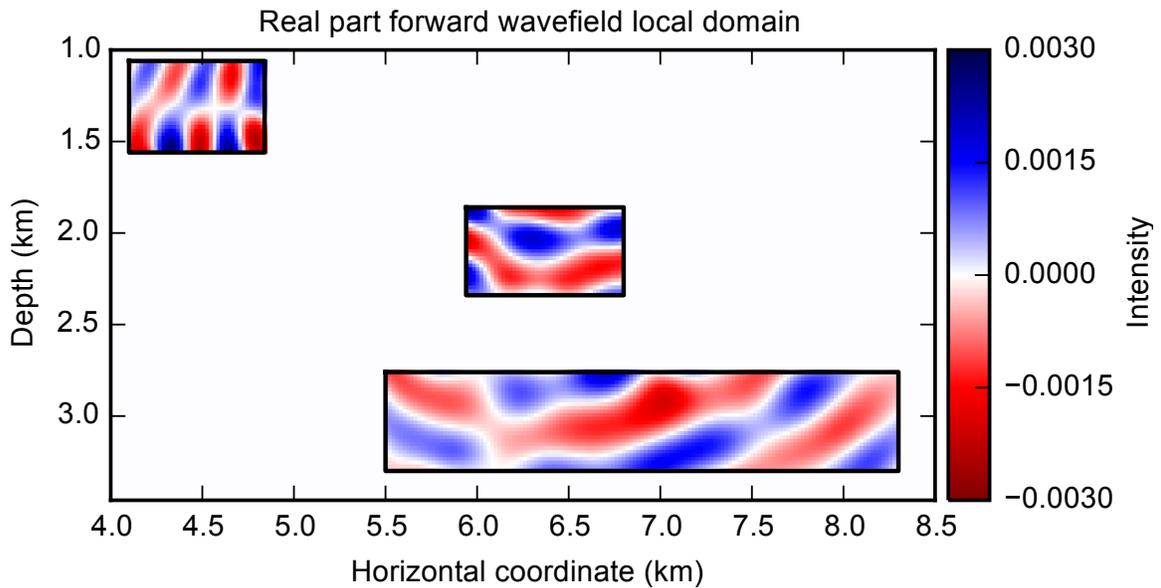


(b)

Figure 3-12: Figure 3-12a shows the initial baseline model. Figure 3-12b shows the inverted baseline obtained at the end of a frequency sweep. The sweep inverts for one frequency at a time, starting at 3.0 Hz and subsequently inverting for 4.0 Hz, 5.0 Hz, 6.5 Hz, 8.0 Hz and finally 10.0 Hz. The black boxes surrounding the reservoirs outline the multi-domain local solver and are identical to those in Figure 3-11a.

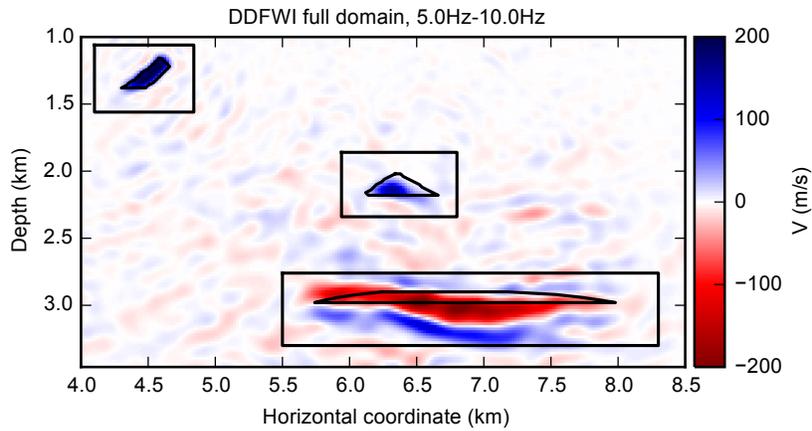


(a)

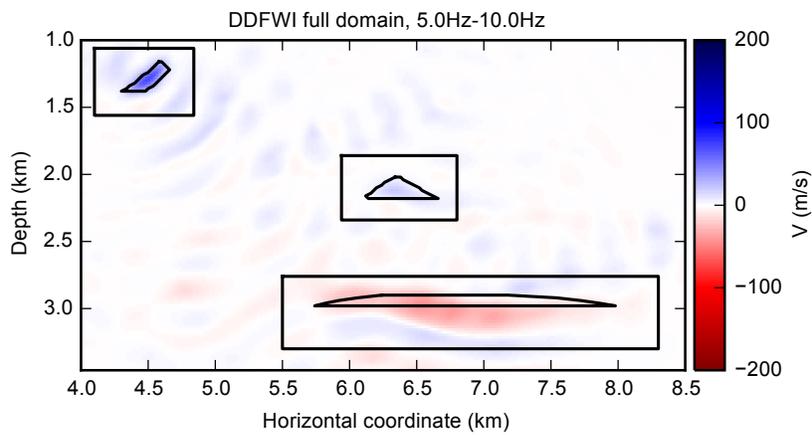


(b)

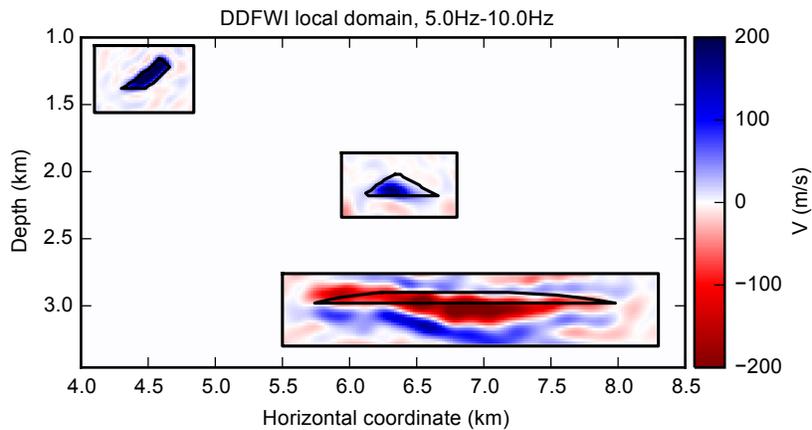
Figure 3-13: To test the local solver we compare the wavefields from the full domain solver and the local domain solver and plot the results in the white box of Figure 3-12b. Within the truncated domains (i.e. black boxes) the two wavefields are numerically identical. This means all interactions between the domains, including local model updates, and the exterior are included. This verifies that the multi-domain configuration of the local solver is setup correctly for this problem.



(a)



(b)



(c)

Figure 3-14: DDFWI results when sequentially inverting for the frequencies 5.0 Hz, 6.5 Hz, 8.0 Hz and 10.0 Hz. Compare the inversion results with the true time-lapse of Figure 3-11b. Figure 3-14a shows the result when inverting for time-lapse change in the full domain using a full domain solver with 16 iterations at each frequency. Figure 3-14b uses the setup, but reduces the number of iterations to 3 at each frequency. Finally Figure 3-14c also uses 3 iterations at each frequency but uses a local solver instead of a full domain solver and only updates the model within the black boxes.

Number of iterations	Number of inversions	t_{local}/t_{full}
1	1	5.91
8	1	0.81
20	1	0.37
8	8	0.17
20	8	0.11
20	20	0.09

Table 3.1: This table compares the computational cost of using the local solver and the full-domain solver. We use a speed-up factor of 13, which is what we observe in the example shown here, i.e. local solves are assumed to cost 1/13 of a full-domain solve. The computational cost of the local FWI includes both the pre-computation of (approximately 1225) full-domain Green’s functions (done once) and subsequent local solves (assuming 3.5 wave solves per source including the line search, for 60 sources this gives 210 wave solves per iteration). We assume that the same number of iterations are used for each algorithm. The third column shows the ratio of the time for the local version of FWI to perform the inversion to that of the full-domain solve.

Chapter 4

Regularizing velocity differences in time-lapse FWI using gradient mismatch information

Abstract

In this chapter we present a method for recovering time-lapse velocity changes using full waveform inversion (FWI). In a preprocessing step we invert for a single intermediate model by simultaneously minimizing the data misfit in the baseline and the monitor surveys. We record the individual FWI gradients corresponding to the baseline and the monitor datasets at each iteration of the inversion. Regions where these gradients consistently have opposing sign are likely to correspond to locations of time-lapse change. This insight is used to generate a spatially varying confidence map for time-lapse change. In a subsequent joint inversion we invert for baseline and monitor models while regularizing the difference between the models with this spatially varying confidence map. Unlike double difference full waveform inversion (DDFWI) we do not require identical source and receiver positions in the baseline and monitor surveys.

This chapter is based on Willemsen et al. (2015b)

4.1 Introduction

FWI (Lailly, 1983; Tarantola, 1984; Pratt, 1999) uses the entire seismic record to invert for subsurface material properties such as seismic velocity and density. When monitoring a reservoir, the primary objective is to recover changes in the material properties due to production changes. A straight-forward application of FWI to this time-lapse problem is to invert for the baseline (i.e., initial) state and the monitor (i.e., new) state independently. Subtracting these models intuitively approximates the time-lapse change in material parameters. However, artifacts are also present due to the nonlinear nature and ill-posedness of the uncoupled inverse problems for the baseline and monitor models.

In an attempt to overcome some of these problems Watanabe et al. (2004) and Denli et al. (2009) introduced DDFWI for monitoring production related changes. In DDFWI the baseline data residual from an initial baseline inversion is subtracted from the monitor dataset. Conceptually this can be seen as removing a part of the data that the initial model could not explain. After the monitor model is inverted from this modified data and the baseline model is subtracted a better time-lapse estimate is obtained. The data subtraction in DDFWI imposes constraints on the seismic acquisition. Both the source and receiver locations have to be the same in the baseline and monitor surveys. Differences are not uncommon in marine surveys as is illustrated for instance by Eggenberger et al. (2014). These differences will result in artifacts in the velocity change estimate.

Several methods have been introduced that avoid the requirement of identical acquisition. Zamanian et al. (2014) invert for the time-lapse change by casting the inverse problem in a hierarchical Bayesian framework. Maharramov et al. (2014) circumvent some of the problems of subtracting independent baseline and monitor inversions by swapping baseline and monitor datasets a fixed number of times. In a different approach Maharramov and Biondi (2014) introduce a joint inversion in which Total Variation (TV) regularization suppresses the oscillatory model-difference artifacts that arise in joint FWI with noisy datasets. Another approach is to itera-

tively swap baseline and monitor datasets in FWI and record which regions change consistently as a result. This information can be used to generate a confidence map, which can regularize the model-difference in a subsequent time-lapse inversion. This was done by Yang (2014) in a method called Alternating FWI (AFWI). The idea we introduce in this study is similar to Yang’s approach. Instead of performing a potentially long sequence of expensive full waveform inversions for the confidence map, we perform a single joint inversion with a single model. In this joint inversion we minimize both the baseline and monitor data residuals with this single intermediate model. Regions where the gradient of the baseline data term consistently has a different sign than the gradient of the monitor data term are considered to be potential regions of time-lapse change. We quantify this principle and construct a confidence map from the gradient history. Similar to the work of Yang (2014) we use this (different) confidence map to regularize the model difference in a final joint inversion.

4.2 Method

To obtain a confidence map we first minimize the objective function

$$\tilde{\chi}(m(\mathbf{x})) = \frac{1}{2} |\mathcal{F}_0(m(\mathbf{x})) - d_0|_{d_0}^2 + \frac{1}{2} |\mathcal{F}_1(m(\mathbf{x})) - d_1|_{d_1}^2, \quad (4.1)$$

where the misfit is $\tilde{\chi}$ and a single model m is used to fit both the baseline data d_0 and the monitor data d_1 through forward model $\mathcal{F}_0(m(\mathbf{x}))$ and $\mathcal{F}_1(m(\mathbf{x}))$ respectively. The baseline and monitor acquisition are not required to be the same. Both the data norms $|\cdot|_{d_0}$ and $|\cdot|_{d_1}$ are implemented as the L_2 norm. Model regularization can be applied to m but this is not done in the examples in this study. The gradient of (4.1) with respect to the model consists of two parts:

$$\frac{\delta \tilde{\chi}}{\delta m(\mathbf{x})} = g_0(\mathbf{x}) + g_1(\mathbf{x}), \quad (4.2)$$

where $g_0(\mathbf{x})$ and $g_1(\mathbf{x})$ are the standard FWI gradients of the baseline and the monitor data terms. The concepts we introduce in this research are applied to constant density

acoustics, but the ideas extend naturally to different physics. Equation 4.1 can be minimized with any iterative optimization routine, but in this research we use L-BFGS (Nocedal and Wright, 2006b). When inverting with the objective function given in (4.1), we expect to get an intermediate model $m(\mathbf{x})$ that explains both the baseline data, d_0 , and monitor data, d_1 , as well as possible. At locations \mathbf{x} where there is a real time lapse model change we expect that the FWI gradient of the baseline data term, $g_0(\mathbf{x})$, will have the opposite sign to the FWI gradient of the monitor term, $g_1(\mathbf{x})$. Intuitively this means that in regions of true time-lapse change an inversion using just the baseline data would move away from intermediate model m in the opposite direction to that an inversion using just the monitor data would move in. This insight can be used when constructing a confidence map. When inverting for the intermediate model $m(\mathbf{x})$ with an iterative inversion scheme we can keep the gradient history of both the data terms. We now define a confidence map $\beta_g(\mathbf{x})$ based on the gradient history,

$$\beta_g(\mathbf{x}) = \sum_i^n \left[|sgn(g_{0,i}(\mathbf{x})) - sgn(g_{1,i}(\mathbf{x}))| \times \left(\frac{1}{2} (|g_{0,i}(\mathbf{x})| + |g_{1,i}(\mathbf{x})|) \right) \right], \quad (4.3)$$

where $g_{0,i}(\mathbf{x})$ and $g_{1,i}(\mathbf{x})$ are the gradients of the baseline and monitor terms at iteration i respectively. $|f(\mathbf{x})|$ is the absolute value of f at each location \mathbf{x} and not a norm. To reduce the effects of differences in illumination we precondition the gradients using the inverse diagonal of the pseudo-Hessian proposed by Shin et al. (2001). The confidence map β_g in (4.3) has high values when the gradient of the baseline and the monitor term consistently have opposing sign and large amplitude. The reason for using the entire gradient history instead of the last iteration is to suppress some of the randomness and the nonlinear effects that take place when minimizing (4.1).

The confidence map $\beta_g(\mathbf{x})$ is used in a subsequent joint inversion where the following objective function χ is minimized for the baseline model m_0 and the monitor

model m_1 ,

$$\begin{aligned} \chi(m_0(\mathbf{x}), m_1(\mathbf{x})) &= \frac{1}{2} |\mathcal{F}_0(m_0(\mathbf{x})) - d_0|_{d_0}^2 \\ &+ \frac{1}{2} |\mathcal{F}_1(m_1(\mathbf{x})) - d_1|_{d_1}^2 + \frac{1}{2} \epsilon \left| \frac{m_0(\mathbf{x}) - m_1(\mathbf{x})}{\beta_g(\mathbf{x})} \right|_m^2. \end{aligned} \quad (4.4)$$

The data norms $|\cdot|_{d_0}$ and $|\cdot|_{d_1}$ and the model norm $|\cdot|_m$ are implemented as the L_2 norm. The scalar ϵ weights the model difference regularization term. Depending on the choice of ϵ , the model regularization term can be used to strongly suppress model differences outside the regions with high beta. Additional regularization can be applied to the individual models, but this is not investigated in this study. Note that there is nothing in this formulation that requires the baseline and monitor acquisitions to be the same. In the remainder of the study we refer to the process of first minimizing (4.1) and recording the gradients in (4.2) for confidence map (4.3) and then solving (4.4) as Gradient Confidence FWI (GC-FWI).

4.3 Numerical investigation

The performance of GC-FWI is compared with DDFWI on a synthetic model under noisy conditions. We compare using unrealistically favorable conditions for DDFWI. The source and receiver locations are exactly the same in both surveys and so is the source wavelet. These tests thus give us a conservative estimate of the expected improvements from GC-FWI.

4.3.1 Setup of synthetic problem

The model on which the synthetic study is done is the Marmousi model, see Figure 4-1. The model is discretized using 151 nodes in the z direction and 461 nodes in the x direction. The node spacing is 20 m in both directions. The baseline and monitor surveys have 19 identical shot locations equally distributed over the surface and 461 identical receiver locations. The source wavelet is a 6 Hz ricker wavelet and is exactly the same in the baseline and monitor surveys. The same time-domain solver is used

for generating the data and for solving the inverse problem. The model includes a Perfectly Matched Layer (PML) on all sides. No model regularization is used on the individual baseline and monitor models in GC-FWI or in DDFWI. We generate uncorrelated Gaussian noise in the time-domain and add this to the data. We define the noise level r as the energy of the noise relative to the energy of the noiseless signal in the entire seismic survey,

$$r = \frac{\sum_i^{n_s} \sum_j^{n_r} \sum_k^{n_t} n_{i,j,k}^2}{\sum_i^{n_s} \sum_j^{n_r} \sum_k^{n_t} s_{i,j,k}^2}. \quad (4.5)$$

In (4.5), n_s is the number of sources, n_r is the number of receivers, and n_t is the number of time samples in a trace. For a specific noise level we first invert for the baseline model starting from a heavily smoothed initial guess. Figure 4-2 shows that good reconstruction is achieved for $r = 0.64$. When comparing DDFWI and GC-FWI at a specific noise level r , the corresponding inverted baseline model is used in DDFWI and in GC-FWI as the initial guess for m in (4.1). Figure 4-3 shows the true time-lapse velocity change used in this study.

4.3.2 Calculating the confidence map $\beta_g(\mathbf{x})$

We perform 20 iterations to minimize (4.1) in a single joint inversion and record the gradients $g_0(\mathbf{x})$ and $g_1(\mathbf{x})$ for noise levels $r = 0.04$ and $r = 0.64$. The gradients are preconditioned (Shin et al., 2001) with the purpose of correcting for differences in illumination. The preconditioned gradient history is then used to obtain the confidence map $\beta_g(\mathbf{x})$ using (4.3). We normalize the confidence maps by dividing each by its maximum value. The results are compared in Figure 4-4.

At the locations marked with arrows in Figure 4-4, we plot the evolution of the preconditioned gradients using matching colors in Figure 4-5. We see that most often the preconditioned gradients $g_0(\mathbf{x})$ and $g_1(\mathbf{x})$ have opposite signs and large amplitude at the location marked by the black arrow. The baseline data term and the monitor data term try to move the intermediate model m , computed by minimizing (4.1), in opposite directions. This is a sign that the baseline and monitor velocities may differ

and thus a true time-lapse change is likely at this location. Equation 4.3 for the confidence map therefore assigns a high value of $\beta_g(\mathbf{x})$ to the location of the black arrow. At the location of the red arrow the gradients of both data terms are often small and have the same sign. The low amplitude and the agreement in sign makes it unlikely that a true time-lapse change is present at this location, because both data sets agree that the model fits the data well, and if they try to adjust it at all do so in the same direction. In the preconditioned gradient evolution plot of Figure 4-5 we observe some transient behavior, which is a reason for involving more than one iteration when computing the confidence map $\beta_g(\mathbf{x})$ in (4.3).

4.3.3 Comparing GC-FWI and DDFWI

With the confidence map $\beta_g(\mathbf{x})$ we regularize the model difference in the joint inversion (4.4). We perform 15 nonlinear iterations and minimize the objective function using L-BFGS. We compare the results of GC-FWI at noise level $r = 0.64$ with 20 nonlinear iterations of DDFWI starting from the initial inverted baseline model corresponding to that noise level. Since the weighing of a regularization term is not trivial we show the results for three different values of parameter ϵ , which weights the model regularization in (4.4).

We display the results of the comparison between GC-FWI and DDFWI in Figure 4-6. We see that the level at which GC-FWI suppresses noise in the time-lapse estimate depends on the model regularization weight ϵ . At the low end of the investigated values (i.e., $\epsilon = 10.0$) this noise is only slightly less than in DDFWI. At higher values of ϵ the noise in the recovered GC-FWI time-lapse estimates goes down. At the high end of the investigated values of ϵ (i.e., $\epsilon = 1000.0$) we observe the strongest noise suppression, but we also notice that the recovered time-lapse change on the rightmost perturbation is suppressed below the true value of Figure 4-3.

4.4 Discussion

In this study we use identical wavelets and identical source and receiver locations in the baseline and monitor surveys. In reality these conditions are hard to satisfy and deviations will inherently result in artifacts in DDFWI as is documented by for instance Yang (2014). Instead of subtracting data, GC-FWI works in the model space by comparing the stacked gradients of the baseline and monitor survey evaluated at an intermediate model. It is true that differences in source and receiver locations in baseline and monitor surveys will change the gradient contributions of the shots, but the stack of these contributions seems quite robust to changes in source location. Tests indicate that if the 19 monitor shots are offset from the baseline locations by half the shot spacing, the largest shift possible, the resulting confidence map $\beta_g(\mathbf{x})$ is qualitatively the same. We have not yet investigated how realistic differences in the source wavelet would influence GC-FWI.

For the example in this study, the cost of GC-FWI is approximately 3.5 times that of DDFWI. The cost of $\beta_g(\mathbf{x})$ is 40 standard FWI iterations because each of the 20 iterations of (4.1) is twice the cost of a standard FWI iteration due to the presence of two data terms. Each of the 15 iterations of the subsequent joint inversion (4.4) is also twice as expensive as a standard FWI iteration for the same reason. This brings the total cost of GC-FWI to 70 standard FWI iterations compared to the 20 standard FWI iterations of DDFWI in this example. In this calculation we do not include the cost of the initial inverted baseline model that is used in DDFWI and as the initial guess in (4.1).

Even though GC-FWI significantly reduces the constraints, imposed by DDFWI, on the acquisition of the repeat survey, current time-lapse datasets have very similar source and receiver locations. This knowledge can potentially be used to increase the accuracy of the confidence map $\beta_g(\mathbf{x})$. Instead of comparing the stacked gradient of the baseline and the monitor data term in (4.1) we could compare the gradient history of each shot individually. Since the shot and receiver locations are similar in the baseline and monitor survey, the major difference in the gradient contribution of

a shot will be due to noise and time-lapse model differences. Generating $\beta_{g,s}(\mathbf{x})$ from the history of each shot s and then stacking the result for the final confidence map $\beta_g(\mathbf{x})$ may give improved results. If the source and receiver locations in the baseline and monitor surveys are similar and DDFWI is possible, $\beta_g(\mathbf{x})$ could also be used to regularize the model difference in DDFWI. A significant portion of the noise in the DDFWI time-lapse estimate in Figure 4-6 could be suppressed in this way.

4.5 Conclusion

The method introduced in this study performs an initial inversion where the baseline and the monitor data misfit are minimized with a single model. By comparing the gradients of both data terms at each iteration of the inversion we gain insight about the probability of a time-lapse velocity change in a region. This knowledge is encoded in the confidence map $\beta_g(\mathbf{x})$, which is then used to regularize the model difference in a final joint inversion. This method, which we refer to as GC-FWI, is flexible to differences in acquisition and shows promising results on synthetic examples.

4.6 Acknowledgements

I acknowledge the interactions with my co-author Alison Malcolm for the SEG abstract on which this chapter is based (Willemsen et al., 2015b). We acknowledge Laurent Demanet from MIT and Baishali Roy and Jun Cao from ConocoPhillips for insightful discussions. We would also like to thank MIT and the ERL consortium members for funding.

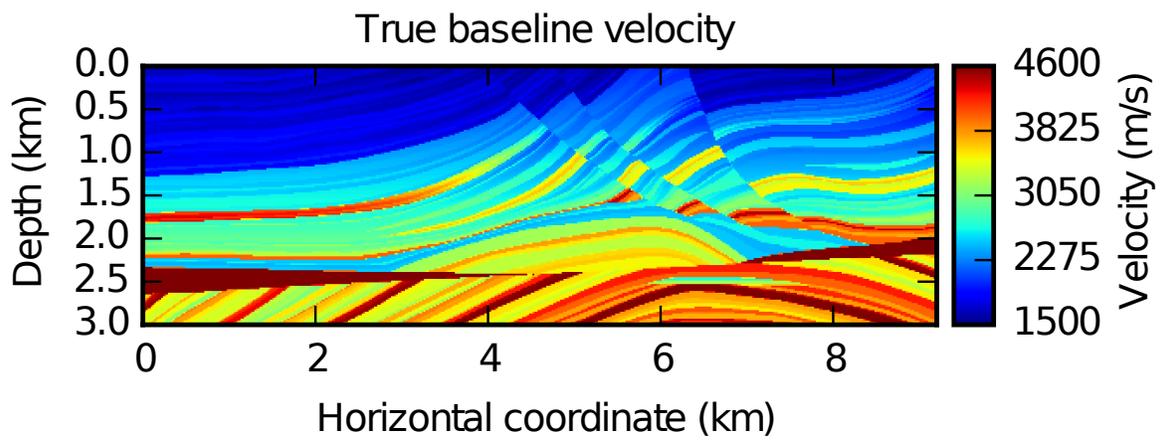


Figure 4-1: The true baseline model.

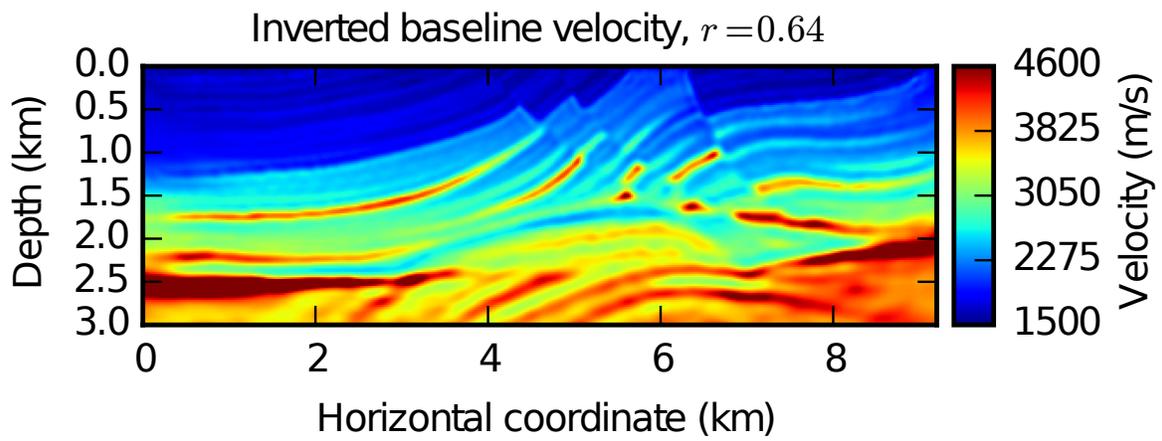


Figure 4-2: The baseline model inverted from the baseline data using standard FWI for $r = 0.64$.

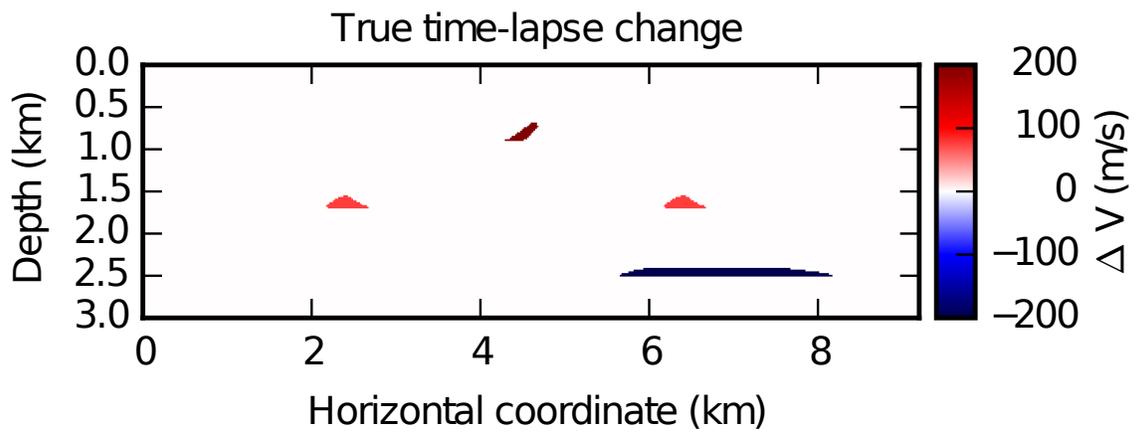


Figure 4-3: The true time-lapse velocity change investigated in this study.

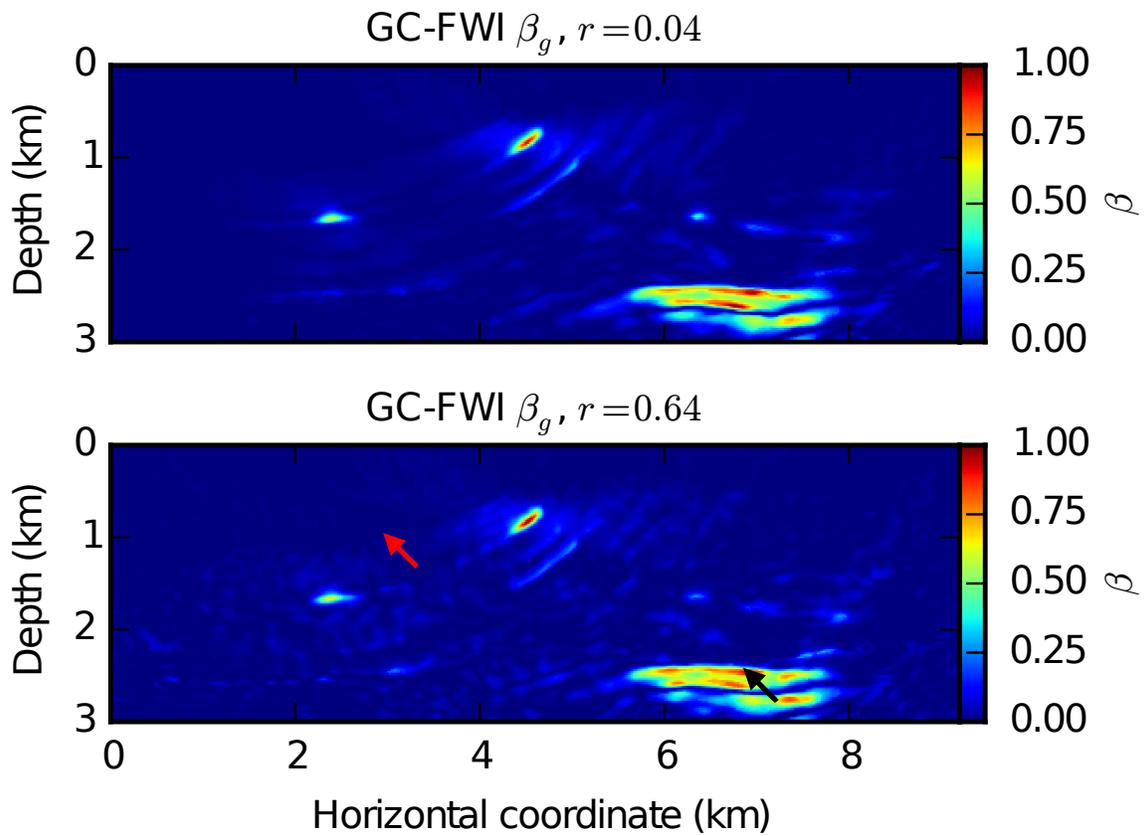


Figure 4-4: Confidence map $\beta_g(x)$ for GC-FWI for noise levels $r = 0.04$ and $r = 0.64$. The black and red arrows correspond to the preconditioned gradient evolution curves of the same color in Figure 4-5.

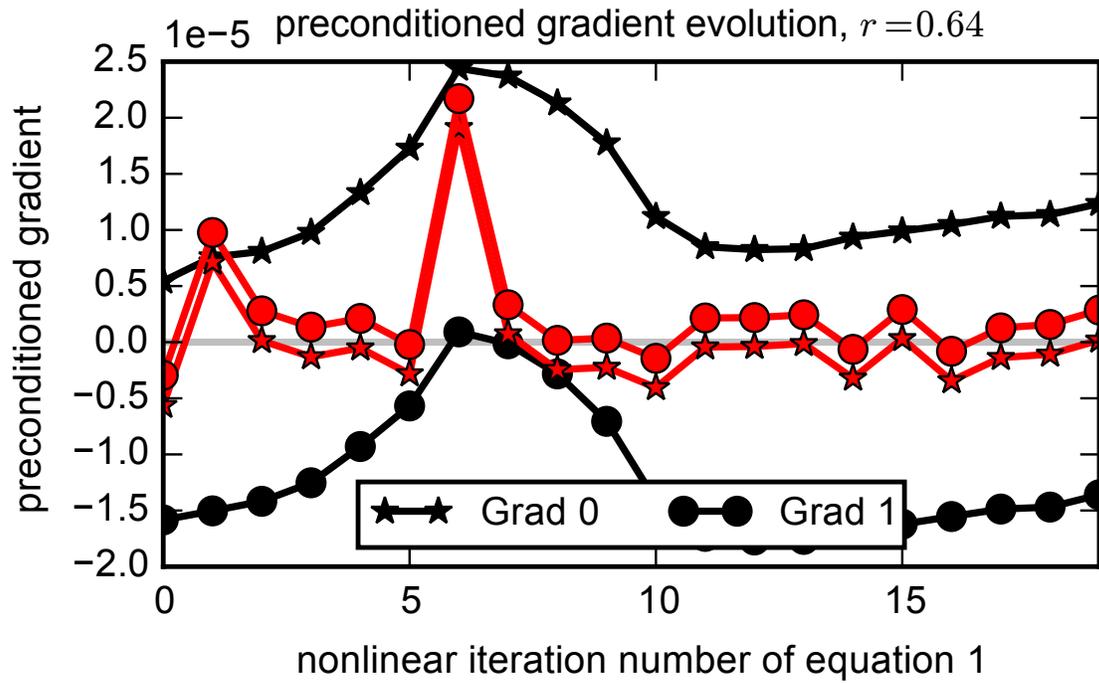


Figure 4-5: Evolution of the preconditioned gradients of both data terms of (4.1) in the inversion, at the locations indicated by arrows of corresponding color in Figure 4-4. True time-lapse regions correspond with gradients of large amplitude and different signs (black) whereas in regions without timelapse changes the gradients have small amplitude and often the same sign (red).

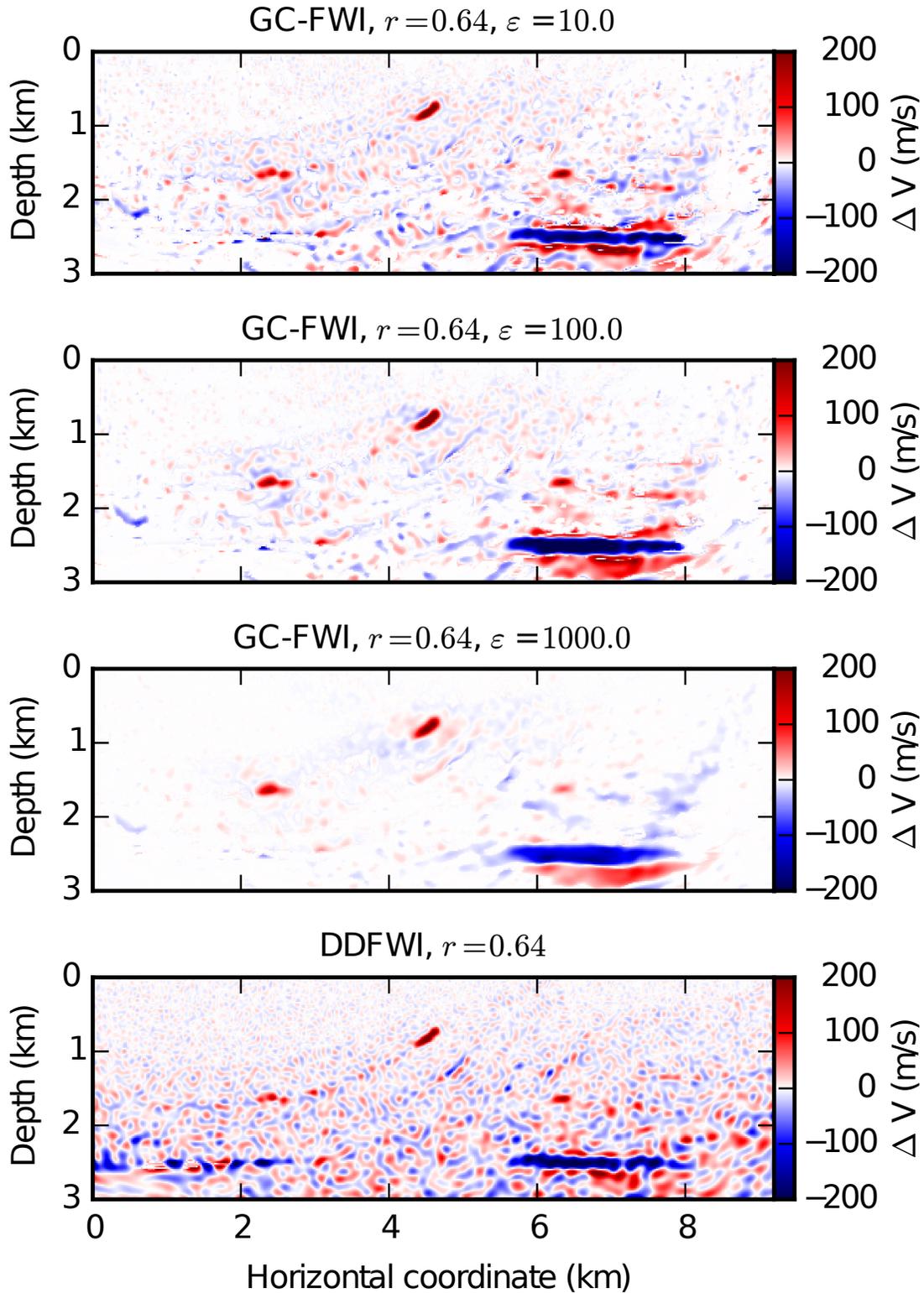


Figure 4-6: Comparison of GC-FWI using three different values of the model-regularization weighting term ε with DDFWI for noise level $r = 0.64$. Figure 4-3 shows the true model time-lapse change. At low regularization (top) GC-FWI gives results similar to those of DDFWI, but as ε is increased GC-FWI suppresses spurious changes and results in a more accurate recovery of the time-lapse changes.

Chapter 5

Evaluating the performance of acoustic FWI time-lapse workflows on realistic synthetic data

Abstract

Recently there has been an increased interest in applying Full Waveform Inversion (FWI) to the time-lapse problem. In this chapter we evaluate the performance of several time-lapse FWI algorithms on a realistic North Sea reservoir model. Because full domain elastic wavefield simulations are still expensive in 3D, we focus in this synthetic example on acoustic inversion for the P-wave velocity time-lapse change from the recorded data. Since the recorded waves traveled through an elastic Earth, making an acoustic approximation in the inversion will result in artifacts. We investigate the impact of these artifacts by using three recorded datasets with increasingly realistic physics, starting at constant density acoustics and ending at fully elastic. In these results we observe that the time-lapse estimate contains significant artifacts because of post-critical reflections. We then evaluate the impact of the initial model and also apply the GC-FWI confidence map regularization of chapter 4 to constrain the time-lapse estimate.

This chapter is an unpublished paper, extending Willemsen et al. (2016a)

5.1 Introduction

Seismic material properties such as density, P-wave velocity and S-wave velocity determine how a wavefield propagates through the Earth. During the lifetime of a reservoir, many processes such as hydrocarbon production and water/CO₂ injection take place. These processes may change the pore fluids, pore pressure and effective stress in the reservoir. Seismic properties of the bulk rock change as a result. Gassmann's law (Gassmann, 1951) is often used to estimate the effects of pore fluid substitution on the bulk seismic velocity and density (Kumar et al., 2006). The bulk modulus and density and therefore the velocity of oil depend on the pressure, especially when the bubble point is passed and gas escapes from the oil (Han et al., 2001). Pore pressure changes in the reservoir change the effective stress field, not only in the reservoir but also in the surrounding rocks due to geomechanical interactions (Tura et al., 2005). These changes in effective stress modify the stiffness of grain contacts and can open or close thin cracks and pores (Johnston, 2013), altering the velocity. In some reservoirs a drop in pore pressure can result in reservoir compaction and subsidence of the overburden (Geertsma et al., 1973). This changes the travel time of the seismic waves by both changing the velocity and the thickness of the layers (Tura et al., 2005). All these processes demonstrate that changes in seismic properties are intricately related to physical processes in the reservoir. Time-lapse seismology attempts to recover the seismic changes so they can be correlated to these physical processes. Monitoring reservoir production can help to optimize it. Examples of applications include the monitoring of injected water (Calvert et al., 2014) and CO₂ (Davis et al., 2003; White, 2009; Davis, 2015) during secondary or enhanced oil recovery with the objective of detecting unswept areas.

Changes in velocity and layer thickness causes time-shifts in recorded traces, and as a result in time-migrated images. These time-shifts can for instance be computed by finding the local cross-correlation peak between baseline and monitor images (Rickett et al., 2006). The measured time-shift at a particular location is the sum of all changes in the travelttime caused by the layers above. A useful indicator for time

lapse change at a particular location in the Earth is obtained by differentiating this time-shift. This so-called time strain (Rickett et al., 2006, 2007) is nonzero when there is a gradient in the time-shift, as is the case when the time-shift at the top of a layer is different than the timeshift at the bottom of that layer. Landrø and Stammeijer (2004) introduce a method for untangling the coupled effect of compaction and velocity change on the time strain by comparing near and far offset data. They also introduce another method for untangling compaction and velocity changes by looking at combined changes in travel time and reflection amplitude. By using a relation between the vertical strain (i.e. thickness change of layer) and the velocity change, Røste et al. (2006) and Hatchell and Bourne (2006) show that time-strain and vertical strain are directly related. Hodgson et al. (2007) uses this observation to link the 4D observations to pore-pressure changes using a geomechanical model. Examples of field applications where amplitude changes or time shifts are used to infer reservoir changes are given by for instance Ghaderi and Landrø (2009), Eggenberger et al. (2014) and Røste et al. (2015).

The methods presented up until now are based on time-shifts and amplitude changes, typically of time-migrated images. Inaccuracies in source-receiver repeatability impact the images and therefore influence the time-shifts and amplitude changes. Source-receiver non-repeatability may therefore lead to time-lapse noise, especially in locations with strong near-surface heterogeneity. In addition, which source-receiver combinations are repeatable is determined by comparing their impact on migrated images. The process of selecting repeatable pairs therefore requires frequent migration and is time consuming. For these reasons, amongst others, there is an interest in using Full Waveform Inversion (FWI) (Lailly, 1983; Tarantola, 1984) for the time-lapse problem. FWI is easier to automate than the conventional time-lapse approach. FWI is also an inverse problem and there are signs that it is better able to deal with non-repeatable acquisition, for instance by using regularization (Maharramov and Biondi, 2014; Yang, 2014). The research introduced in this chapter falls in this category.

An intuitively straight-forward application of FWI to the time-lapse problem is

simply inverting for the baseline and the monitor models independently and then subtracting the results. If the baseline and monitor inversions start from the same initial model this is commonly referred to as parallel FWI. When instead the monitor FWI inversion starts from the inverted baseline model this is often called sequential FWI (Maharramov et al., 2014). An example comparing parallel and sequential FWI is given by Routh et al. (2012). The time-lapse estimate can contain artifacts because the independent inversions for the baseline and monitor data may converge to different local minima (Mulder and Plessix, 2008). Watanabe et al. (2004) introduced Double Difference FWI (DDFWI) which subtracts the baseline dataset from the monitor dataset and then inverts for this data difference. The advantage of this approach is that when the survey repeatability is good, it is possible to directly invert for the time-lapse change that caused the observed data difference. When the source and receiver locations are not perfectly repeated or when there is non-repeatable noise, the data difference will contain effects other than those caused by the time-lapse change. Other examples of synthetic DDFWI studies include the work of Denli et al. (2009) and Asnaashari et al. (2011). Yang et al. (2015) compares parallel, sequential and double-difference FWI and also investigates the robustness of DDFWI in the presence of noise or imperfect repeatability. Joint FWI is the process where the baseline and monitor model are inverted for simultaneously. Since no data subtraction takes place, the baseline and monitor surveys are not required to have exactly the same acquisition. Regularization of the model difference can be used to mitigate non-repeatability footprints in the reconstructed time-lapse estimate (Maharramov et al., 2014). Yang (2014) gives an example of this when he obtains regularization parameters from a preprocessing step and refers to them as a confidence map. This confidence map regularization is subsequently used for Joint FWI where the baseline and monitor sources are displaced by more than a kilometer. Maharramov and Biondi (2014) use Total Variation (TV) regularization for the model difference in Joint FWI, where they make use of the idea that the model difference is expected to be localized. Zamanian et al. (2014) obtains a very accurate time-lapse reconstruction under imperfect repeatability by casting the joint inversion in a hierarchical Bayesian framework. Zhang and

Huang (2013) demonstrate that even when repeatability is good, using regularization to constrain velocity updates to the reservoir region can significantly enhance the DDFWI time-lapse reconstruction.

Despite the theoretical work, there is a only limited number of applications of time-lapse FWI on real data, see for instance the work of Queißer et al. (2010), Yang (2014) and Maharramov et al. (2015). In these studies the acoustic approximation is made on field data which inherently contain elastic effects from the elastic Earth. To the best of our knowledge, there are no studies that investigate how this common acoustic approximation affects the time-lapse reconstruction using elastic observed data on a realistic geological model. It is important to build understanding for how the P-velocity time-lapse estimate is influenced by elastic effects in the data in order to evaluate field applications. The effect of the acoustic approximation on regular FWI has been studied by for instance Mulder and Plessix (2008), but the effects on time-lapse reconstruction have not received the same level of attention. The significance of this chapter is that we use an approximately realistic acoustic workflow on a geological model based on a real North Sea reservoir in which we model the data with an elastic solver and investigate the error in the time-lapse results. In particular we focus the investigation on the impact of elastic data on the P-wave time-lapse estimate. The workflow we use is closely related to a standard FWI workflow, although a few simplifying assumption are made. These simplifications are clearly documented when they occur.

This chapter is organized as follows. We first introduce the theoretical background of time-lapse FWI. Then we generate constant density acoustic, variable density acoustic and elastic data. We use this data in a constant density acoustic inversion and use a different code to avoid the so-called inverse crime (Colton and Kress, 1998; Kaipio and Somersalo, 2006). Since this is a synthetic study, we have perfect knowledge of the true time-lapse change. With this knowledge we can evaluate how increasingly realistic physics will impact the time-lapse reconstruction. We then propose and demonstrate some techniques to partially overcome the artifacts introduced to the P-wave time-lapse estimate by the elastic data. We recommend that

time-lapse studies using real observed data should always include a synthetic study like the one we present. Since the true model is available in such a synthetic study, there is an opportunity to better understand which factors influence the accuracy of the inversion. These factors may differ based on the acquisition and the complexities of the Earth model. Based on the recommendations from the corresponding synthetic study, the workflow on the real data can then be adapted accordingly.

5.2 Methodology

We study the performance of time-lapse FWI algorithms on a realistic synthetic model. An internal reservoir simulator provides a 3D cube of approximate seismic properties of a real North Sea field. A 2D slice is extracted from this cube and will serve as the true Earth model in this study. We use the same elastic solver from the commercial code SeisSpace to generate data with increasingly complicated physics on this 2D slice. Even though the synthetic true data will contain more complicated physics, we invert for V_p only. By incrementally adding additional physics to the true data we investigate how inversion for only V_p is affected.

5.2.1 Baseline velocity model

Figures 5-1a, 5-1b and 5-1c show the true acoustic velocity V_p , the density ρ and the shear velocity V_s respectively. The high velocity layer at depths slightly larger than 3 km represents the fractured chalk reservoir. The top of the anticline at $x = 3.9$ km and $z = 2.9$ km shows lower V_p and ρ than the surrounding reservoir due to the presence of hydrocarbons. The overburden consists primarily of approximately horizontal layers, although some notable square-shaped deviations are visible right above the anticline. These square-shaped regions have contrasting V_p and density compared to the sediments surrounding them. They represent gas clouds above the reservoir, which have also been observed at other North sea fields (Liu et al., 2012). An eighty meter deep water layer represents the shallow North Sea.

The original reservoir model contains a shear velocity gradient in the shallow

sediments at the bottom of the ocean. In this model the shear velocity is zero in the water layer and approximately 500.0 m/s at a depth of 50 m into the sediment. Within this 50 m long gradient there are pixels with V_s very close to zero and very small corresponding wavelengths. In order to maintain a certain number of grid points per wavelength, the grid spacing needs to be prohibitively fine. For this computational reason the V_s gradient is turned into a sharp jump. In reality a smoother V_s gradient will be present in the true Earth than the jump in Figure 5-1c.

5.2.2 Time-lapse perturbation

When producing hydrocarbons the pore fluids change. The pore pressure also changes due to the extraction of hydrocarbons and the injection of water. Both the changes in pressure and pore fluid affect the seismic properties V_p , V_s and ρ . The baseline and the monitor models we consider are separated in time by five years and are obtained from a reservoir simulator. Figure 5-2 shows a zoomed in image of the true time-lapse change investigated in this paper. The time-lapse change outside of this zoomed region is negligible. The largest changes are taking place within the reservoir. Two artificial elliptical time-lapse perturbations are added at the horizontal locations 4.2 km and 5.9 km. They represent proposed waste-water injection locations and are added to the reservoir simulator obtained time-lapse change manually. Notice that these regions have no ρ or V_s time-lapse change. In reality the injection of waste-water and the associated pressure change will have an influence on ρ and V_s as well. A more realistic time-lapse change is obtained when running the reservoir simulator with the proposed waste-water injectors. But in the interest of time this is not done, since it would require another expensive run of the reservoir simulator. Subsidence and the associated velocity decrease due to overburden stretch is not simulated.

5.2.3 Data generation

We use the same 2D elastic solver from the commercial SeisSpace code to generate data for three different scenarios.

- Constant Density Acoustic (**CDA**): $V_p(\mathbf{x})$ model, constant ρ , constant zero V_s
- Variable Density Acoustic (**VDA**): $V_p(\mathbf{x})$ model, $\rho(\mathbf{x})$ model, constant zero V_s
- Elastic (**EL**): $V_p(\mathbf{x})$ model, $\rho(\mathbf{x})$ model, $V_s(\mathbf{x})$ model

with the 2D models displayed in Figures 5-1. This lets us investigate how increasingly realistic physics influence V_p only inversion. To limit the number of parameters we consider we are not including free surface effects. In real North Sea data the shallow water generates strong multiples, which either have to be removed from the data or must be matched using the forward model. We also do not include changes in the velocity and depth of the water column in this time-lapse study. The source is a Ricker wavelet with a peak frequency of 6.0 Hz. In a real seismic survey higher frequency sources are typically used. The data is then low-pass filtered and higher frequencies are slowly added using criteria like those of Sirgue and Pratt (2004) to avoid cycle skipping. Since such a frequency marching approach would add complexity to the study, we choose to use the fixed wavelet instead with low enough peak frequency (6.0 Hz) to converge towards the true Earth model. We simulate 40 shots with 250 m spacing, covering the entire surface of the model. These shots are fired at a depth of 12.5 m. There are 405 Life of Field Seismic (LoFS) pressure receivers with 25 m spacing at a depth of 87.5 m. These receivers represent permanent cables that are placed on the bottom of the ocean. The sea is 80 m deep in our model, so the receivers are 7.5 m beneath the surface. This is deeper than for instance the 1.5 m for the real world LoFS installed at Ekofisk (Bertrand et al., 2013). The reason for placing the receivers deeper in our numerical study is that it makes them coincide with the first sediment grid point in the 12.5 m uniformly spaced inversion grid. The unrealistic burial depth of the LoFS receivers in our numerical study is expected to have negligible effect, because the dominant wavelength for a 6.0 Hz peak frequency Ricker wavelet is on the order of 300 m. Unlike streamers, the LoFS receiver locations do not vary with shot. Because the cables are permanent and there is no subsidence in this study, the receiver locations in the baseline and monitor surveys are exactly the same. The grid spacing we use to generate the true data is finer than that of

the uniform 12.5 m of the inversion grid. Due to the primarily horizontal geology, the vertical grid spacing is finer than the horizontal spacing in the true model. The horizontal spacing we use is 12.5 m and the vertical spacing is 2.0 m.

The baseline shot gathers for a source at $x = 2.0$ km for the three different scenarios are displayed in Figure 5-3. Three prominent events are labeled in the CDA shot gather. The headwave traveling at the bottom of the ocean is given the number one. This wave travels with the speed of the shallow sediment, which is slightly less than 2000 m/s. The event marked with the number two is the direct wave hitting the ocean bottom at a very flat angle. Due to the large V_p contrast at the reservoir, the top of reservoir reflection has a very large amplitude. This is the event marked with the number three.

For the VDA scenario the variable density of Figure 5-1b is added to the solver. We see that the VDA shot gather in Figure 5-3b looks very similar to the CDA one from Figure 5-3a, with most changes being in the amplitude. The event with the number four for instance has a noticeably different amplitude from the one in the CDA shot gather. In the CDA shot gather this event is caused by a decrease in V_p along the entire horizontal extent of the model at $z = 1.5$ km. This decrease in V_p coincides with the corresponding low density layer in Figure 5-1b, further enhancing the impedance contrast and therefore affecting the amplitude of event four. Another important difference between VDA and CDA is that at large offset the direct wave has a different phase. The direct wave hits the ocean bottom where the receivers are buried at a post-critical angle and is therefore expected to experience a phase shift. This phase shift is determined by the contrast in material parameters as is discussed by Zhu and McMechan (2012) for spherical wavefronts. The addition of the large density contrast at the ocean bottom in VDA is expected to result in a different phase shift.

Next we add the variable V_s from Figure 5-1c to the VDA simulation to get a fully elastic simulation. The shot gather in Figure 5-3c is markedly different from the acoustic ones. We see that several slow events are introduced in the fully elastic simulation. The slowest event marked with the number 5 travels at a velocity slightly

less than the shear velocity. This event is the Scholte wave, which is also known as the mud roll. All the post-critical reflections will have their phase shift determined by the V_p , ρ and V_s contrasts at layers.

5.2.4 Baseline inversion algorithm

The first step of time-lapse FWI is to invert for the baseline model using FWI (Lailly, 1983; Tarantola, 1984; Pratt, 1999). Because we make the acoustic approximation we invert for the P-velocity, even if the true datasets contains more complicated physics. FWI minimizes the objective function

$$\chi(m) = \|d - \mathbf{S}u(m)\|_2^2 \quad (5.1)$$

where d is the true data, \mathbf{S} is the sampling operator and $u(m)$ is the wavefield in the current model estimate m and $\|\cdot\|_2^2$ is the squared l^2 norm.

Colton and Kress (1998) use the term 'inverse crime' when referring to the procedure of using the same solver to generate the true data and to generate the data on the current model estimate. They mention that this so-called inverse crime can lead to trivial inversion and therefore recommend that the solver used in the inversion should have no connection to the solver used for the generation of the synthetic true data. By using the Python Seismic Inversion Toolbox (PySIT) for constant density acoustic inversion of the true data from the SeisSpace code (Figure 5-3), we avoid the inverse crime and the potentially trivial inversion associated with it.

The amplitude in the EL true data is affected by for instance P-S conversions and variable density. Because of this we cannot invert directly for the amplitude using a P-velocity only inversion and so an amplitude normalization operator is added to (5.1). To do this, at each time sample within a trace we calculate the energy within a certain time window. We do this for both the true data and for the data on the current model estimate. We then normalize one to the other by multiplying by this precomputed energy-ratio trace. This gives a scaling factor that is a function of time as well as source and receiver positions.

The objective function in (5.1) can either be minimized in the time or the frequency domain. In this paper we use the time-domain and avoid cycle skipping by slowly increasing the time-window of inversion. The source wavelet has a 6Hz peak frequency and therefore a relatively large wavelength. Real field data is usually recorded using a higher peak frequency wavelet, but low frequency components can be extracted through for instance a band-pass filter. The full Newton method (see for instance (Virieux and Operto, 2009)) solves the following system of equations at each iteration

$$\vec{s}_k = \mathbf{H}_k^{-1} \vec{g}_k \quad (5.2)$$

where \vec{s}_k is the search direction vector, H_k is the Hessian and \vec{g}_k is the gradient at nonlinear iteration k . For minimizing (5.1) we use the iterative L-BFGS quasi-Newton optimization algorithm (Nocedal and Wright, 2006a). This algorithm uses the gradients from earlier nonlinear iterations to approximate the application of \mathbf{H}_k^{-1} to a vector. It does this without forming the prohibitively large matrix \mathbf{H}_k^{-1} in memory. Brossier et al. (2009) show an example of an application of L-BFGS to FWI.

5.2.5 Time-lapse algorithms

In this paper we investigate the performance of the following FWI time-lapse algorithms.

- Parallel FWI
- Double Difference FWI (DDFWI)
- Joint FWI

To the best of our knowledge, there are only very limited field studies of time-lapse FWI algorithms. For instance, Watanabe et al. (2004) use DDFWI in a cross-well survey where they invert for changes in acoustic velocity V_p . Yang (2014) uses DDFWI and Parallel FWI as reference to also obtain changes in V_p at the Valhall reservoir

in the North Sea. In both of these examples the field data contain elastic effects and the inversion is for the time-lapse changes in V_p . It is poorly understood how the imprint of more complicated physics on the data affect the quality of the these inversions. In this study we start with CDA data and then gradually transition to more complicated physics (i.e. VDA, EL) to better understand their impact on the time-lapse inversion. We use PySIT to avoid the inverse crime for the same reason as for the baseline inversion. We now introduce the different time-lapse FWI algorithms.

Parallel FWI

Starting from the same initial model m_i , parallel FWI minimizes the following two FWI objective functions independently for the same number of iterations

$$\chi_0(m_0) = \left\| d_0 - \mathbf{S}u(m_0) \right\|_2^2 \quad (5.3)$$

$$\chi_1(m_1) = \left\| d_1 - \mathbf{S}u(m_1) \right\|_2^2, \quad (5.4)$$

where d_0 is the baseline data and d_1 is the monitor data. The FWI objective functions follow notation of (5.1). The inverted baseline model m_0 is subtracted from the inverted baseline model m_1 to obtain the parallel FWI time-lapse estimate. Parallel FWI is arguably the most straight-forward way to use the entire waveform in a time-lapse context. However, it is well documented (Maharramov et al., 2014; Watanabe et al., 2004; Yang, 2014) that the Parallel FWI time-lapse estimate is poor. In the context of this paper we use it as a benchmark.

DDFWI

Double Difference FWI first inverts for the baseline model m_0 by minimizing

$$\chi(m_0) = \left\| d_0 - \mathbf{S}u(m_0) \right\|_2^2 \quad (5.5)$$

In any nontrivial FWI, a part of the data will remain unexplained after inversion. This so-called residual is defined as the following for our baseline inversion

$$r_0 = d_0 - \mathbf{S}u(m_0). \quad (5.6)$$

We subtract the unexplained r_0 from the monitor data d_1 , resulting in modified monitor data \hat{d}_1 .

$$\begin{aligned} \hat{d}_1 &= d_1 - r_0 \\ &= d_1 - (d_0 - \mathbf{S}u(m_0)) \end{aligned} \quad (5.7)$$

We then minimize a standard FWI objective function with \hat{d}_1 as the data.

$$\begin{aligned} \hat{\chi}(m_1) &= \|\hat{d}_1 - \mathbf{S}u(m_0)\|_2^2 \\ &= \|(d_1 - d_0) - (\mathbf{S}u(m_1) - \mathbf{S}u(m_0))\| \end{aligned} \quad (5.8)$$

$$= \|(\mathbf{S}u(m_0) + \delta d) - \mathbf{S}u(m_1)\| \quad (5.9)$$

where (5.8) shows the double difference form and $\delta d = d_1 - d_0$. We invert only for m_1 and keep m_0 fixed as the inverted baseline model. The equivalent (5.9) shows that the initial residual is δd if we choose m_0 as initial model for m_1 . The DDFWI algorithm therefore performs data subtraction to focus on the time-lapse signal δd in the true data. This data subtraction assumes perfect repeatability of source and receiver locations and the source wavelet. In reality these conditions will not be met exactly, resulting in artifacts investigated by for instance Yang (2014).

Joint FWI

Joint FWI differs from Parallel FWI in that it inverts for the baseline and the monitor model simultaneously instead of independently

$$\chi_{0,1}(m_0, m_1) = \|d_0 - \mathbf{S}u(m_0)\|_2^2 + \|d_1 - \mathbf{S}u(m_1)\|_2^2. \quad (5.10)$$

After inversion, the inverted baseline model m_0 is again subtracted from the inverted monitor model m_1 . The global minimum of the Joint FWI objective in (5.10) is equal to the sum of the global minima of baseline (5.3) and monitor (5.4) of Parallel FWI. In addition there is no direct cross-talk between m_0 and m_1 . Taking the derivative of (5.10) with respect to m_0 does not involve m_1 and vice-versa. In Appendix C we demonstrate why Joint FWI results in a better time-lapse estimate than Parallel FWI despite these similarities.

Confidence map regularization

Recent studies by Yang (2014) and Willemsen et al. (2015b) (i.e. chapter 4) have investigated confidence maps to regularize the time-lapse inversion. These confidence maps are computed in the model space and locate regions where time-lapse change is likely given the baseline and monitor data. The confidence map introduced in chapter 4 minimizes the objective function

$$\hat{\chi}(m) = \|\hat{d}_0 - \mathbf{S}u(m)\|_2^2 + \|\hat{d}_1 - \mathbf{S}u(m)\|_2^2, \quad (5.11)$$

where we invert for a single model m , matching both the baseline and the monitor data. Intuitively we expect the model m to be an 'intermediate model' at locations where the true baseline and monitor models differ. While iteratively minimizing (5.11) with a local inversion method of choice, we save the gradients for the baseline term and for the monitor term. At locations of true time-lapse change, the baseline and the monitor gradients are likely to have opposing signs evaluated on the intermediate model. In other words, the baseline and monitor data attempt to perturb the intermediate model in opposing directions at locations of true time-lapse change. This suspected correlation between conflicting gradient information and regions of time-lapse change suggests that a confidence map can be obtained by looking at regions in the model where the baseline and monitor gradients consistently have opposing sign

during the inversion. One particular choice we look at is:

$$\beta_g(x) = \sum_i^n \left[|sgn(g_{0,i}(x)) - sgn(g_{1,i}(x))| \times \left(\frac{1}{2} (|g_{0,i}(x)| + |g_{1,i}(x)|) \right) \right], \quad (5.12)$$

where $\beta_g(x)$ is the gradient based confidence map, $g_{0,i}$ and $g_{1,i}$ are the baseline gradient and monitor gradient in iteration i respectively and sgn is the sign function. The first term within the summation is nonzero only when the gradients have opposing sign, corresponding to the idea that regions of time-lapse change are likely to have opposing baseline and monitor gradients when evaluated on the intermediate model. The second term acts as weighing with the magnitude of the gradient. The summation stacks over all n inversion iterations of (5.11) in order to reduce noise. The regularization term

$$\epsilon \left| \frac{m_0(x) - m_1(x)}{\beta_g(x)} \right|_m^2, \quad (5.13)$$

can be added to (5.10) or (5.9) to regularize the time-lapse inversion. It discourages updates in regions with low confidence β_g . In practice we use illumination compensation by Shin et al. (2001) to compensate for differences in gradient amplitude due to different illuminations. We normalize β_g to have a maximum value of 1.0 and we determine a minimum β_g value to avoid instability when dividing by very small β_g .

5.3 Results

We use the CDA, VDA and elastic data generated by the SeisSpace commercial code as input for the constant density P-wave inversion in PySIT. Figure 5-3 shows selected shot gathers for one of the sources at $x = 2.0$ km. The inversion grid we use has a uniform spacing of 12.5 m in both the horizontal and the vertical direction. Figure 5-1a shows the reference true P-wave velocity model model. Figure 5-4a shows the initial model from which we start the inversion. Realistically, the initial model should be constructed using standard methods for velocity analysis. Instead we heavily smooth

the true P-wave velocity model while preserving the water layer and average shallow sediment velocity. It is reasonable to know the depth of the water layer with high accuracy and the P-wave velocity contrast at the ocean bottom. Figure 5-4b shows the inverse crime results, where we both generate the true data on Figure 5-1a and invert using the PySIT code. Figure 5-4c displays the inversion results using CDA data generated by the SeisSpace code. The comparison of Figures 5-4b and 5-4c demonstrates the impact of the inverse crime on the baseline inversion. When using CDA data generated by the SeisSpace code we observe minor perturbations around source and receiver locations, wing-shaped artifacts around the gas clouds, a deterioration of the reservoir velocity and an over-all noisier velocity reconstruction. Figures 5-4d and 5-4e show how the addition of extra physics (VDA, EL) impact the inversion for P-wave velocity. Artifacts around the source and receiver locations increase in amplitude, especially for the VDA data. Due to the shallow ocean and the buried cables the wavefront comes in at a post-critical angle, resulting in a phase-shift depending on contrast in P-velocity, S-velocity and density. This phase shift is most prominent in the direct wave in the VDA shot gather in Figure 5-3b. The addition of variable density and S-wave velocity also deteriorates the reservoir reconstruction and the gas cloud outline. But overall the baseline velocity model still contains many of the characteristics of the true velocity model. We notice that the large amplitude shear wave events in 5-3c do not impact the inversion significantly. The events have a very small apparent velocity and will therefore not interact constructively with the simulated wavefield in the current model estimate, resulting in a negligible contribution to the FWI gradient.

The perturbations at source and receiver locations in Figures 5-4c - 5-4e can be prevented by masking and tapering the shallow part of the gradient. This prevents updates in the shallowest part of the model where the initial velocity model is quite accurate. We performed such a masked inversion but noticed that the overall quality of the inverted baseline model was slightly reduced. This suggests that the source-receiver artifacts in Figures 5-4c - 5-4e do not deteriorate the resolution at depth, but act primarily as fudge factors to compensate for differences in the forward models of

PySIT and SeisSpace. We decide not to use the masked inverted baseline models.

Ray tracing experiments show that even for large source-receiver offsets the diving waves are mostly constrained to the overburden sediment. This means that the chalk layer is predominantly sampled by reflections. We think this explains why the chalk layer in Figures 5-4b - 5-4e is inverted less accurately than the overburden. The chalk layer reflections add primarily high-wavenumber oscillatory updates to the smooth initial model. Without diving waves it is hard to add the larger scale features. With the baseline models available we now investigate the different time-lapse algorithms.

5.3.1 Parallel FWI

For Parallel FWI we first select an initial model. Here we compare two choices for the initial model m_i :

1. The smooth initial model from Figure 5-4a
2. The inverted baseline model selected from Figures 5-4b - 5-4e for the corresponding true data

Figures 5-5a and 5-5b show inverse crime results, where the initial model is the smooth initial model of Figure 5-4a or the inverted baseline model of 5-4b respectively. As a comparison we also show the CDA data generated by the SeisSpace code. These results are found in Figures 5-5c and 5-5d. Compared to the true P-velocity time-lapse in Figure 5-2a we observe a lot of artifacts, especially when we avoid the inverse crime in Figures 5-5c and 5-5d. This would only get worse when additional physics (VDA, EL) is introduced. This is no surprise, as mentioned earlier it is well known that Parallel FWI does not yield desirable results. But since it is a straight-forward application of FWI to the time-lapse problem it serves as reference. Figure 5-6 shows the corresponding vertical profiles through the time-lapse sections.

5.3.2 DDFWI

Figures 5-7a - 5-7d display the DDFWI time-lapse inversion results on their corresponding baseline models from Figures 5-4b - 5-4e. To obtain these results we first subtract the baseline data from the monitor data to obtain the true time-lapse signal δd for each of the four scenarios we investigate. Then this data difference is used in equation 5.9. Even though we only show the time-lapse results within a zoomed-in section of the full domain, the inversion updates the full computational model. Figure 5-7a demonstrates the inverse crime DDFWI result. Figures 5-7b - 5-7d demonstrate the impact of increasingly realistic physics in the true data on the acoustic approximation. By comparing Figures 5-7a and 5-7b we see that the inverse crime dataset gives slightly better results than the CDA dataset, but the impact is limited. DDFWI clearly gives a more accurate reconstruction than the Parallel FWI results in Figure 5-5a - 5-5d.

The VDA dataset adds variable density effects to the CDA dataset. Using this VDA dataset we obtain the DDFWI results of Figure 5-7c. Finally, when using the EL dataset we obtain Figure 5-7d. We see that this process of adding physics to the true data results in a deterioration of the time-lapse estimate using the acoustic approximation. This is not surprising since the mismatch between the acoustic approximation and the physics of the true data increases. Vertical profiles through these Figures tell the same story in Figure 5-8. In this figure we see that the amplitude of the reconstruction at the depth 2.85 km gradually decreases when transitioning from inverse crime data to EL data. In the EL DDFWI we also observe that the positive overshoot at depth of 3.0 km has larger amplitude than the negative velocity change at the true perturbation at depth 2.85 km. Not only do we significantly underestimate the true time-lapse change, we also introduce an artifact which is more prominent than this change. Figure 5-7 shows that the time-lapse estimate becomes more smeared out with increasing data complexity.

5.3.3 Joint FWI

The Joint FWI (J-FWI) results are very similar to the DDFWI results in this study. This similarity may be explained by the observation in Appendix C, where we show that similar to DDFWI, Joint FWI directly inverts for the true time-lapse signal δd , at least in the first iteration. Figure 5-9 displays the results for the EL data. We again observe a highly smeared time-lapse estimate with sidelobes of comparable amplitude to the estimate at the true reservoir time-lapse change location at depth 2.85 km. A vertical profile in Figure 5-10 compares the elastic DDFWI and J-FWI. This figure shows that both algorithms produce estimates that are very similar for the EL data in this study. Both results significantly underestimate the true time-lapse change at the reservoir and both produce a very large amplitude artifact directly below it. Both algorithms started from the elastic baseline model of Figure 5-4e and both use the same elastic true data. In order to better understand these artifacts due to the elastic data, we now investigate the impact of both the elastic physics and the quality of the elastic initial model on the time-lapse estimates.

5.3.4 Elastic data

Comparing Figures 5-7c and 5-7d, we observe that DDFWI deteriorates significantly when the true data is elastic. A very similar undesirable result is observed when we use J-FWI on the elastic data. This raises the question if the elastic effects in the data cause the significant deterioration. To investigate this hypothesis we inspect the true data more closely.

Figure 5-11a zooms in on a section of the VDA shot gather of Figure 5-3b at large offset. In this window it is very similar to the CDA shot gather which is not plotted here. The most prominent event in Figure 5-11a is the top of reservoir reflection. Similarly, Figure 5-11b is a zoom of the EL shot gather of Figure 5-3c. When comparing Figures 5-11a and 5-11b we observe a striking difference in the top of reservoir reflection at this large offset. The polarity appears reversed and the arrival time of the peak is different. Not visible in this Figure is that at smaller source-

receiver offsets the CDA and EL top of reservoir reflections gradually become more similar in both the shape and arrival times of the peak. Within an offset range of approximately 2.0 km the top of reservoir reflection is virtually indistinguishable in the CDA, VDA and EL gathers. Ray tracing experiments on the true velocity model suggest that source-receiver offsets of roughly 3.0 km hit the reservoir at the critical angle. Zhu and McMechan (2012) demonstrate that spherical wavefronts will start experiencing phase-shifts at incidence angles before the critical angle. This phase shift grows and approaches 180 degrees (polarity reversal) at very large (i.e. near-flat) incidence angles. The transition of approximately identical reflections at near offset to an approximate polarity flip at large offsets in our data, together with the ray tracing observations, suggest that the top of reservoir reflections become post-critical.

Figure 5-11c shows part of the VDA time-lapse gather δd_{VDA} , where the baseline P-velocity and density are perturbed by the true time-lapse from Figures 5-2a and 5-2b respectively. Figure 5-11d shows the EL equivalent δd_{EL} , where in addition the S-velocity has time-lapse from Figure 5-2c. Notice that the elliptical overburden perturbations are only present in the true P-velocity time-lapse. Their effect on a top of reservoir reflection passing through them is a time-delay. Because the postcritical top of reservoir reflection is not the same in the VDA and EL baseline shot gathers, delaying it causes the time-lapse shot gathers δd_{VDA} and δd_{EL} to be different as well. In addition, time-lapse changes in the reservoir change the material contrast at the chalk layer. As a result the amplitude and phase-shift of the post-critical reflections change in Figures 5-11a and 5-11b, as predicted by the Zoeppritz equations. This results in an additional difference between δd_{VDA} and δd_{EL} . For the near-offset top-of-reservoir reflections where the baseline signals are identical, δd_{VDA} and δd_{EL} are close to identical.

It is important to realize that the norm in the DDFWI objective function in (5.9) represents integration of the residual over time for each source-receiver combination and a subsequent summation over all sources and receivers. Figure 5-12 plots the contribution to the EL DDFWI objective function of each source-receiver combination before this summation. In the first iteration of DDFWI our starting model is the

inverted baseline model and the residual is the true time-lapse signal δd . Figure 5-12 therefore shows that δd is particularly large for certain sources around 2 km and receivers around 8 km and the other way around through reciprocity. The large amplitude post-critical top of reservoir reflection experiences the time delay of the large elliptical overburden perturbation on the right and the perturbations at the reservoir, resulting in large amplitude δd_{EL} . The left overburden perturbation has a much smaller velocity change and covers a much smaller area, making its impact on the reflection travel times relatively small.

The EL DDFWI inversion result, whose zoomed in section is shown in Figure 5-7d, has a lower objective value (i.e. the sum over all the source-receiver contributions) than the starting model. This does not imply that the objective function contribution of each source-receiver combination improved uniformly. Figure 5-13 shows the logarithm of the relative change of each source-receiver contribution. Source-receiver combinations represented by red colors have been improved (i.e. decreased objective value contribution). Blue colors represent source-receiver combinations whose objective value contribution has actually gone up. We use a logarithmic plot of the difference, because without it a plot of the difference will be dominated by the improvement at the locations of the large amplitude blobs in 5-12. This illustrates the dominant effect on the inversion of the post-critical reflections experiencing the reservoir change and passing through the right elliptical overburden perturbation. We see that the elastic DDFWI time-lapse estimate corresponding to Figure 5-7d reduces the initially large objective function contribution of these events by approximately a factor of 100. But we demonstrate in Figure 5-11 that these events are different in δd_{EL} than they are in δd_{VDA} because the phase shift at the reservoir depends on the elastic parameters. Since the EL DDFWI inversion inverts for these large amplitude differing events in particular we should expect different results compared to VDA or CDA DDFWI inversions. This is what we observe when we compare Figures 5-7b and 5-7c to 5-7d. We also observe in Figure 5-13 that at the end of the inversion a significant share of the near-offset source-receiver combinations has deteriorated. This includes a substantial number of source-receiver combinations centered around

the reservoir time-lapse change between $x = 3.5$ km and $x = 4.5$ km. In other words, the DDFWI inversion focuses on the large amplitude δd events corresponding to post-critical reflections, but when matching these many near-offset reflections deteriorate. The negative impact of post-critical reflections on the baseline inversion using the acoustic approximation has been documented by for instance Mulder and Plessix (2008). We observe that the impact on the time-lapse inversion is more obvious than on the baseline inversion in this study.

5.3.5 Limiting the maximum offset

We now use the observation that the near offset δd is similar in CDA, VDA and EL data, unlike the post-critical reflections. This similar δd suggests that we should obtain similar results when inverting for near-offset EL time-lapse data as when using near-offset VDA or CDA data. Figure 5-14 shows the DDFWI objective function contribution when inverting for the EL data with max source-receiver offset of 2.0 km. The initial objective function contributions still correspond to Figure 5-12. We now only invert for the near-offsets for which δd_{EL} is similar to δd_{VDA} and δd_{CDA} . With this approach we avoid the post-critical reflections we cannot match with acoustic physics. In this way we hope to avoid the elastic artifacts which were apparent in the full offset elastic DDFWI result of Figure 5-7d.

Figure 5-15 shows the EL DDFWI time-lapse estimate when limiting the maximum offset to 2.0 km. Compared to full offset equivalent in Figure 5-7d we observe an improved reconstruction at the reservoir. In the full offset DDFWI result of Figure 5-7d we observe that the maximum amplitude in the reservoir region has positive sign (blue), even though the true time-lapse change at the reservoir in Figure 5-2a is predominantly negative (red). When we limit the maximum offset to 2.0 km in Figure 5-15 the largest amplitude is negative (red), as is the true reservoir time-lapse. Despite these improvements, the reconstructed V_p still has much lower amplitude than in the true time-lapse perturbation and the location is also slightly too deep. We also note that the reconstruction of the overburden time-lapse perturbations deteriorates in Figure 5-15 compared to when we use all offsets in Figure 5-7d. By limiting

the maximum offset range we potentially lose diving waves and deeper reflections passing through the perturbations. Instead of limiting the offset to remove the post-critical part of the top-of-reservoir reflection, a better approach would be to mute the post-critical part of the top-of-reservoir reflection. This leaves all other beneficial large-offset events like diving waves intact. The only reason we limit offsets is that it is easier to implement in this numerical investigation.

Figure 5-15 marks an oscillatory perturbation by an ellipse. This event is less pronounced when all offsets are used and our interpretation is that it caused by the insufficient reconstruction of the right overburden perturbation. For instance, a zero offset top-of-reservoir reflection will pass through the right overburden perturbation, hit the reservoir, and will then return to the surface by passing through the overburden perturbation again. In the true data the right overburden perturbation delays the top of reservoir reflection right below it by a small amount, resulting in a δd concentrated around an arrival time of the top of reservoir reflection. This will be backpropagated as a reflector in the time-lapse estimate in the ellipse, unless the right overburden perturbation would accurately delay the top-of-reservoir reflection on the inverted baseline model by the required amount. But we observe that this is not the case in Figure 5-15, because the amplitude of the right overburden perturbation is much smaller than in the true time-lapse of 5-2a.

5.3.6 Role of the initial model

So far we did EL DDFWI on the EL baseline model of Figure 5-4e. This EL baseline model is the result of using the elastic data over the full offset range. This EL baseline inversion of Figure 5-4e therefore includes the post-critical reflections. The acoustic baseline inversion attempts to match the phase shift and eventually polarity reversal of the large amplitude top of reservoir reflection at large offsets. Since we make the acoustic approximation and invert for only V_p , we cannot accurately model the phase of the post-critical reflection. The result is that the baseline velocity model of Figure 5-4e contains inaccuracies that compensate for this, even if they are not immediately obvious by just looking at the figure. We therefore want to investigate how this EL

baseline model affects the EL DDFWI inversion. The way we investigate this is by replacing the EL baseline model with a better baseline model not polluted by the post-critical reflections. In this study we choose as this better baseline model the inverted CDA baseline model of Figure 5-4c. This baseline velocity model can be seen as the result of a hypothetical scenario where processing perfectly removed the elastic effects from the baseline model. EL DDFWI on this baseline model is therefore an indication of the biggest improvement we can expect by using a better baseline model. With this good baseline velocity model we again investigate DDFWI using the elastic data. As before we limit the maximum offset to 2.0km to avoid the post-critical reflections. Figure 5-16 shows that the amplitude at the reservoir is closer to the true amplitude now. At the same time, the inaccurate event in the ellipse of Figure 5-4e is still there in Figure 5-16. We interpret this as being caused by the inadequate reconstruction of the right overburden perturbation, which is still the case here.

5.3.7 Gradient based confidence map

While not the main objective of this study, regularization is an important component of a practical time-lapse FWI workflow. In this study we investigate the gradient based confidence map using the EL baseline and monitor data. We perform 20 inversion iterations on equation 5.11 while starting from the velocity model in Figure 5-4e. While minimizing the objective function we record the baseline and monitor gradients at each iteration. Using (5.12) we construct the confidence map whose zoomed in section is displayed in Figure 5-17 after normalization so that the maximum value is 1.0. In Figure 5-18 we plot the evolution of the illumination compensated baseline and monitor gradients at four pixels indicated by four stars with different colors. The blue star is at a pixel at the top of the reservoir in the large amplitude time-lapse region. The red star is in the thin and low amplitude left overburden perturbation. The magenta star is in the right overburden perturbation and the yellow star is in a region where there is no time-lapse change. The yellow star lies in the region where and artificial time-lapse reflector is seen in Figures 5-15 and 5-16. Regions of high

confidence such as the those represented by the blue and magenta stars exhibit large amplitude gradients with opposite sign as expected. Regions of low confidence (e.g. red and yellow star) tend to have gradients with low amplitude and the same sign.

We see in Figure 5-17 that we have high confidence that there is a time-lapse change around the boundary of the reservoir. In addition we have high confidence at the right overburden perturbation area. We also have high confidence in an incorrect wing-shaped region that comes in from the top-left to the reservoir and then moves to the top-right of Figure 5-17. This feature extends for some distance outside of this zoomed-in section and moves towards the surface at $x = 2.0$ km on the left and $x = 8.0$ km on the right. Recall the large amplitude event in Figure 5-12 at $x = 2.0$ km and $x = 8.0$ km which we interpreted as being the post-critical top of reservoir reflection being perturbed by the reservoir perturbation and delayed by the right overburden perturbation. The gradient sign is not only opposite at these true time-lapse regions, but also along the remainder of the first Fresnel zone of the post-critical top of reservoir reflection. We believe this explains the wing-shaped high-confidence artifact in Figure 5-17. We also see that the confidence map incorrectly gives low confidence at the left elliptical overburden perturbation represented by the red star. We invert with a 6.0 Hz peak frequency Ricker wavelet without frequency marching. The dominant wavelength in the region around the left overburden perturbation is therefore 350 m - 400 m. The travel time delay caused by this thin (≈ 25 m) and low amplitude time-lapse change has only a small effect on these long wavelengths. In the full offset EL DDFWI result of Figure 5-7d we also observe that the left overburden perturbation is also hard to detect.

Despite the imperfections in the confidence map we use it to regularize DDFWI through (5.13) using exactly the same setting as we used for Figure 5-16. The zoomed in section can be seen in Figure 5-19. In the ellipse in the bottom right corner we see that the artificial event which was present in Figures 5-15 and 5-16 is significantly weaker. The right overburden perturbation is very weakly recovered, just as in Figures 5-15 and 5-16 when we also limited the max offset to 2.0 km. In Figure 5-20 we plot vertical profiles through Figures 5-15, 5-16 and the new Figure 5-19 to compare them.

We see that the addition of the confidence map does not make the time-lapse estimate in Figure 5-19 differ that much from Figure 5-16 along the vertical profile. The left overburden reconstruction has slightly deteriorated, but the amplitude estimate at the reservoir has improved and the positive sidelobe right below it has been truncated.

5.4 Discussion

The results we show demonstrate the effect of more realistic physics in the data on an acoustic time-lapse inversion. To keep the number of scenarios manageable we restricted parts of the workflow. For instance, we did not investigate the role of bandpass filtering the wavelet and the data used in inversion. In all time-lapse inversion results displayed in this chapter we have difficulty inverting for the left elliptical overburden time-lapse perturbation, even with DDFWI under the ideal conditions of inverse crime data as can be seen in Figures 5-7a and 5-8. The peak frequency of our Ricker wavelet is 6 Hz with a corresponding dominant wavelength of more than 300 m in the 2000 m/s sediments. The left elliptical overburden perturbation has a thickness of approximately 25 m and is therefore much smaller than the dominant wavelength. In addition to this, the amplitude of the time-lapse change is also very small. Just as in conventional FWI, a frequency marching scheme where the bandpass filter is incrementally moved to higher frequencies as the model improves may be helpful. Moving the bandpass to higher frequencies results in shorter wavelengths that are more sensitive to travel time delays and thin reflectors. The effects of such a procedure on the time-lapse inversion results is not included in our analysis.

When limiting the maximum offset to 2.0 km in Figures 5-15, 5-16 and 5-19 we observe that the reconstruction of the right overburden perturbation deteriorates significantly. When limiting the maximum offset we may lose potential diving waves or deeper reflections in addition to the desired loss of the large-amplitude postcritical reservoir reflection passing through the right overburden perturbation. Without diving waves and deeper reflections it is harder to accurately reconstruct the amplitude and shape of this perturbation. A better alternative is to specifically mute the prob-

lematic post-critical reflection without limiting the overall maximum source-receiver offset.

The baseline model in Figure 5-4e is obtained using the full offset EL data. The inversion therefore included the post-critical reflections in the inversion. Figures 5-11a and 5-11b demonstrate that this top-of-reservoir reflection has a different phase in the EL data and that the peak is centered around a slightly different time. We therefore expect that the inverted baseline model contains (potentially kinematic) artifacts. We demonstrate the importance of the baseline model on DDFWI in Figures 5-15 and 5-16, by exchanging the EL baseline model for a better one. In this study we choose the CDA baseline model as this better baseline model. The CDA baseline model represents the optimal baseline model that would (hypothetically) have been obtained if we had been able to perfectly remove all elastic effects in the baseline inversion. Using this CDA baseline model, DDFWI inversion of the near-offset EL data results in a significantly better time-lapse estimate in the reservoir. In reality the baseline model will always contain some artifacts as a result of the elastic effects in the data, even if post-critical reflections are carefully removed. But when the baseline model is closer to what would have been obtained on data with problematic physics perfectly removed, we demonstrate that better DDFWI results are expected.

The confidence map β_g we compute was obtained by using the full offset elastic data. It therefore contains the problematic large-amplitude post-critical top-of-reservoir reflections. This event passes through the right elliptical overburden perturbation. When minimizing (5.11) we approach an intermediate model whose top-of-reservoir reflection will be slightly delayed compared to the monitor data and advanced compared to the baseline data. The baseline and monitor gradients are therefore of opposing sign in a region more or less surrounding the first Fresnel zone of the post-critical top-of-reservoir reflection. This causes an artifact in the confidence map β_g in Figure 5-17. Despite this artifact, β_g still has advantages over uniform regularization corresponding to a confidence map β_g which is uniformly 1.0 everywhere. Even though we have confidence in regions where there is no real time-lapse change, we do exclude many regions where updates might otherwise take place, such as the

fake time-lapse reflectors in the ellipse of Figure 5-15. A more problematic scenario takes place when the confidence map assigns low confidence to regions of true time-lapse change. This is what happens to the left overburden perturbation. The large dominant wavelength corresponding to the 6 Hz peak frequency Ricker wavelet is not sensitive enough for the left overburden perturbation with the small velocity change. Time-lapse regularization is still an active topic of research. In addition to using confidence map based methods it would be interesting to investigate the effectiveness of Total Variation regularization (Maharramov and Biondi, 2014). This regularization is very effective in removing oscillatory artifacts in the time-lapse estimate.

The analysis in this chapter is specific to a synthetic representation of a real field in the North Sea. The observations made in this synthetic study are helpful when designing an effective study on the real field. Most importantly, in our synthetic study we see the detrimental effect of the post-critical top of reservoir reflections in the field we study. Without this knowledge, a straight-forward application of DDFWI with the acoustic approximation on the associated field data may result in artifacts at the reservoir. In a real field study, knowledge of the true time-lapse change is not perfect. We may not be able to identify artifacts for what they are. For instance, the oscillary event in the ellipse of Figure 5-15 might incorrectly be interpreted as a real time-lapse change. Our synthetic analysis demonstrated that this event can be caused by the incorrect recovery of the overburden time-lapse change. By performing a synthetic time-lapse study before using the field data, we can design a workflow with reduced artifacts when using the synthetic data. The hope is that these observations would translate to the real field data.

To get more confidence in our synthetic study, some of the simplifying assumptions made in this chapter should be undone. For instance, the study should be extended to 3D. Instead of starting the baseline inversion from a heavily smoothed true model, the initial model should be obtained from tomography. In this chapter we did not use a band-pass filter to do frequency marching, which should be done in a more realistic study. In addition, free surface boundary conditions should be used so the effects of the shallow water multiples on the time-lapse inversion can be come clear. In reality

the depth and velocity of the water column may also change between the baseline and monitor survey. Another way to increase the realism of the synthetic study would be to add anisotropy and attenuation to the true data and investigate their impact on the time-lapse reconstruction.

5.5 Conclusions

In this chapter we investigated the effects of increasingly realistic data on an acoustic inversion for the P-wave velocity. The performance of different FWI based time-lapse algorithms is compared on a realistic geometry. We observe that the time-lapse estimate deteriorates significantly because of the phase-shift in the elastic post-critical reflections. By limiting the offset range we exclude these events from the inversion and obtain a better time-lapse estimate at the reservoir. We then demonstrate that the importance of the initial model and show how a confidence map can be used to remove some of the artifacts in this particular model. We also demonstrate the impact of the inverse crime by comparing inversion for true data generated by the inversion code (PySIT) with true data generated with a different code (SeisSpace).

5.6 Acknowledgements

The work I present in this chapter is based on my summer internship at ConocoPhillips in 2015. Together with Baishali Roy and Jun Cao I compiled the main observations in an SEG abstract (Willemsen et al., 2016a). I really appreciate their insightful analysis and also comments about both the science and the presentation. After the internship finished I independently wrote this chapter to add more theory and background information. Any inaccuracies are solely the result of my work. I am grateful to ConocoPhillips for permission to publish the work of the summer internship. I would also like to thank MIT and the ERL consortium members for funding. In this paper the Python Seismic Inversion Toolbox (PySIT) was used for the inversion of the true data generated by SeisSpace.

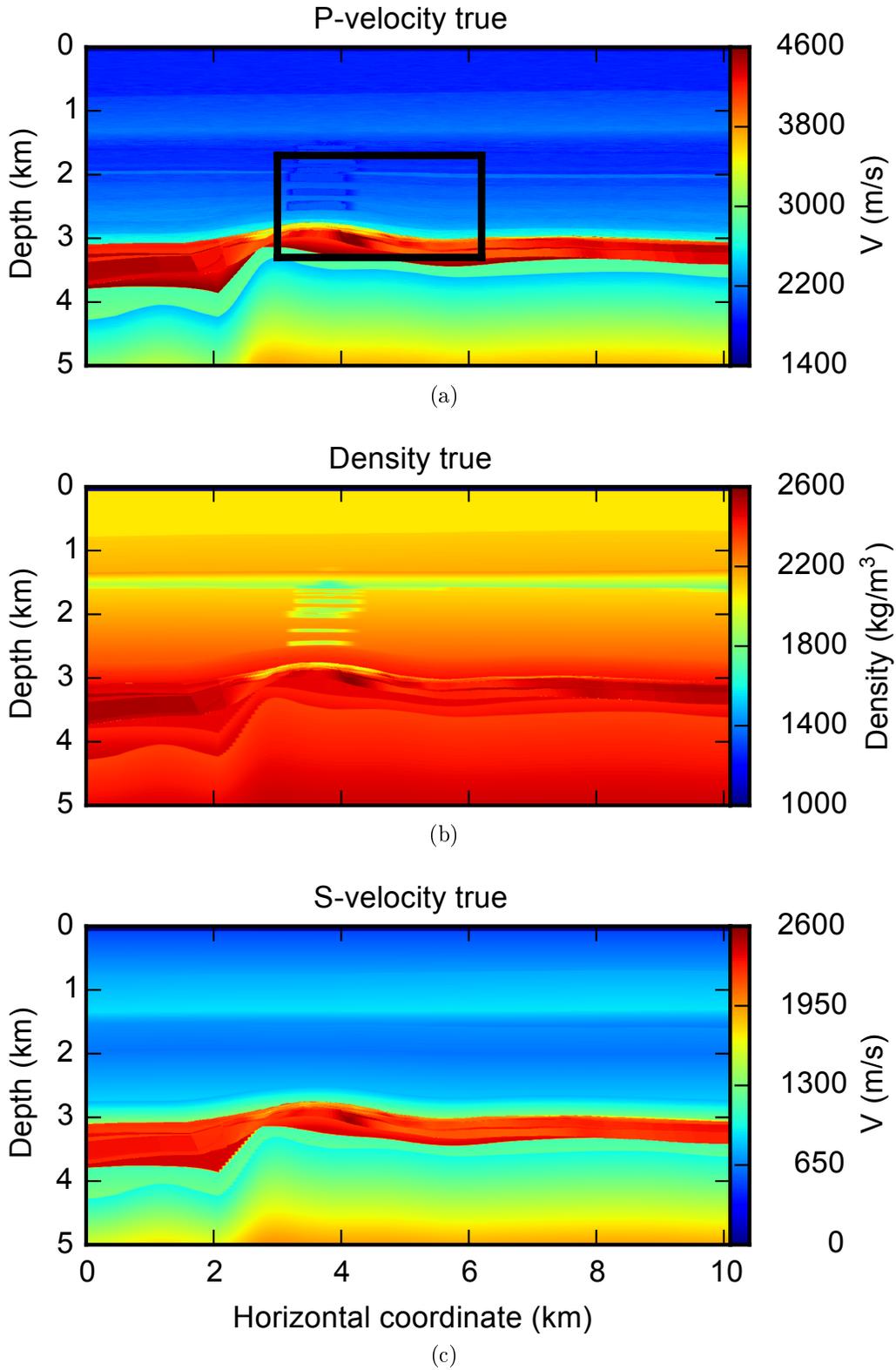


Figure 5-1: True models with 2.0 m vertical spacing and 12.5 m horizontal spacing: (a) V_p (b) ρ (c) V_s . The black box marks the region on which the time-lapse study primarily focuses.

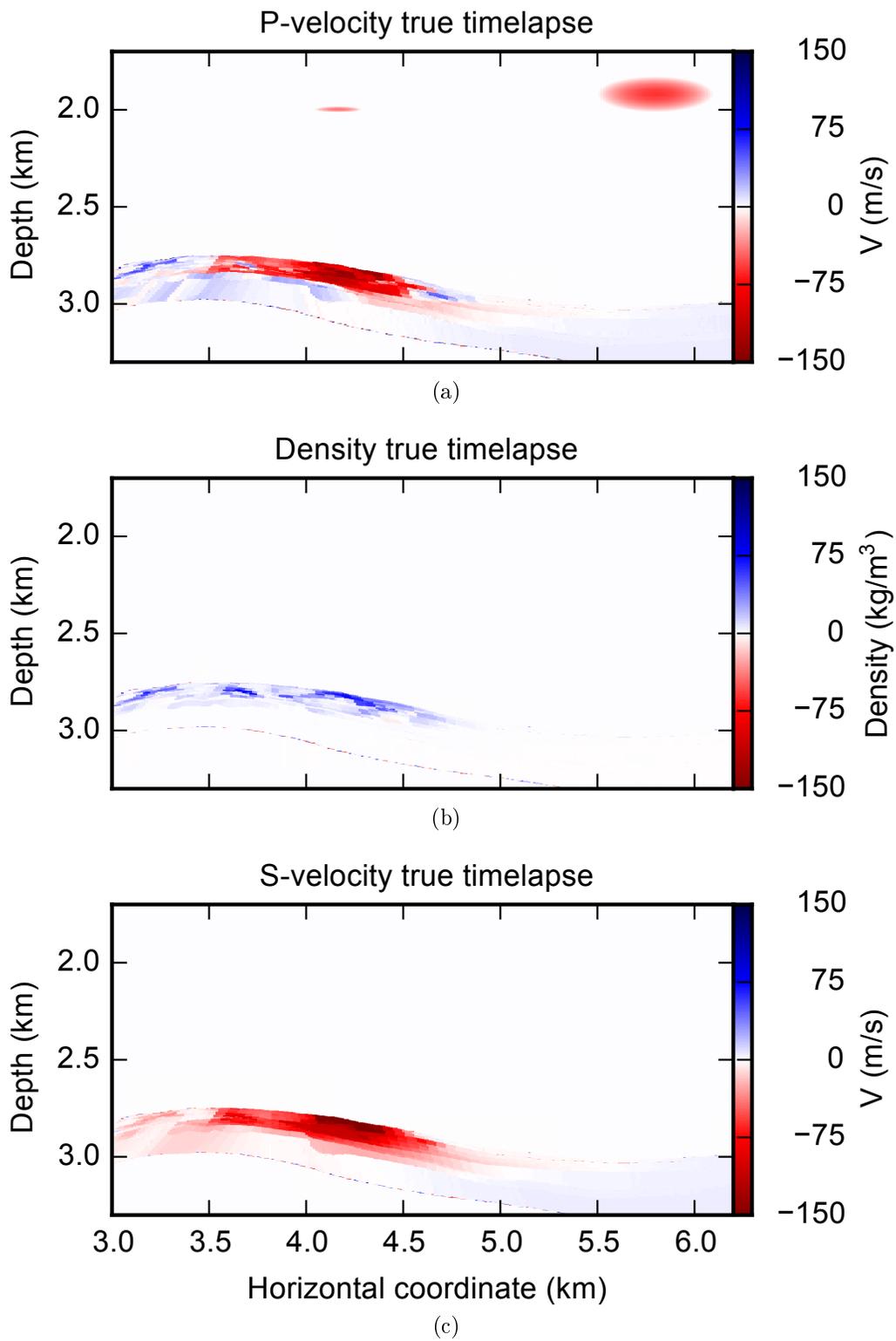


Figure 5-2: True time-lapse zoom corresponding to black box in Figure 5-1a: (a) V_p (b) ρ (c) V_s .

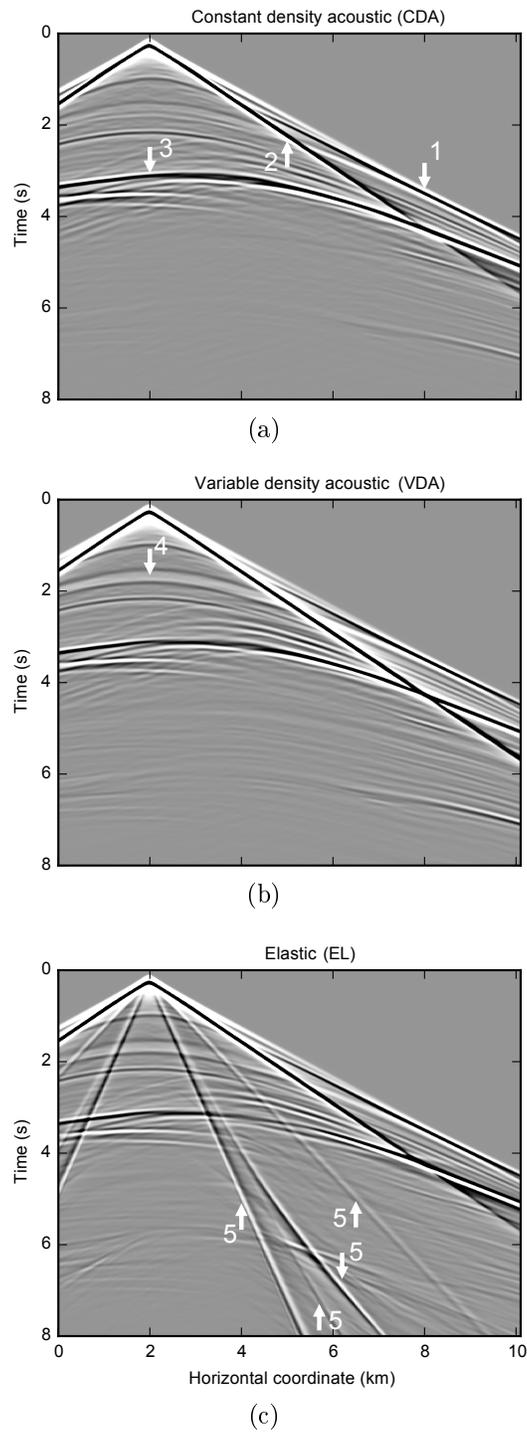


Figure 5-3: Baseline pressure shot gathers for a) CDA, b) VDA and c) elastic for a shot at $x = 2.0$ km. Several events of interest are marked with numbers. 1) The head wave propagating along the bottom of the ocean. 2) Direct wave. 3) Top of reservoir reflection. These three events are visible in all three simulations. 4) Reflection on low density layer. 5) Slow events due to introduction shear velocity.

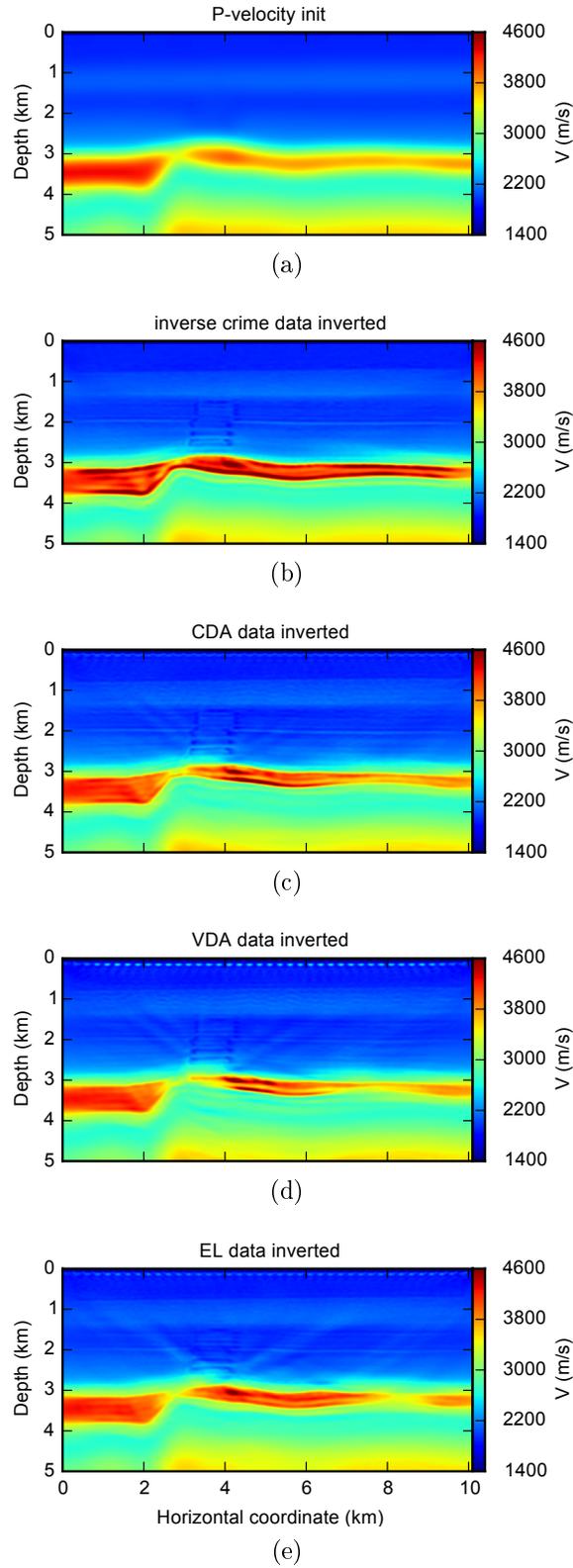


Figure 5-4: Baseline CDA inversion results starting from initial model (a) for different types of input data. (b) inverse crime data. (c) CDA data. (d) VDA data. (e) EL data. Figure 5-1a shows the true P-wave velocity.

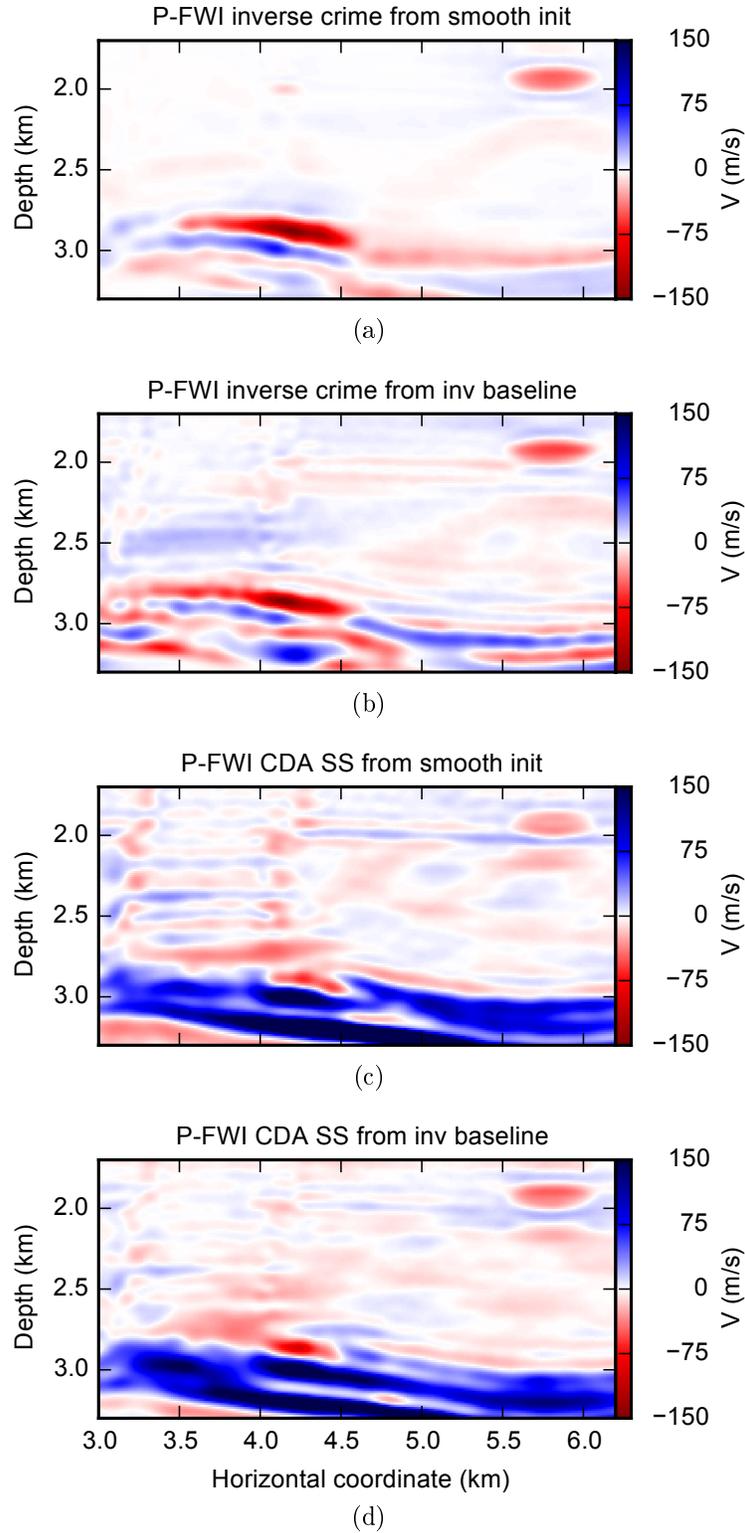


Figure 5-5: Parallel FWI for P-wave velocity. (a) and (b) use inverse crime data and (c) and (d) use CDA data from SeisSpace: (a) and (c) start from the smooth initial model of Figure 5-4a, (b) starts from Figure 5-4b and (d) starts from Figure 5-4c. True reference is displayed in Figure 5-2a.

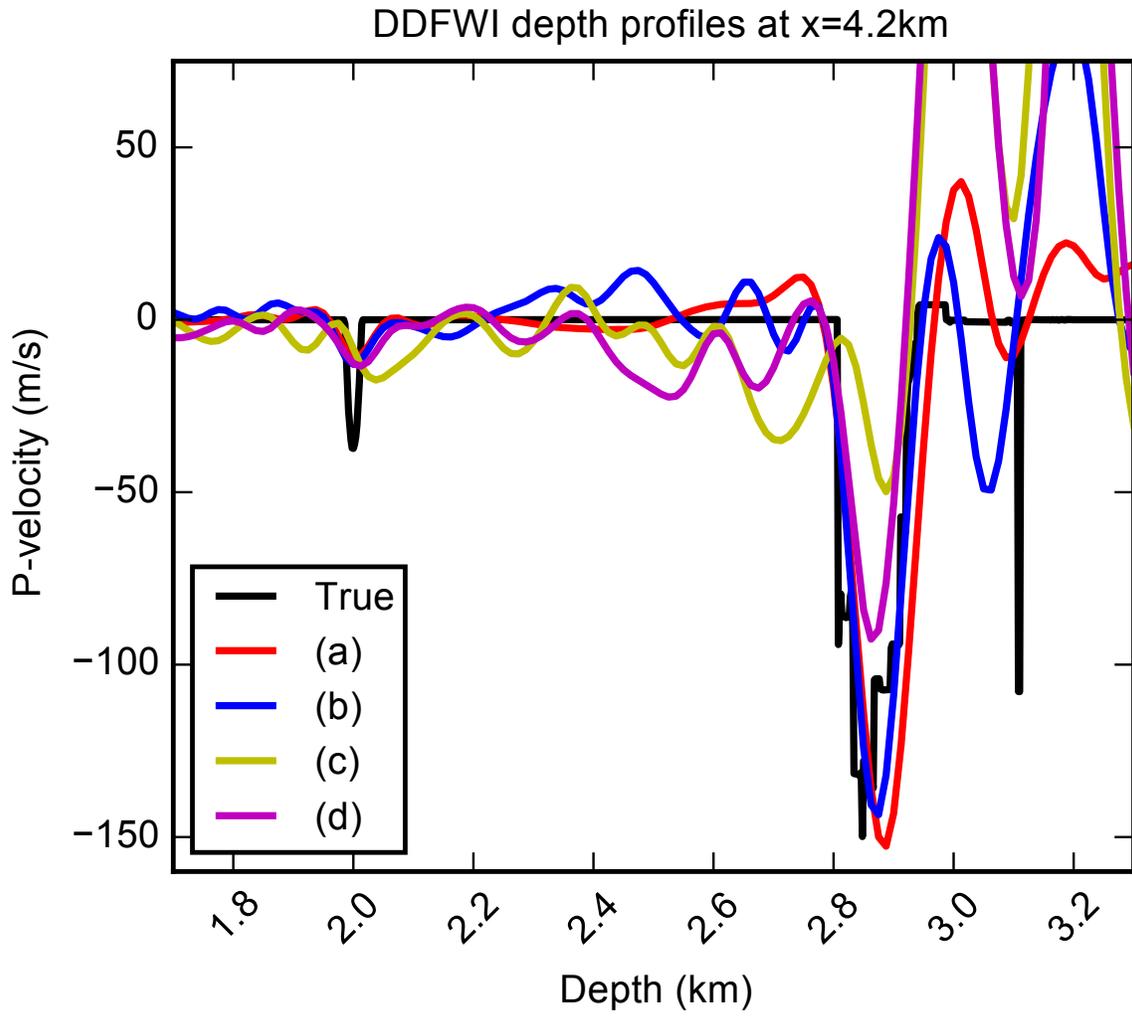


Figure 5-6: Parallel FWI depth profiles with labels referring to Figures 5-5a - 5-5d.

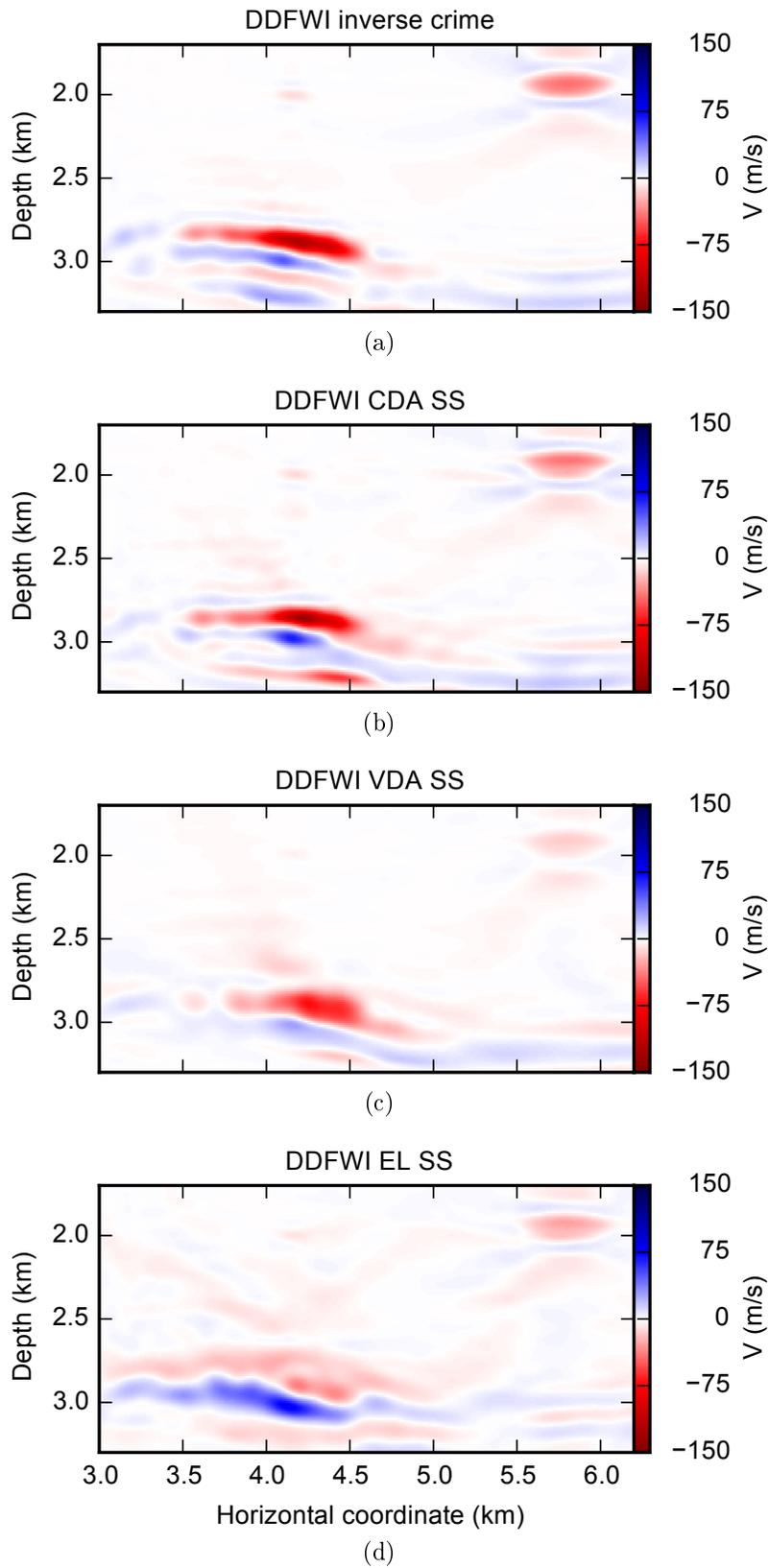


Figure 5-7: DDFWI results for the inverse crime, and the three SeisSpace datasets (CDA, VDA and EL) using the corresponding baseline models from Figures 5-4b - 5-4e.

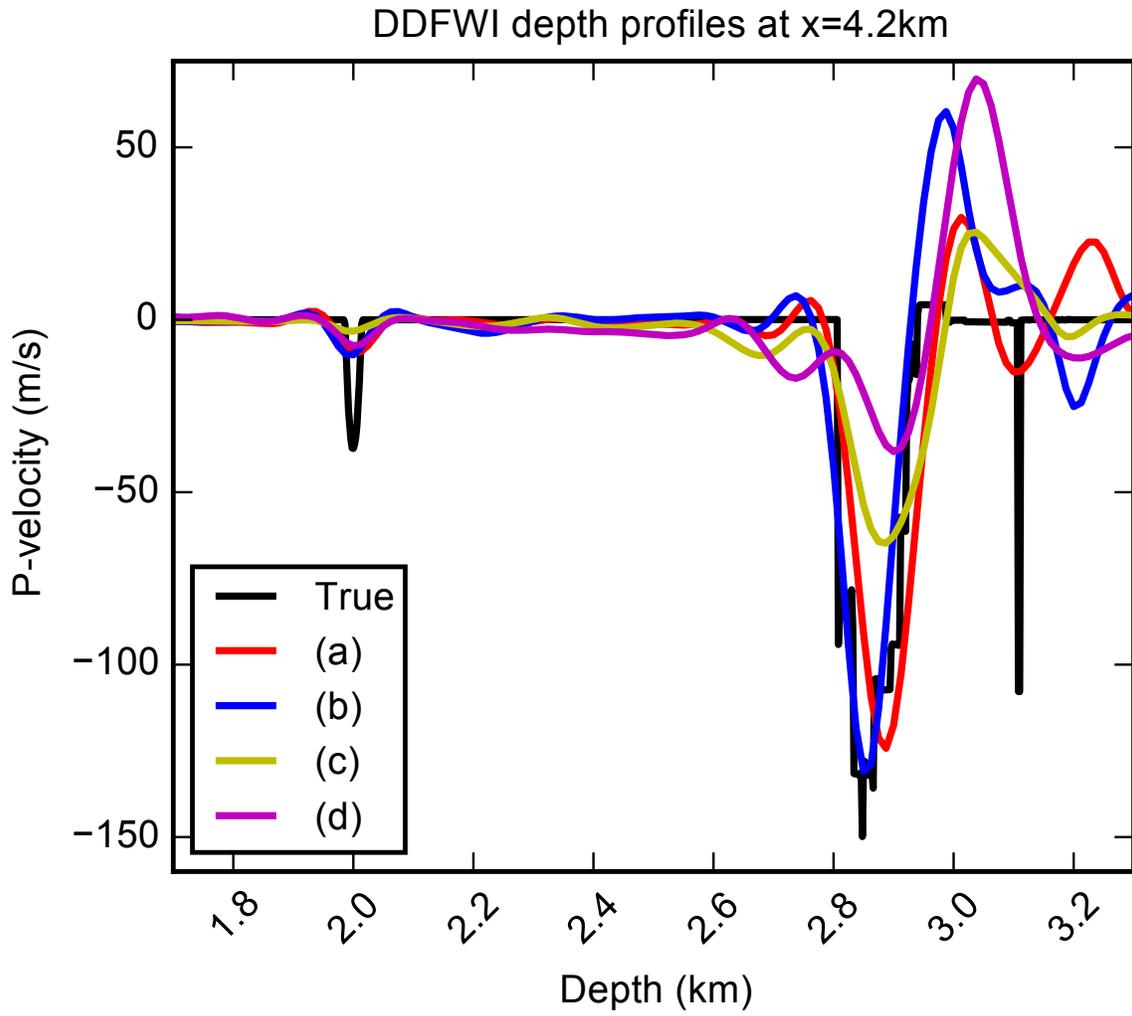


Figure 5-8: DDFWI depth profiles with labels referring to Figures 5-7a - 5-7d.

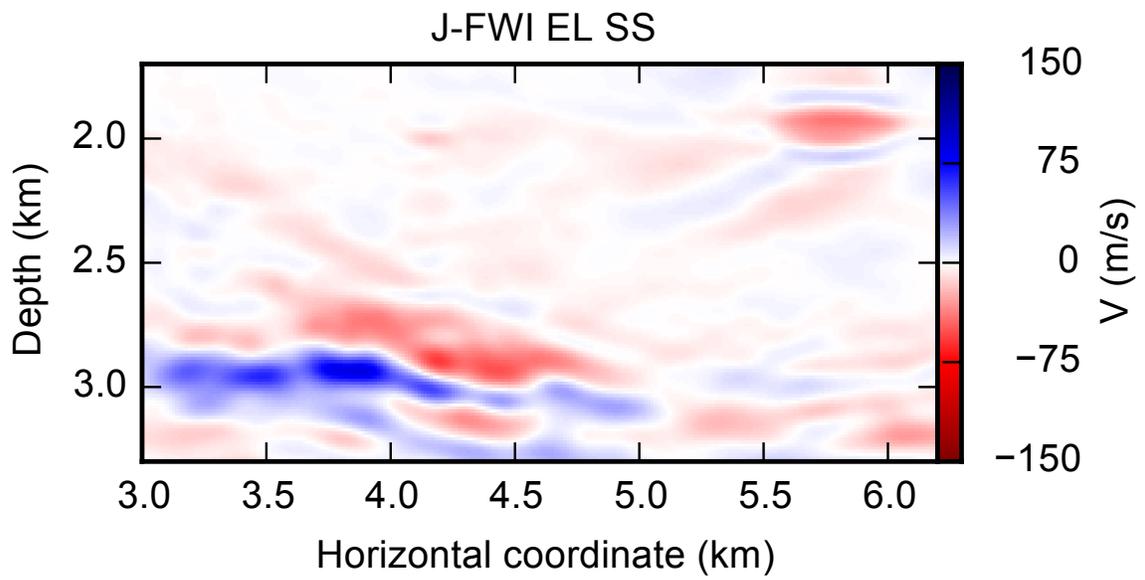


Figure 5-9: Joint FWI for elastic data.

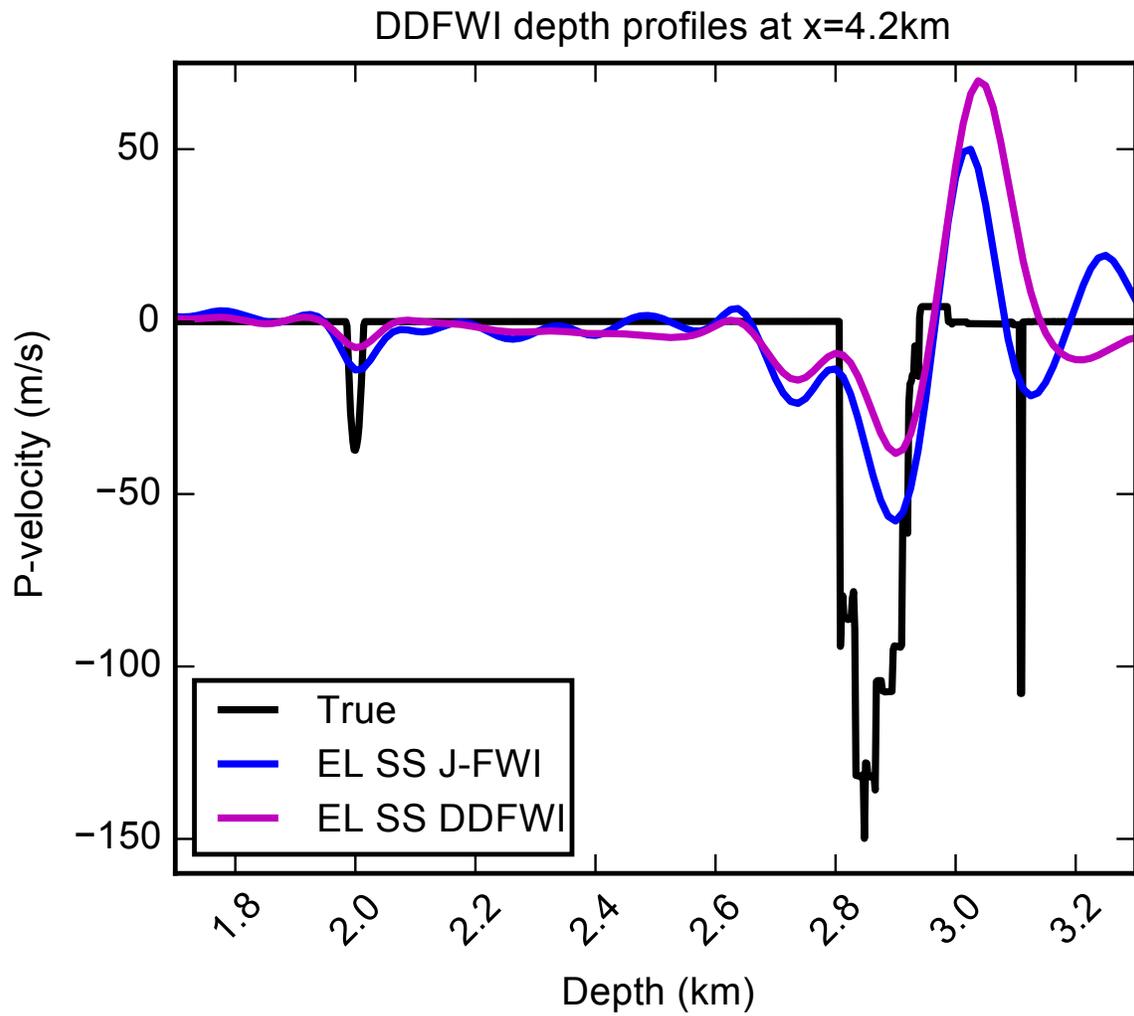


Figure 5-10: Depth profiles comparing the elastic DDFWI and Joint FWI time-lapse estimates along a vertical profile.

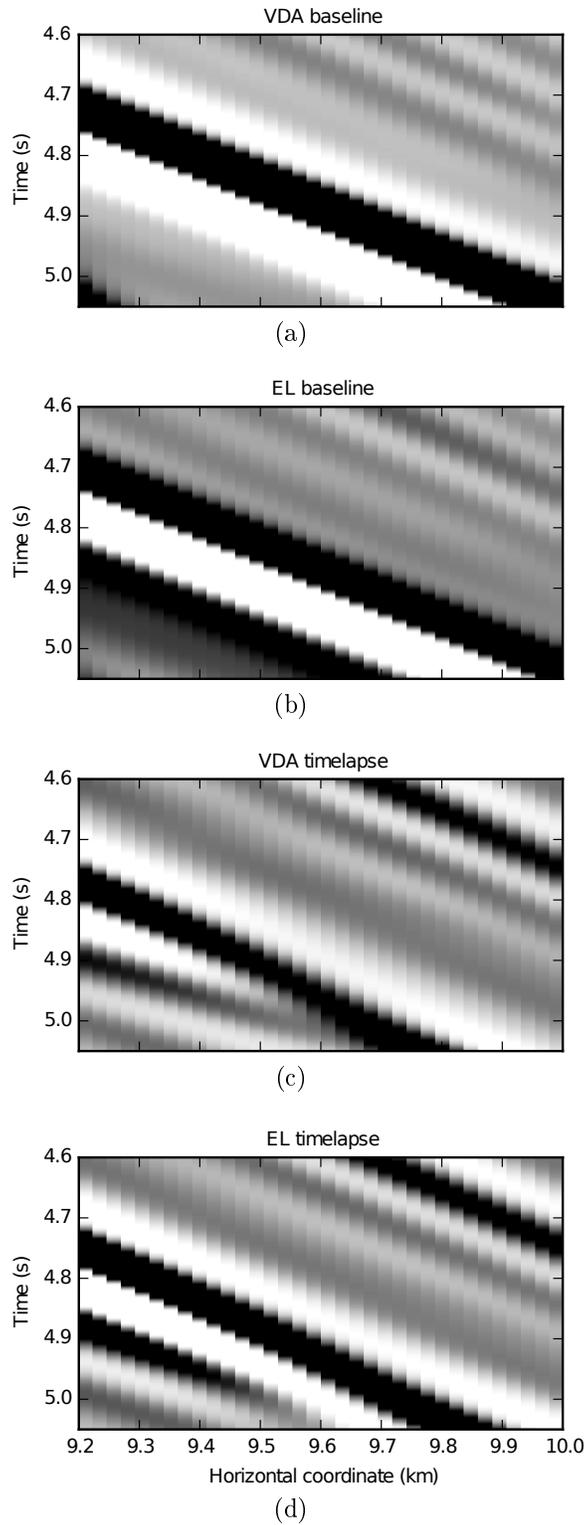


Figure 5-11: Zoom in on large-offset top of reservoir reflection for earlier pressure shot gathers in Figures 5-3b and 5-3c. In Figures 5-11a and 5-11b we see that these events have large differences in phase in the VDA and EL data. We also see that the time-lapse shot gathers δd_{VDA} and δd_{EL} in Figures 5-11c and 5-11d are slightly shifted in time and have slightly different shape.

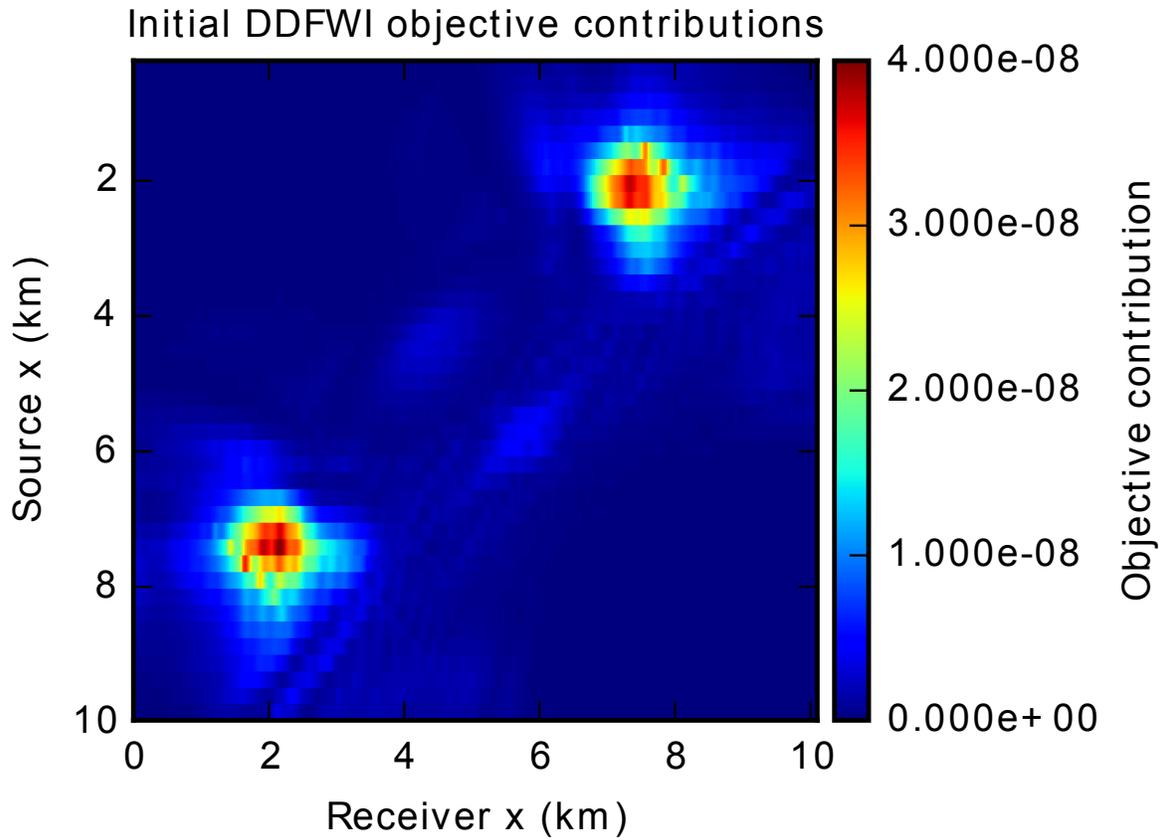


Figure 5-12: Each source-receiver combination contributes to the DDFWI objective function of (5.9) with the EL data. This Figure shows the results on the initial (i.e. inverted baseline) model where the residual is δd_{EL} . Minor deviations from diagonal symmetry in the Figure are the result of the source spacing being coarser than the receiver spacing.

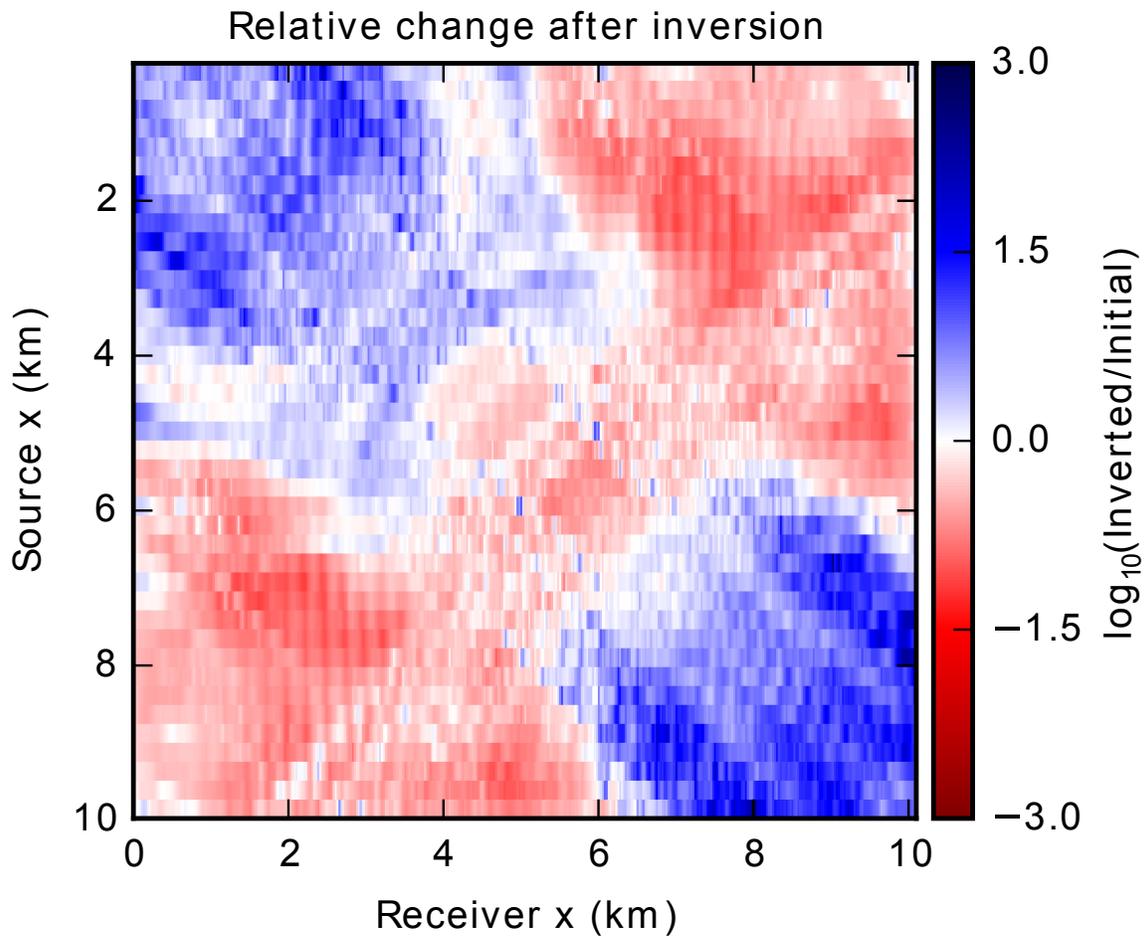


Figure 5-13: Figure 5-12 showed the EL DDFWI objective value contributions of each source-receiver combination. Here we show the relative change of the objective value contribution for each source-receiver combination at the end of the inversion, where the time-lapse estimate is displayed in Figure 5-7d.

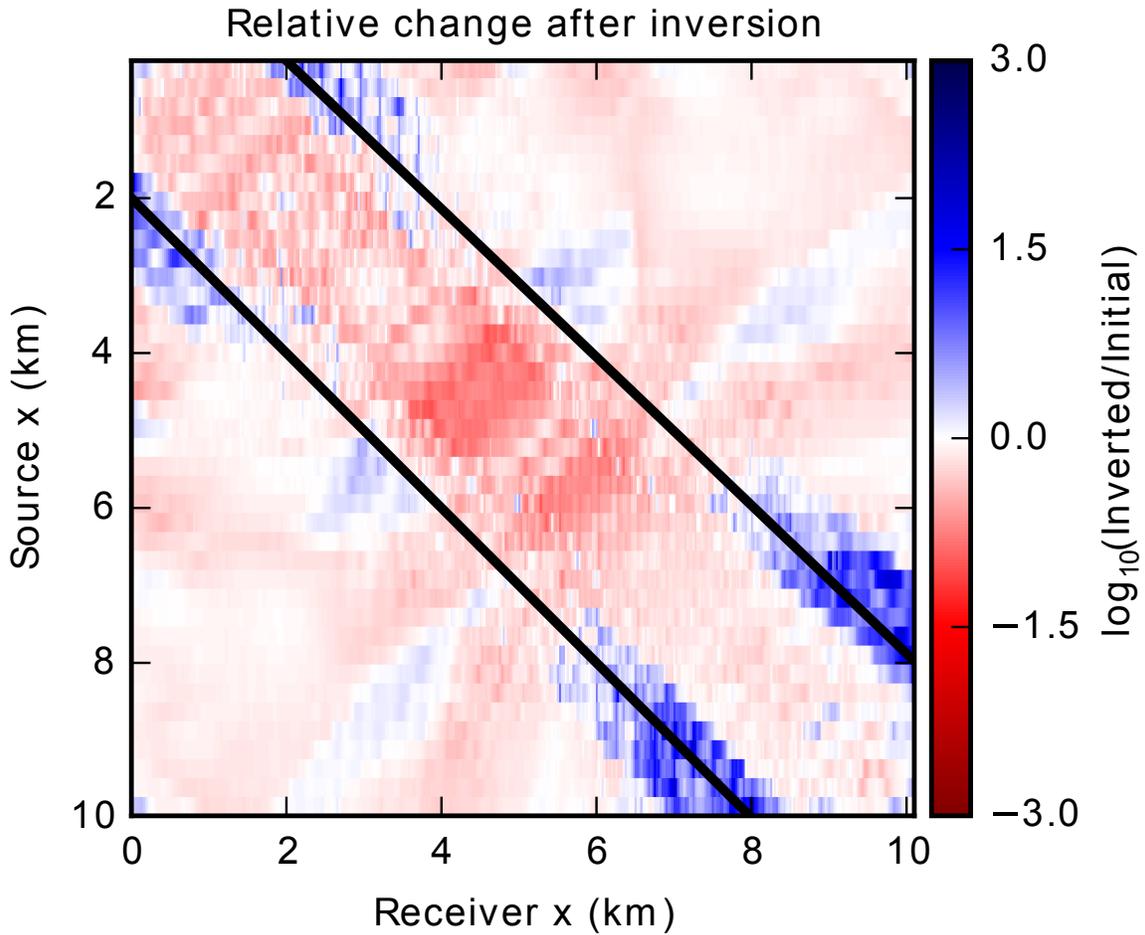


Figure 5-14: Figure 5-12 showed the EL DDFWI objective value contributions of each source-receiver combination. Here we show the relative change of the objective value contribution for each source-receiver combination at the end of the inversion when we limit the maximum source-receiver offset to 2.0 km. The source-receiver combinations used in the EL DDFWI inversion fall within the two black diagonal lines. The source-receiver combinations outside these two black lines have offset exceeding 2.0 km. They are plotted here to inspect how their objective function contribution changes, even though they are not used in the inversion.

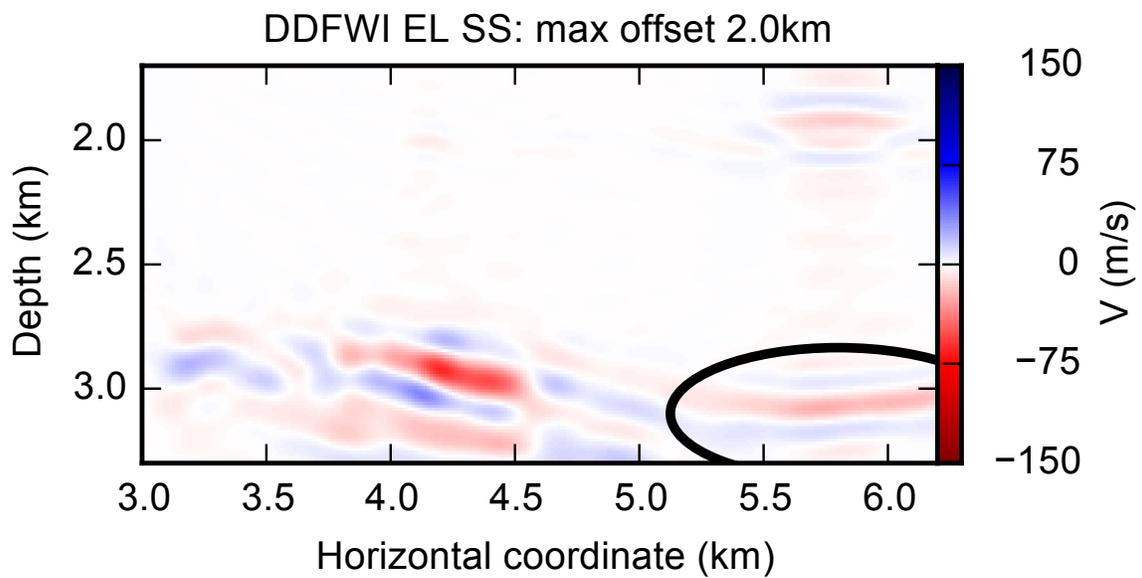


Figure 5-15: DDFWI with the maximum source-receiver offset limited to 2.0 km. The ellipse highlights an artifact that was less pronounced when all offsets were used in Figure 5-7d.

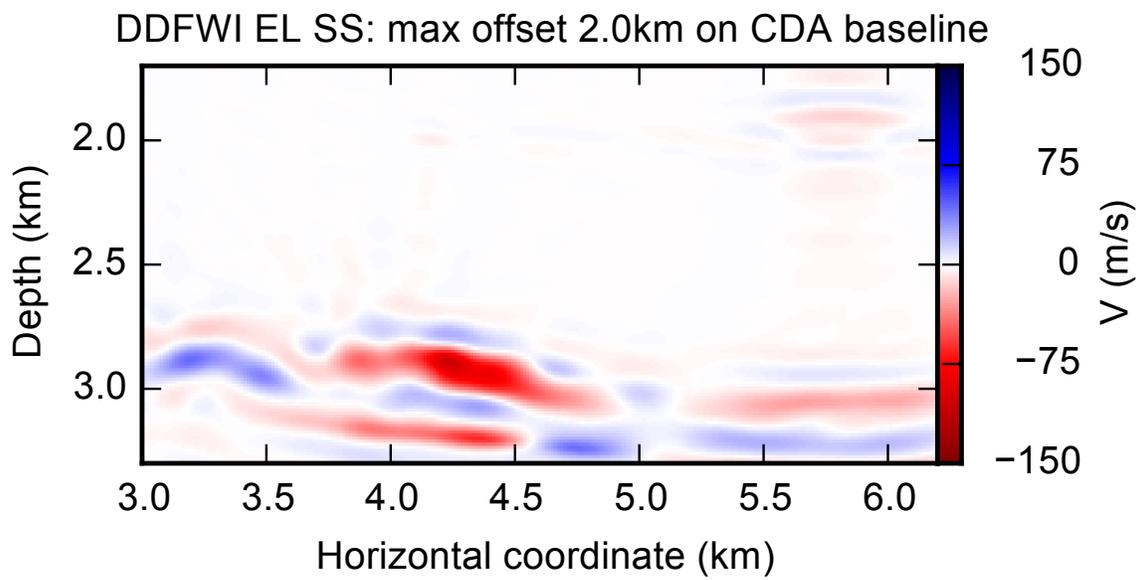


Figure 5-16: DDFWI with the maximum source-receiver offset limited to 2.0 km starting on the CDA baseline velocity model of Figure 5-4c.

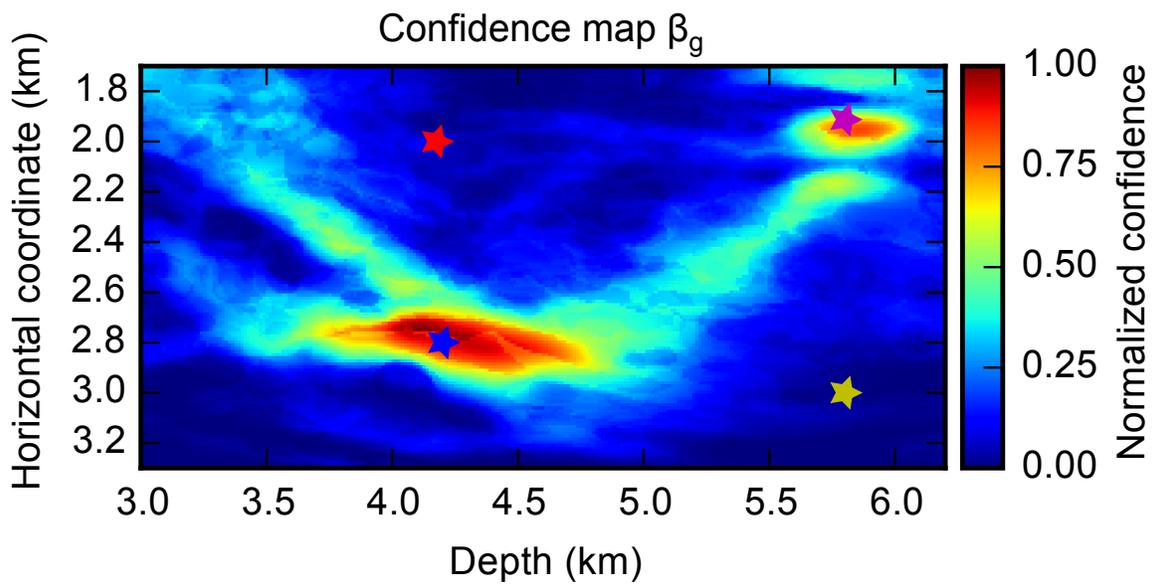


Figure 5-17: Zoomed in section of the gradient based confidence map. The four stars correspond to four points at which we plot the illumination compensated gradient evolution in Figure 5-18. The blue star ($\beta_g = 0.92$) corresponds to the reservoir time-lapse area. The red ($\beta_g = 0.08$) and magenta ($\beta_g = 0.70$) stars are in the left and right elliptical overburden perturbations respectively. The yellow star ($\beta_g = 0.01$) is in the region of Figures 5-15 and 5-16 where an artificial time-lapse reflector is seen.

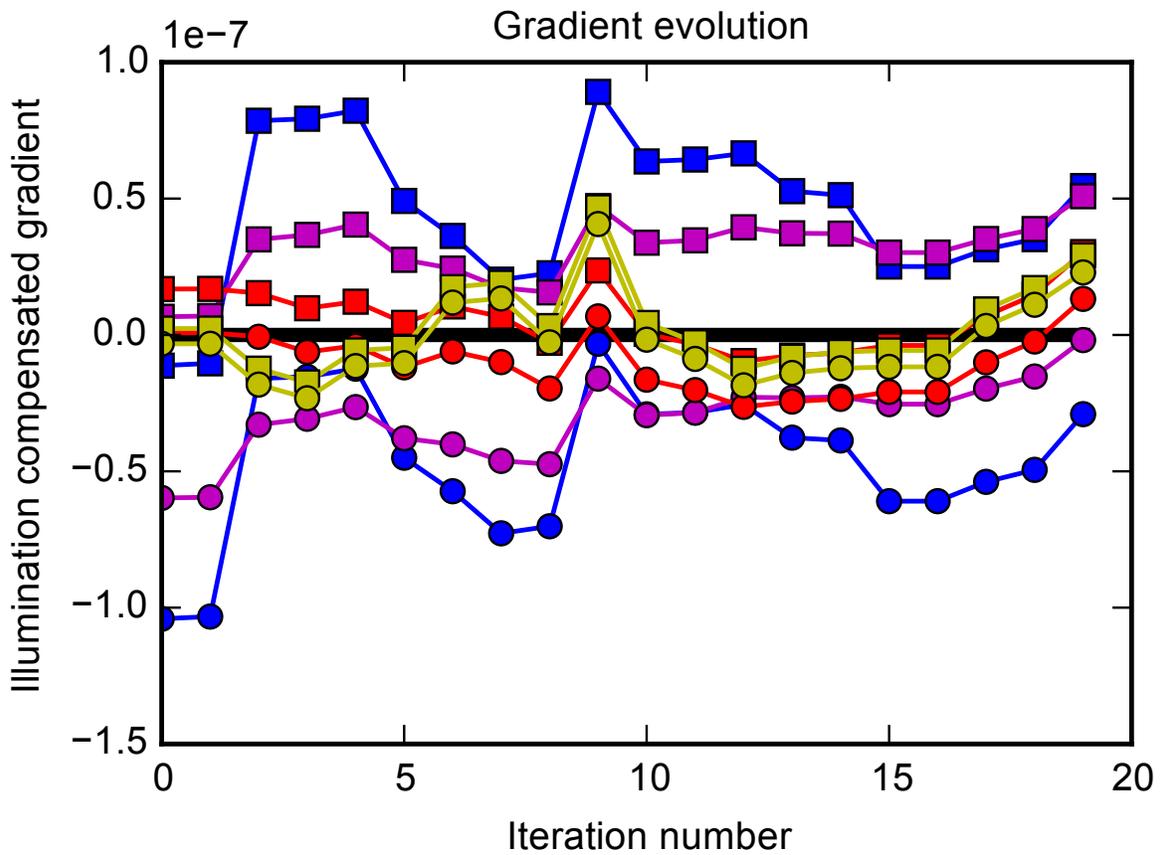


Figure 5-18: The squares and circles represent the illumination compensated baseline and monitor gradients respectively at four pixels of interest. These pixels are indicated by stars of matching color in Figure 5-17.

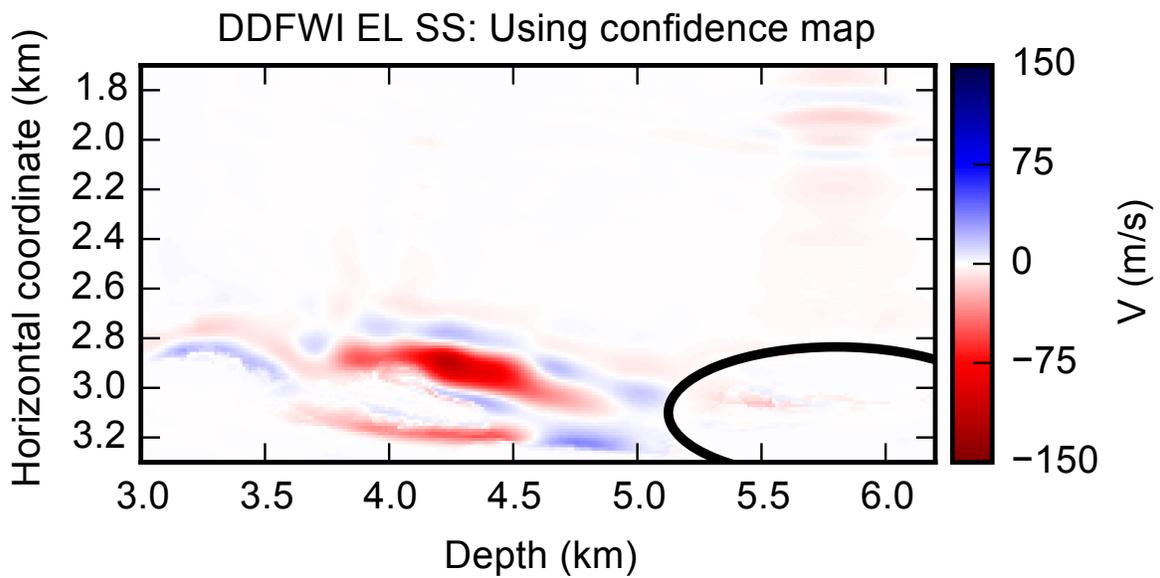


Figure 5-19: Same setting as Figure 5-16. DDFWI with the maximum source-receiver offset limited to 2.0 km starting on the CDA baseline velocity model of Figure 5-4c. We use the confidence map from 5-17 to regularize the model difference. The ellipse corresponds to the incorrect event which was visible in Figures 5-15 and 5-16, but is attenuated significantly.

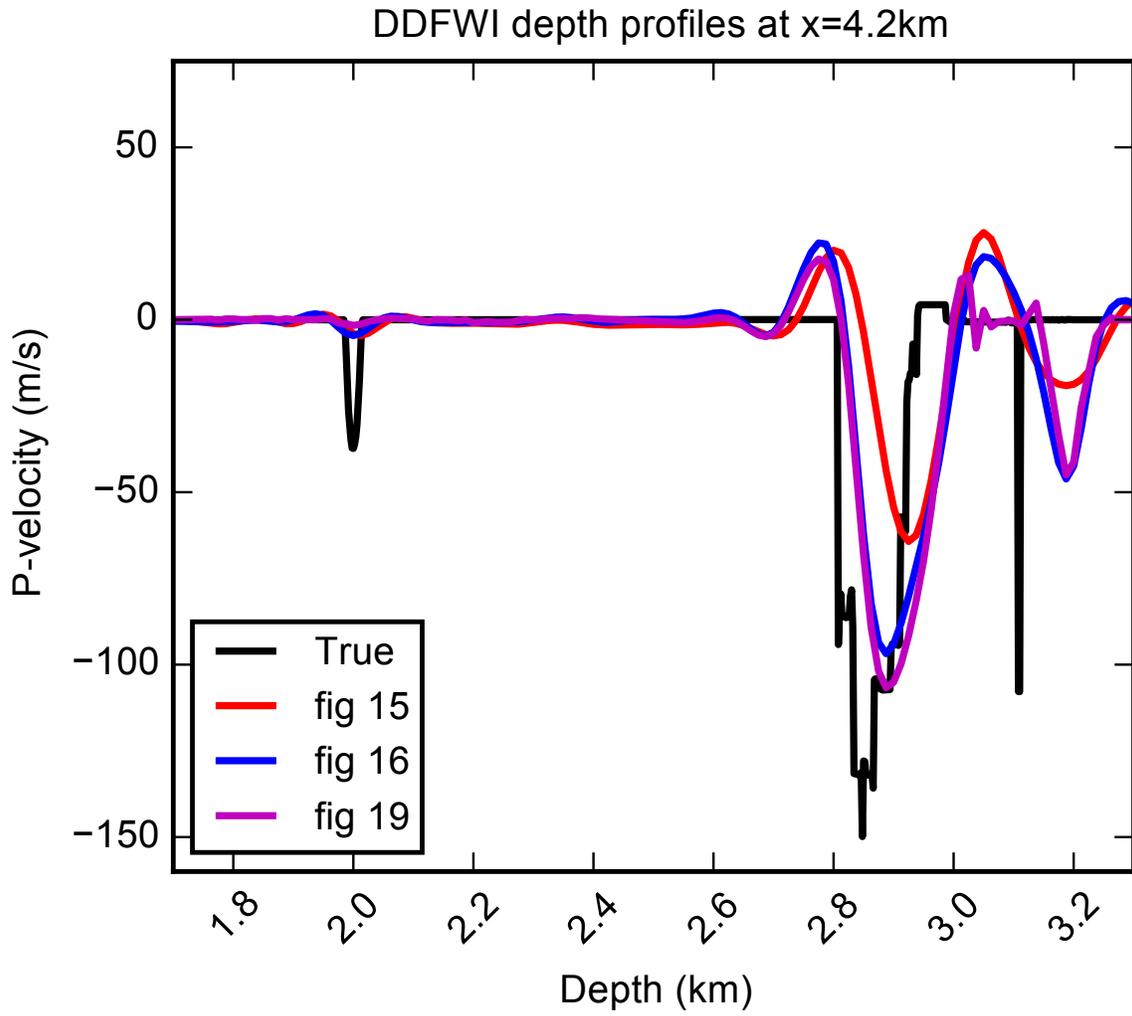


Figure 5-20: Comparing Figures 5-15,5-16 and 5-19 along a depth profile.

Chapter 6

An efficient coupled acoustic-elastic local solver applied to phase inversion

Abstract

In characterizing reservoirs, we are often interested in detailed elastic parameters about only a very limited part of the subsurface. To that end, we introduce a local solver which uses an acoustic solver to propagate the wavefield to a sub-domain on which we use a local elastic solver. This avoids the use of an expensive full domain elastic solver while still incorporating elastic physics in the region where it is most important. We apply the local solver to modeling phase variation with angle. We include an inversion example to demonstrate that our recovered phase from the hybrid solver is sufficiently accurate to recover important subsurface reservoir properties.

6.1 Introduction

When seismic waves reflect from interfaces in the Earth, their reflected amplitude depends on the incidence angle of the incoming wave. A physical model for describing the reflection coefficient in terms of the incidence angle is the Zoeppritz (1919) equations. The Zoeppritz equations and their various approximations (e.g. Aki and Richards (2002)) assume an incident plane-wave. As a result, the derived Plane-wave

This chapter is based on Willemsen and Malcolm (2016) and currently submitted paper (??)

Reflection Coefficients (PRCs) are only accurate for plane waves. In most physical experiments however, waves radiate outwards from point-sources in (approximately) spherical shells. O'Brien (1963) noted that plane wave theory poorly describes the reflection coefficient of spherical waves close to the critical incidence angle. This critical incidence angle is defined as the incidence angle for which Snell's law prescribes that the refracted wave moves parallel to the interface (i.e. it is a head wave). Červený (1959) shows that the Spherical-wave Reflection Coefficients (SRCs) are related to PRCs. This intuitively makes sense, since Fourier analysis shows that a spherical wavefront can be decomposed into wavenumber vectors, each representing plane-waves with different incidence angles. The SRC as presented by Červený (1959) is therefore a weighted integral of PRCs over a range of incidence angles. The specifics of how the reflection amplitude varies with incidence angle is the subject of study in AVA (Amplitude Variation with Angle) and AVO (Amplitude Variation with Offset). The P-velocity, density and S-velocity of the material on both sides of the interface together determine the particular shape of the reflection coefficient curve when plotted against angle (AVA) or offset (AVO). Since oil and gas influence these seismic material properties, studying AVA/AVO is an important tool for seismic interpreters.

In addition to AVA, seismic waves also experience Phase Variation with Angle (PVA). Waves reflecting on the interface at larger (i.e. shallower) incidence angles than the critical incidence angle are called post-critical reflections. The PRCs of these post-critical reflections are complex numbers (see for instance Downton and Ursenbach (2006) for a recent discussion). This complex reflection coefficient results in a phase-shift of the reflected wave. In contrast, PRCs for incidence angles smaller than the critical angle have zero phase-shift. It is important to emphasize that in this chapter we use the term phase-shift exclusively for shifts that change the shape of the wavelet. Time-shifts are technically also phase-shifts, but these are not the focus of PVA. Similar to AVA, the characteristics of this PVA curve depend on the contrast in elastic properties along the interface. This makes PVA a useful quantity for inversion either by itself or in addition to AVA. It is worth pointing out that AVA and PVA are observed in multiple types of reflected waves. These effects are not limited to PP

or SS reflections, but also occur in converted waves like PS and SP reflections. For instance Gong and McMechan (2016) use both AVA and PVA for the combination of PP and PS reflections to estimate material properties at an interface. However in this chapter we focus on PP reflections only, since they are present in both land and marine surveys and typically have the best signal to noise ratio. Recall that the SRC for a specific incidence angle contains PRCs of all incidence angles through a weighted integration. This means that the SRC for a pre-critical incidence angle already includes a phase shift (Červený and Hron, 1961), unlike the PRC for the same pre-critical incidence angle. It is worth pointing out that the PP reflection only experiences PVA when the reflector has a critical angle. If the P-velocity contrast is insufficient to generate PVA in the PP reflection, then the SP reflection may be used as alternative since a critical angle will almost always be present (Zhu and McMechan, 2012)

Observations made by Zhu and McMechan (2012) suggest that the phase of the PP reflection’s wavelet is stable during propagation through a heterogeneous overburden. This is contrasted by the amplitude of the reflection, which reduces at each interface in the overburden due to transmission losses caused by energy partitioning. The relative stability of the phase of the wavelet during propagation makes PVA a useful quantity to determine material properties at an interface, either by itself or as extra constraint in addition to AVA. In order to invert for the material properties at a target reflector we need to setup an inverse problem. In this inverse problem we try to minimize the discrepancy between the simulated reflection response and the observed one. Several forward models have been studied for generating these simulated reflection responses. In this study we focus on the modeling of and inverting for the phase of the reflection.

AVA for pre-critical reflections has traditionally received a lot of attention in the literature, see Buland and Omre (2003) and Smith and Gidlow (1987) for example. The analysis by O’Brien (1963) suggests that this modeling will break for more realistic spherical waves in the neighborhood of the critical angle. There are only a few studies making use of the AVA/PVA response of large-angle reflections for determining material properties at a reflector of interest. One of the issues for such

studies is the selection of an appropriate forward model. Arguably the most accurate forward model is a wavefield simulation on an Earth model containing the reflectors of interest. The amplitude and phase of the modeled reflections can then be compared directly with the observed reflections in the data. In the very least such a simulation requires P-velocity, density and S-velocity, to include the parameters to which reflection coefficients are sensitive. The simplest type of wavefield simulation for reflection modeling is therefore elastic, but these simulations are computationally expensive. While acoustic simulations are significantly cheaper, they cannot substitute for elastic solvers since phase-shifts of PP reflections are influenced significantly by shear velocity contrast, as we demonstrate in this chapter. Some studies have used alternative forward models for the inversion of large-angle reflections. For example, Zhu et al. (2012b); Zhu and McMechan (2014) use the $\tau - p$ transform to decompose the recorded data into plane waves. A ray tracer is then used to index these plane waves for multiple reflection angles at reflector locations of interest. From these indexed plane-waves the AVA and PVA of the corresponding PRC is determined. These PRCs at the reflection points are then compared with PRCs from Zoeppritz modeling to invert for the material contrast. Since it used a ray-based method, this approach does not model finite-frequency effects of the wavefield as it propagates through the overburden. The method also requires the explicit picking of reflectors and their dip angles from migrated images. Instead of using a ray-tracer to measure PVA inversion, Zhu et al. (2012a) use a Finite Difference (FD) wavefield simulator to generate Angle Domain Common Image Gathers (ADCIG) (Sava et al., 2001). (A good overview of how these ADCIGs are computed and their limitations is given by Jin et al. (2014)). Zhu et al. (2012a) show that the phase of the imaged reflector varies as function of reflection incidence angle in these ADCIGs. In theory one would expect that these observations could subsequently be used to invert for material parameters, although to our knowledge this has not been done. As with any imaging method, such an approach is sensitive to defocussing as a result of inaccuracies in the velocity model used for migration. Both the imperfections in constructing ADCIGs and defocusing due to inaccurate velocity models are expected to lead to artifacts when used in an

inversion, which provides the impetus to the development of full-waveform methods that directly fit the phase of the data themselves, rather than first computing the ADCIGs.

Both this proposed approach of directly fitting the phase of the waveform and the ADCIG approach of Zhu et al. (2012a) require a FD wavefield simulator. It is this FD simulator which results in the largest expense of these methods. There is a lot of fundamental research to improve their efficiency, see for instance the work of Wang et al. (2011) and Zepeda-Núñez and Demanet (2016). For the purpose of modeling reflections there are additional ways to reduce the cost of the wavefield simulation. Hobro et al. (2014) and Chapman et al. (2014) for instance show that elastic PP reflections can be approximated using an acoustic FD solver. They do so by adding additional source terms at layer interfaces. This allows them to match the reflection response of the elastic simulation accurately up to the critical angle. The cost of this procedure is only twice that of a regular acoustic simulation, which is significantly less than the cost of a single elastic wavefield simulation. But the inability of this method to accurately model post-critical reflections and their associated phase-shifts makes it less applicable to PVA inversion. An alternative way of bringing down the cost of the wavefield simulation of the reflection is by using a local solver centered around the reflector of interest.

While elastic FD simulations accurately model the reflection response for all incidence angles, their high computational cost has been a limiting factor to their applicability to AVA/PVA. Local solvers reduce this computational barrier by limiting the size of the computational domain to a smaller region of interest. This usually requires a pre-computation step in which several full domain simulations are performed. After these precomputations the local wavefields can be generated at much faster speed than full domain simulations. These localized wavefields can then for instance be used for inversions for salt boundary (Willemsen et al., 2016b), time-lapse (Borisov et al., 2015; Robertsson et al., 2000) (**cite our accepted TLE paper here when published**) and reflector properties from PVA as we do in this chapter.

Broadly speaking, local solvers can be subdivided into exact local solvers (van

Manen et al., 2007; Willemsen et al., 2016b) and inexact local solvers. By relaxing accuracy constraints, the inexact local solvers attain higher computational efficiency. The inexact (potentially anisotropic) elastic local solver formulation of for instance Robertsson and Chapman (2000) allows the model to be perturbed within the local domain. Computational gains are achieved by discarding waves scattering from those perturbations that exit the local domain, scatter in the inhomogeneous background model and subsequently reenter the local domain. Primary PP reflections used in the AVA/PVA analyses are waves that by their very nature reflect only once. They therefore do not reenter a local domain centered around a reflector and are accurately modeled by this local solver, and thus modeling primary reflections with this type of inexact local solver will not result in significant accuracy loss. This makes it possible to change the elastic material properties around the reflector and still accurately model the reflection response we are interested in for inversion. The elastic local solver of Robertsson and Chapman (2000) requires many elastic full domain Green’s function computations to connect the local domain and the acquisition. In this chapter we try to reduce the computational cost of this step by making a physical simplification. This simplification is motivated by the objective of this chapter to accurately model the PVA of PP reflections with the aim of using it in an inversion. Recall the observation of Zhu and McMechan (2012) that the phase of PP reflections is approximately unchanged by propagation through an elastic overburden. This raises the idea of using acoustic Green’s functions instead for propagating the PP reflection through the overburden. Both Elastic and Acoustic propagation are expected to leave the phase approximately unmodified. Since elastic simulations are significantly more expensive than acoustic simulations (see for instance Hobro et al. (2014)) this approach has the potential to reduce the precomputation cost of the local solver while still accurately modeling the phase of the reflected wave. The acoustic approximation is not always good as Cance and Capdeville (2015) demonstrate. When the elastic medium contains significant heterogeneity with typical length-scale much smaller than the dominant wavelength effective anisotropy is induced. Using the same heterogeneous P-velocity and density in an acoustic simulation does not reproduce this anisotropy.

This difference in effective anisotropy experienced in acoustic and elastic simulations results in predominantly arrival time differences. Figures 4b and 5b in the work of Cance and Capdeville (2015) demonstrate that even with highly heterogeneous elastic models, acoustic propagation gives wavelets of similar shape although they have minor arrival time differences. We therefore expect that for the purpose of propagating the local elastic reflections through the overburden, acoustic propagation is sufficient for preserving the shape of the wavelet we invert for. The local solver we introduce in this chapter therefore mixes physics by using an acoustic solver to propagate the wavefield through the overburden to a local elastic solver around the reflector, where the elastic behavior determines the phase shift of interest. While focusing on PVA inversion in this chapter, potentially combining this with AVA better constrains the inverse problem for the elastic material contrast. Zhu and McMechan (2014) show in their example that even when not accurately modeling overburden amplitude losses, the addition of amplitude information helps better constrain P-velocity and density.

There are a few other seismic studies in literature where different solvers are combined. Monteiller et al. (2013) for instance use an efficient spherically symmetric wave solver to propagate teleseismic body waves through the Earth. Only in the relatively small region of interest they are coupled with a 3D heterogeneous solver. Capdeville et al. (2003) use a similar approach, where the Earth is again separated into two parts. Deeper parts of the Earth are assumed to be spherically symmetric, making efficient solutions possible. Only the top shell has 3D heterogeneity. This approach again limits the expensive solver to a small region, using a simplified solver for the exterior. Finally Yu et al. (2016) derive a coupled acoustic-elastic set of equations to aid elastic reverse time migration using ocean bottom geophones.

6.2 Methods

In this chapter, we adapt the local solver of Robertsson and Chapman (2000) to reduce the computational cost of the phase modelling necessary for PVA inversion. Specifically, we substitute computationally efficient Constant Density Acoustic (CDA)

Green’s functions for Elastic (EL) Green’s functions to couple the local solver to the acquisition. Observations by Zhu and McMechan (2012) suggest that phase of the seismic wave is relatively insensitive to the material through which it propagates, motivating this computationally efficient substitution.

6.2.1 Local solver

The local solver used in this study is a modification of the local solver introduced by Robertsson and Chapman (2000). This local solver generates an approximation of the scattered field due to a model perturbation within the local domain. The only component of the wavefield discarded by the local solver is the wavefield scattering on the perturbation, leaving the local domain, and then reentering before propagating to the receivers (i.e. long-range interactions). Such a discarded event is displayed in gray in Figure 6-1.

As mentioned above, if we use an efficient scalar CDA solver instead of an elastic solver to propagate the wavefield through the overburden, the observed phase is expected to be similar. The elastic material parameters are most important at the reflection point, where their contrasts determine the phase shift of the postcritical reflection. We thus use the elastic local solver around the reflector of interest and propagate the PP reflection to the acquisition using efficiently generated CDA Green’s functions.

6.2.2 Standard elastic local solver

Before discussing the CDA modifications we first introduce the framework in which Robertsson and Chapman (2000) introduce the elastic local solver. In their approach both the full domain Green’s functions and the local wavefields are computed using the same elastic staggered grid implementation. In this study we use the algorithm introduced by Levander (1988) for elastic wavefield simulations. The equation of

motion in 2D is

$$\rho \frac{\partial v_x(\mathbf{x}, t)}{\partial t} = \frac{\partial \tau_{xx}(\mathbf{x}, t)}{\partial x} + \frac{\partial \tau_{xz}(\mathbf{x}, t)}{\partial z} \quad (6.1)$$

$$\rho \frac{\partial v_z(\mathbf{x}, t)}{\partial t} = \frac{\partial \tau_{zx}(\mathbf{x}, t)}{\partial x} + \frac{\partial \tau_{zz}(\mathbf{x}, t)}{\partial z} \quad (6.2)$$

and the constitutive equations for an isotropic elastic medium are

$$\frac{\partial \tau_{xx}(\mathbf{x}, t)}{\partial t} = (\lambda(\mathbf{x}) + 2\mu(\mathbf{x})) \frac{v_x(\mathbf{x}, t)}{\partial x} + \lambda(\mathbf{x}) \frac{v_z(\mathbf{x}, t)}{\partial z} \quad (6.3)$$

$$\frac{\partial \tau_{zz}(\mathbf{x}, t)}{\partial t} = (\lambda(\mathbf{x}) + 2\mu(\mathbf{x})) \frac{v_z(\mathbf{x}, t)}{\partial z} + \lambda(\mathbf{x}) \frac{v_x(\mathbf{x}, t)}{\partial x} \quad (6.4)$$

$$\frac{\partial \tau_{xz}(\mathbf{x}, t)}{\partial t} = \mu(\mathbf{x}) \left(\frac{v_x(\mathbf{x}, t)}{\partial z} + \frac{v_z(\mathbf{x}, t)}{\partial x} \right) \quad (6.5)$$

A bold font is used to represent vector quantities. Specifically the 2D position vector $\mathbf{x} = (x, z)$. Because of symmetry of the stress tensor $\tau_{xz} = \tau_{zx}$. The elastic solver updates the velocities and the stresses in a turnwise manner. First the velocities in (6.1)-(6.2) are updated. These are then used to update the stresses in (6.3)-(6.5). Equations 6.1-6.5 are solved on a staggered grid with second order accuracy in time and fourth order accuracy in space.

We use the framework introduced by Robertsson and Chapman (2000) to create a local solver for an isotropic elastic medium, but it can be extended to anisotropy and attenuation as well. Before introducing the local solver, we first define the following terminology

- Background wavefield: The wavefield in background model $m^0(\mathbf{x})$
- Perturbed wavefield: The wavefield in the perturbed model $m(\mathbf{x})$
- Scattered wavefield: The difference between the perturbed and the background wavefield, caused by model update $\Delta m(\mathbf{x})$ limited to the interior of the local solver domain

For the z -component of the velocity for example, the notation we use is respectively $v_z^0(\mathbf{x}, t)$, $v_z(\mathbf{x}, t)$ and $v_z^{sc}(\mathbf{x}, t)$.

Figure 6-2 shows a section of a staggered elastic grid zoomed in on part of the top-right boundary of the local solver. The four different symbols represent the locations where the elastic wavefield quantities from (6.1)-(6.5) are evaluated. On this staggered grid these wavefield components are evaluated at different physical locations. The black grid lines create grid cells, each including the four staggered grid symbols. The legend in Figure 6-2 makes clear which elastic wavefield component is computed at which symbol. The local solver computes the perturbed wavefield within the interior of the truncated domain. The transition between the interior and the exterior is represented by a magenta line. This magenta line intersects grid cells on the boundary by design. The model perturbations $\Delta m(\mathbf{x})$ are restricted to the inside of the inner red line, inside truncated domain. At the shaded symbols outside of the truncated domain the scattered wavefield quantities are computed instead of the perturbed wavefield quantities. These scattered wavefields become nonzero only when $\Delta m(\mathbf{x})$ is nonzero. The local solver of Robertsson and Chapman (2000) requires storage of the background wavefields at all locations between the two red lines, as we explain soon. Finally, the two bold yellow lines represent the locations between which the acoustic Green's functions need to be stored in the algorithm introduced later in this chapter. These acoustic Green's functions only require storage of a single pressure field in each cell, unlike the 5 fields required by the elastic solver.

The spatial derivatives in (6.1)-(6.5) are evaluated with fourth order accuracy on a staggered grid, although extensions to different orders of accuracy are straightforward. As a concrete example of how this equation is discretized on the staggered grid we now give the update equation for the perturbed z-component of the velocity in (6.2). Graphically this is represented by the left-most blue stencil in Figure 6-2, which is centered around the unfilled (i.e. perturbed) v_z symbol. For now ignore that this particular update stencil involves some shaded symbols and simply assume all of the symbols are unfilled. In other words, assume for now that the stencil only involves perturbed wavefield components. We update the velocity

$$v_z(\mathbf{x}_{(i,k)}, t) = v_z(\mathbf{x}_{(i,k)}, t - \Delta t) + \Delta v_z(\mathbf{x}_{(i,k)}, t) \quad (6.6)$$

using

$$\begin{aligned} \Delta v_z(\mathbf{x}_{(i,k)}, t) = & \frac{\Delta t}{\rho} \left[\frac{c_1}{\Delta x} \left(\tau_{zx}(x_{i+1}, z_k, t - \frac{1}{2}\Delta t) - \tau_{zx}(x_i, z_k, t - \frac{1}{2}\Delta t) \right) \right. \\ & + \frac{c_2}{\Delta x} \left(\tau_{zx}(x_{i+2}, z_k, t - \frac{1}{2}\Delta t) - \tau_{zx}(x_{i-1}, z_k, t - \frac{1}{2}\Delta t) \right) \\ & + \frac{c_1}{\Delta z} \left(\tau_{zz}(x_i, z_{k+1}, t - \frac{1}{2}\Delta t) - \tau_{zz}(x_i, z_k, t - \frac{1}{2}\Delta t) \right) \\ & \left. + \frac{c_2}{\Delta z} \left(\tau_{zz}(x_i, z_{k+2}, t - \frac{1}{2}\Delta t) - \tau_{zz}(x_i, z_{k-1}, t - \frac{1}{2}\Delta t) \right) \right]. \quad (6.7) \end{aligned}$$

Notice how the staggered spatial derivatives are evaluated at the staggered time-step (i.e. $t - \frac{1}{2}\Delta t$). In equation 6.7, the fourth order accurate staggered grid coefficients c_1 and c_2 are $\frac{9}{8}$ and $-\frac{1}{24}$ respectively. The subscript on the position vector $\mathbf{x}_{(i,k)}$ represents the corresponding location of the wavefield quantity in cell number i, k . On the right hand side the vector \mathbf{x} is decomposed in its x and z coordinates in this 2D example. Extensions to 3D are trivial. An update equation completely analogous to (6.7) can be derived for the propagation of the scattered v_z^{sc} in the shaded region. This analogous update equation replaces all perturbed wavefield quantities with their scattered counterparts. Update equations for elastic wavefield quantities other than v_z use analogous derivative stencils to those shown in Figure 6-2.

Sufficiently far away from the magenta boundary in Figure 6-2 the update equations for the perturbed and scattered wavefields are trivial to compute. But within the two red lines, the derivative stencils cross the magenta line separating the locations where the perturbed and scattered wavefields are computed and stored. The update equations for nodes between the red lines therefore require a modification. We now go back to the specific case of the leftmost stencil in Figure 6-2. In our earlier discussion of equation 6.7 we made the assumption that the perturbed wavefield quantities were available at all required locations. With i, k centered at the central cell of the stencil, observe that the update stencil accesses the scattered τ_{zz}^{sc} at the shaded symbols in cells i, k and $i, k - 1$ instead of the required perturbed τ_{zz} . By adding the background field τ_{zz}^0 to the scattered τ_{zz}^{sc} at these locations, we obtain by definition the perturbed τ_{zz} and consequently the update Equation 6.7 can be evalu-

ated correctly. This background field τ_{zz}^0 is obtained by convolving the precomputed τ_{zz}^0 Green's function with the source wavelet for the particular source we simulate locally. Applying this correction procedure to all nodes between the red boundaries automatically introduces the background wavefield into the local domain in case there are no model updates $\Delta m(\mathbf{x})$.

Similarly, the second stencil in Figure 6-2 has the opposite problem when updating v_z^{sc} at the shaded symbol. Part of its stencil crosses the magenta line and accesses the perturbed field τ_{xx} stored at the unfilled symbols. At these locations we need to subtract the background wavefield from the perturbed wavefield so we can correctly evaluate the stencil for updating the scattered wavefield at the shaded symbol. In case the model is updated with $\Delta m(\mathbf{x})$ in the unshaded region this correctly computes the perturbed fields at the unfilled symbols and the scattered field at the filled symbols. Multiple scattering between the background model at the shaded nodes and the potentially perturbed model at the unfilled nodes is computed correctly. By definition, when we talk about the local domain in this chapter we mean the entire local solver grid. Therefore the local domain is the combination of the shaded and unshaded nodes, surrounded by its PML boundary condition outside the outer yellow line.

To this point, we have discussed the mechanisms for computing the elastic wavefield locally using precomputed background Green's functions to inject the wavefield into the local domain. In any seismic inversion we need to compare the simulated wavefield with the observed one. This requires propagation of the elastic wavefield from the local domain to the receiver locations, typically on the surface or in a borehole. Robertsson and Chapman (2000) use a boundary integral to propagate the elastic scattered field from the local solver to the physical receiver locations. This requires an additional set of full domain elastic Green's functions.

The elastic local solver is the combination of the shaded and unshaded regions together with a PML surrounding the ensemble. The number of nodes included in the local solver is typically a fraction of that of the full computational domain, leading to substantial savings. As illustrated in Figure 6-1, the only wavefield component that

is discarded in this approach is a scattered wave leaving the local domain, scattering on a background inhomogeneity and then reentering. In this chapter we choose the shaded region to be approximately 5 layers thick for reasons which will become clear soon.

6.2.3 CDA coupling

As discussed above, a straight forward implementation of the algorithm of Robertsson and Chapman (2000) would use expensive elastic Green’s functions to propagate the wavefield between the boundary of the local domain and the acquisition surface. In this section, we discuss how these can be replaced with CDA Green’s functions. This replacement is motivated by the observation of Zhu and McMechan (2012) that the phase of a PP-reflection is primarily affected by scattering at the reflector and less so by propagation through the (heterogeneous) elastic overburden. We expect that this observation remains valid when restricting ourselves to heterogeneous overburdens without density and shear velocity contrasts (i.e. a CDA simulation). We therefore define the background model as CDA and use the associated background CDA Green’s functions to propagate the wavefield through the overburden. The scalar CDA solver from the numerical code PySIT (Hewett et al., 2013) is used to compute these Green’s functions. Concretely, the CDA background Green’s functions are obtained by defining the background model $m_0(\mathbf{x})$ as exactly the background P-velocity on the full computational domain

$$m^0(\mathbf{x}) = \{V_{p,0}(\mathbf{x})\}. \quad (6.8)$$

We then solve the equation

$$\frac{1}{V_{p,0}(\mathbf{x})^2} \frac{\partial^2 p(\mathbf{x}, t)}{\partial t^2} - \nabla^2 p(\mathbf{x}, t) = f(\mathbf{x}, t) \quad (6.9)$$

with the source defined by Dirac delta functions for the Green's function computations.

$$f(\mathbf{x}, t) = \delta(\mathbf{x} - \mathbf{x}_0)\delta(t - t_0). \quad (6.10)$$

As is typical for the scalar wave equation, PySIT solves (6.9) on a regular Finite Difference (FD) grid.

The elastic local solver is second order accurate in time and fourth order accurate in space. We match these orders of accuracy in PySIT when solving for the CDA background Green's functions. The algorithm we introduce in this chapter can be expanded to higher orders of accuracy without difficulty.

CDA propagation on the background model $m_0(\mathbf{x})$ physically corresponds to propagation through a constant density liquid (i.e. no shear velocity). Elastic model updates $\Delta m(\mathbf{x})$ to this liquid background state should be restricted to the interior of the truncated domain. The result is that the material properties at the boundary of the elastic local solver remain CDA, ensuring physical consistency with the injected CDA wavefield. But inside the local domain, away from the boundaries, the perturbed model $m(\mathbf{x})$ is elastic:

$$m(\mathbf{x}) = \{V_p(\mathbf{x}), \rho(\mathbf{x}), V_s(\mathbf{x})\}. \quad (6.11)$$

Figure 6-3 schematically shows the combination of the CDA full domain with the EL local solver. The pink layers represent Perfectly Matched Layer (PML) boundary conditions, although free surface conditions are also possible on the full domain. The PML around the elastic local solver is in this chapter much thinner than the one surrounding the full computational domain. The red boundary corresponds with the injection region between the red lines in Figure 6-2. Within the blue boundary of Figure 6-3 the model update is the elastic region we want to model, for the purposes of this chapter it contains the reflector whose PP reflection we want to model. Between the blue and red boundaries the material gradually transitions from a fully elastic medium to the CDA state of the exterior over a taper with size proportional to the dominant wavelength. The star and triangle symbols represent a source and receiver respectively. The lines connecting these to the red boundary represent background

CDA Green's functions. With the geometry defined, we are now ready to discuss how the CDA field is injected on the staggered EL grid of the local solver.

Injecting the CDA field in the EL local solver

The regular grid nodes where the pressure fields are computed in the CDA solver spatially coincide with the normal stress location (i.e. squares) of the elastic grid in Figure 6-2. The elastic local solver requires the background elastic wavefield computed on the symbols inbetween the two red lines. One approach to obtain the required injection velocities from the pressures is to compute them on the fly at the required locations when solving (6.9) for the CDA background pressure Green's functions. Even though in a CDA simulation we only update pressures at the regular nodes, we can still evaluate (6.1)-(6.2) for the velocities at the staggered grid locations between the red lines in Figure 6-2 by using the following CDA properties

$$\tau_{xx} = \tau_{zz} = p \tag{6.12}$$

$$\tau_{xz} = 0. \tag{6.13}$$

This is justified because at the boundary where we compute the velocities, the model is still the CDA background model. Recall that the (in this case elastic) model update $\Delta m(\mathbf{x})$ is limited to the interior of the inner red line in Figure 6-2. The velocities we compute this way can be stored during the precomputation step. They are then available for injection in the local solver subsequently. With CDA material properties at the red boundary in Figure 6-3 and fully elastic material properties within the blue box in Figure 6-3, an undesirably large contrast in ρ and V_s can occur across the CDA-EL taper zone. In the next section we therefore advocate a more flexible approximate approach which allows injection of the pressure wavefield on a material with spatially variable $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$.

6.2.4 New approximate injection approach for variable ρ/V_s along the boundary of the local domain

The approach used above to couple the acoustic and elastic solvers is intuitive and works well when we taper the local elastic model to constant density and zero S-velocity. But it is important to realize that the properties (6.12) and (6.13) do not hold for a P-wave propagating through an elastic medium. The normal stresses τ_{xx} and τ_{zz} are in general not the same and the shear stress is nonzero. These effects must be accounted for when injecting the P-wave on an elastic medium with potentially spatially variable $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$. We therefore propose a new approximate approach.

Instead of pre-computing and storing the velocities at the boundary as in the previous approach, we store the CDA Green's functions at several extra layers. Using these extra layers of stored Green's functions we can evaluate the spatial derivatives in (6.1)-(6.2) to obtain the injection velocities offline. Then with these layers of velocities we can evaluate the spatial derivatives of (6.3)-(6.5) to get the associated stress field offline.

The main advantage of this new approach is flexibility. When the material at the injection boundary in the local solver is CDA, we get the same injection stresses and velocities as the approach in the previous section. But this new method can also be used to determine the stresses and particle velocities needed to approximately inject the precomputed CDA wavefield on an arbitrary elastic model of choice outside the local domain. No CDA Green's functions need to be recomputed when injecting on a model with a different choice of $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$.

Injecting on a medium with varying $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$ along its boundary makes it possible to reduce the contrast in ρ and V_s . For instance, by no longer tapering to $V_s = 0$ across the taper zone in Figure 6-3, the contrast across the taper is reduced and a more accurate simulation of the elastic PP reflection is expected. Alternatively we can also reduce the size of the taper zone for higher computational gains. Because the background Green's functions were computed on a CDA model, we can only approximately compensate the amplitude of the injected wavefield for density and S-

velocity at the injection points. The injection approach we now introduce is heuristic when the boundary of the local solver is not CDA. Specifically, the steps of the approach are:

- In a precomputation step, store the CDA pressure Green's functions on the 9 layers between the two yellow lines in Figure 6-2.
- First compute both velocity components by evaluating (6.1) and (6.2), using the density $\rho(\mathbf{x})$ of the elastic model and (6.12) and (6.13). Since (6.1) and (6.2) require the evaluation of derivatives, we cannot compute the velocities at all the 9 layers. The reason is that at some locations we do not have access to the full derivative stencil. The velocity components will therefore be available at fewer than the 9 layers where the CDA pressure is available.
- Using the velocity components computed in the previous step and the elastic material properties around the injection boundaries, evaluate (6.3) - (6.5). This will naturally result in $\tau_{xx} \neq \tau_{zz}$ and $\tau_{xz} \neq 0$ when $V_s \neq 0$. Just like for the velocity components, equations (6.3) - (6.5) require evaluation of derivative stencils. The shear stresses can be computed exactly at the three layers between the red boundaries in Figure 6-2.

With this scheme we turn the 9 layers of precomputed CDA background pressures into the velocity and stress tensor at all the required injection locations for the elastic local solver. Even though the CDA Green's function is stored along 9 layers, it requires less storage than the full elastic injection wavefield on the three layers between the red lines. The latter would require 5 quantities for each cell, resulting in a storage cost proportional to 15 layers of the pressure wavefield. In addition, the stored elastic wavefield may not be consistent if the local model changes along the injection boundary (i.e. trying a different shear velocity taper). Instead the procedure we introduce is flexible in that the elastic injection fields can easily be recomputed if the elastic properties along the boundary of the local solver are modified.

The background Green's functions are computed on a medium with no density or shear velocity variation. Even smooth variation in S-velocity and density influences

the amplitude of a P-wave. When injecting a CDA wavefield on a material with varying S-velocity and density we therefore want to 'correct' its amplitude by using compensation factors. This brings the amplitude of the injected wavefield closer to the pressure we would have simulated had we used an elastic background simulation. Appendix D derives the amplitude compensation factors for this approximate algorithm. The factors $l_\rho^{inj}(\mathbf{x})$ and $l_{V_s}^{inj}(\mathbf{x})$ compensate for smooth variation in density and S-velocity respectively. They only account for the instantaneous effect of the density and shear velocity at the injection point. They do not compensate for transmission losses which are observed if an elastic solver is used for the background Green's function computations. We apply term $l_\rho^{inj}(\mathbf{x})$ to the 9 layers of CDA Green's functions before starting the injection procedure. The scaling $l_{V_s}^{inj}(\mathbf{x})$ is applied automatically during the injection procedure through (D.1), as is seen by comparing the amplitude when V_s is zero (CDA) and when it is nonzero.

6.2.5 Exact boundary integral

The local solver computes a wavefield within the local domain, corresponding to the perturbed model $m(\mathbf{x})$. In general, for an inversion the wavefield is also required at the receiver locations where it is compared with observed data. We use Green's functions between the boundary of the local solver and the receivers, shown with gray lines on Figure 6-3, to propagate the scattered wavefield to the receiver locations. This is done by evaluating a boundary integral. We first address the scenario where the model update $\Delta m(\mathbf{x})$ at the boundary is zero, preserving the CDA properties there. In this scenario the boundary integral along the red box in figure 6-3 also assumes the standard CDA form

$$p^{sc}(\mathbf{y}, t) = \int_S \left[G_0(\mathbf{x}, \mathbf{y}, t) * \frac{\partial p^{sc}(\mathbf{x}, t)}{\partial \mathbf{x}_n} - p^{sc}(\mathbf{x}, t) * \frac{G_0(\mathbf{x}, \mathbf{y}, t)}{\partial \mathbf{x}_n} \right] dS_x, \quad (6.14)$$

where $*$ is convolution and the surface integral is over coordinate \mathbf{x} . $G_0(\mathbf{x}, \mathbf{y}, t)$ is the background CDA Green's function between boundary position \mathbf{x} and evaluation position \mathbf{y} by using (6.10) in (6.9) with $\mathbf{x}_0 = \mathbf{y}$. We denote receiver locations with

indexed \mathbf{y} throughout. The boundary integral in (6.14) accurately propagates the scattered field as long as the elastic model update $\Delta m(\mathbf{x})$ is restricted to the interior of the integration boundary. In this chapter we choose to evaluate the normal derivatives in (6.14) using a fourth order discretization to match the discretization order of the finite difference stencils. The integral boundary goes through the shaded area in Figure 6-2 and all around the local domain where the local solver computes the scattered wavefield to be evaluated in (6.14). The particular cells through which the boundary integral goes is determined by how we discretize the normal derivative. To implement a fourth order inward derivative stencil, the boundary integral is evaluated one cell interior to the outer yellow line. In this chapter, we use a central derivative stencil in which we center the boundary integral 3 cells interior to the outer yellow line. It evaluates the Green's function and normal derivative at the square symbols on two grid cells on either side.

6.2.6 Approximate boundary integral for variable ρ/V_s along the boundary of the local domain

As mentioned when introducing the injection routine, we are interested in relaxing the constraint that the perturbed model should taper to CDA propagation along the injection boundary. Similarly we still want to evaluate the pressure boundary integral of (6.14), even though the requirement of restricting $\Delta m(\mathbf{x})$ to the interior of the integral is no longer honored and the wavefield is now elastic. When injecting the pressure wavefield we modified the amplitude of the CDA pressure Green's function to closer mimic how the amplitude would have reacted in an elastic solver due to the difference in density and S-velocity between the source pixel and the evaluation pixel. This was done through the scaling factors $l_{V_s}^{inj}(\mathbf{x})$ and $l_{\rho}^{inj}(\mathbf{x})$ respectively. Now we apply the same principle to the CDA Green's functions used to propagate the

wavefield to the receiver locations in (6.14) to get

$$p^{sc}(\mathbf{y}, t) = \int_S l_\rho^{rec}(\mathbf{x}) l_{V_s}^{rec}(\mathbf{x}) \left[G_0(\mathbf{x}, \mathbf{y}, t) * \frac{\partial p^{sc}(\mathbf{x}, t)}{\partial \mathbf{x}_n} - p^{sc}(\mathbf{x}, t) * \frac{G_0(\mathbf{x}, \mathbf{y}, t)}{\partial \mathbf{x}_n} \right] dS_x, \quad (6.15)$$

where $l_\rho^{rec}(\mathbf{x})$ and $l_{V_s}^{rec}(\mathbf{x})$ are approximately inverses of the injection compensation factors. The pressure at location \mathbf{x} is affected by the density and S-velocity at location \mathbf{x} . The factors $l_\rho^{rec}(\mathbf{x})$ and $l_{V_s}^{rec}(\mathbf{x})$ approximately restore amplitude of the elastic pressure wavefield back to the amplitude under CDA conditions. Equation 6.15 is just the standard CDA boundary integral (6.14) on this CDA corrected pressure wavefield.

The scattered pressure wavefield computed at the receiver locations through (6.15) is then added to the background wavefield to obtain the perturbed wavefield. The background wavefield is obtained by convolving the background Green's function between source and receiver pixel with the source wavelet. We must also correct the amplitudes of the scattered and perturbed wavefields for differences in density and S-velocity between the source and receiver pixels. For the scattered wavefield these scalar compensation factors are folded into compensation factors $l_\rho^{rec}(\mathbf{x})$ and $l_{V_s}^{rec}(\mathbf{x})$. As a result the boundary integral automatically takes care of this for the scattered field. For the background wavefield we multiply the Green's functions between source and receiver with the scalar $l^0(\mathbf{x}_s, \mathbf{x}_r)$. Finally both wavefields are also compensated for S-velocity at the source pixel by $l_{V_s}^{inj}(\mathbf{x}_s)$. All these amplitude compensation factors, together with a schematic overview, are found in Appendix D.

6.2.7 Phase measuring

In order to measure the phase of a wavelet, we follow Fomel et al. (2010) who use the observation that a zero-phase wavelet has maximum correlation with its envelope. To measure the phase in this way, we use a grid-search procedure, with a grid over phase. We shift the wavelet in 1 degree increments and compute the correlation with the envelope of this shifted wavelet. The phase angle that gives the largest correlation is then the phase-shift of the input wavelet. This algorithm is summarized by Zhu

et al. (2011).

6.3 Results

Only when we taper to constant density and zero S-velocity along the boundary of the local solver do we satisfy the theoretical requirements that make the boundary integral and injection exact. We first investigate the local solver in this condition and verify its exactness on a simple two-layer model. We then apply the same configuration to a more realistic Earth model with varying elastic material properties around the boundary of the local solver. After that we will also empirically investigate the effect of tapering to nonzero S-velocity. We then show the results for removing the taper altogether and injecting on a variable density and shear velocity medium. Finally we show synthetic phase inversion results using the local solver.

6.3.1 Two layer example

We now investigate the local solver on a simple elastic two-layer model. In the local solver we taper the elastic true model to constant density and S-velocity at its boundary. For the local solver to be correctly implemented it needs to exactly match the wavefield generated by the full domain elastic solver on this tapered model. Finally we also want to verify that the phase of the reflection in this tapered simulation is similar to that of the true elastic model.

The true elastic model is shown in Figure 6-4, with source location at $x = 1.0$ km and $z = 0.1$ km. The receivers are placed along the entire horizontal extent of the model at the same depth as the source at $z = 0.1$ km. The red and blue boxes in the P-velocity model correspond to those in Figure 6-3. The elastic material parameters of the two layers are summarized in Table 6.1. A Ricker wavelet with peak frequency of 25.0Hz is used as source. We use a grid spacing of 3.25 m to prevent any appreciable dispersion induced changes in the phase of the wavelet while it propagates through the model shown in Figure 6-4. In this way we can focus entirely on phase changes due to reflection. The entire model is surrounded by Perfectly Matched Layer

(PML) boundary conditions to make individual events on the shot gather easier to distinguish.

The model in Figure 6-4 is considered the 'true' model in this experiment. We now investigate whether we can correctly model the elastic PP reflection at the layer boundary while tapering the density and S-velocity to CDA properties. Figure 6-5 shows the tapered density and S-velocity models in this test; the P-velocity is unchanged from Figure 6-4. Within the blue box the full elastic model (i.e. all three panels of Figure 6-4) are used in the local solver. We implement an exponential taper in the region between the blue and red boxes to transition from the EL model in the blue box to the CDA model in the red box and the exterior. The size of the taper is the same on all sides and is chosen conservatively as 240 m, which is three times the peak wavelength in the top layer. Table 6.2 shows all scaling terms, which are usually functions of position but for this simple experiment they are constant. Before the local solver can be used, we first compute the required background Green's functions using the CDA solver on the P-velocity model of Figure 6-4a. These CDA Green's functions are convolved with the source wavelet, multiplied by scalar constants l_p^{inj} and $l_{V_s}^{inj}$ and then injected in the local EL solver. The local solver computes the local elastic wavefield resulting from this injected wavefield, and the scattered pressure component is saved along its boundary. This scattered pressure along the boundary is then used in the boundary integral of (6.15) to compute the scattered wavefield at the receivers due to the elastic model update. The background wavefield is computed by convolving the background Green's function with the source wavelet and then multiplying the result with the scalar $l^0(\mathbf{x}_s, \mathbf{x}_r)$ from equation D.6. Since the P-velocity, density and S-velocity do not vary between source and receiver locations these scaling constants evaluate to 1.0. This background wavefield is added to the scattered wavefield from the boundary integral to obtain the perturbed wavefield at the receiver locations.

Figure 6-6a shows the pressure shot gather computed with the full domain elastic solver using the true model in Figure 6-4. Figure 6-6b shows the shot gather computed using the full domain elastic solver and the P-velocity model of Figure 6-4a and the

tapered density and S-velocity models of Figure 6-5. Finally Figure 6-6c shows the shot gather on the same tapered model using the CDA-EL coupled local solver. Both the full domain tapered shot gather and the local shot gather are multiplied by the scalar $l_{V_s}^{src}(\mathbf{x}_s)$. This compensation factor is needed since the S-velocity at the source location in the true model is non-zero. This reduces the amplitude of the wavefield in this simulation through the S-velocity scaling of (D.1). In both the tapered full domain simulation and the local solver simulation the S-velocity at the source pixel is zero because we are using the CDA approximation outside the local domain. As a result, the amplitude of the entire shot gather is larger than it should be. We correct this by multiplying by the scaling factor $l_{V_s}^{inj}(\mathbf{x}_s)$ as we explain in Appendix D.

The different events in Figure 6-6a are easy to distinguish due to the simplicity of the true model. The event marked with the number 1 is the direct wave. Event 2 is the reflection on the layer boundary 1.0 km below the source position. Event 3 at the horizontal coordinate $x = 3.0$ km shows the head wave separating from the reflection. This head wave is generated at the critical incidence angle, which is 30 degrees in this model. At larger horizontal coordinates (i.e. larger offsets) the reflection is post-critical and a noticeable phase-shift is observed. Using the full domain Elastic solver but replacing the true density and S-velocity models with their tapered equivalents introduces several extra events in Figure 6-6b. The density taper value is equal to the density of the top layer. As a result the top taper in Figure 6-5 only contains an S-velocity contrast. At horizontal coordinate $x = 1.0$ km, just below the source, the reflection of the incident P-wave on the S-velocity taper is negligible. The reflection on the top taper only becomes significant at grazing incidence angles as indicated by event 4. Just below the desired reflection two other minor reflection events can be seen, both crossing the arrow for event 2. These events are reflections on the left and bottom of the taper. In contrast to the reflection on the top taper, the bottom taper reflection happens at normal incidence angles as well since it includes a density contrast. Since the material properties in the bottom layer are larger than those of the top layer but we taper to the same density and S-velocity on all sides, the taper contrast is larger on the bottom. In addition the P-velocity in the bottom layer is twice

as large as in the top layer. While the size of the taper is three peak wavelengths on top, it is therefore only half that on the bottom. Despite these reinforcing factors the amplitude of the bottom taper reflection is relatively small compared to the reflection event 2 we want to model. In Figures 6-7 and 6-8 we investigate whether the taper reflections influence the reflection event whose phase we aim to model.

Figure 6-7 shows a trace at $x = 3.67$ km. The black, blue and red curves plot the traces of the full domain true, full domain tapered, and local domain simulations. Since there are no waves leaving the local solver and subsequently reentering in the simple 2-layer model, the elastic solver of Robertsson and Chapman (2000) does not discard any events. It correctly computes the wavefield resulting from the injected CDA background wavefield scattering on the elastic model updates. The boundary integral (6.15) is in theory also exact. That the red and blue curves in Figure 6-7 almost perfectly overlap is therefore not a surprise. It only demonstrates that the local solver is correctly implemented, at least for this geometry. The next thing to notice is that the black curve of the simulation is also matched quite accurately. The only appreciable deviation happens at $t \approx 1.6$ s. Looking at $x \approx 3.7$ km and $t \approx 1.6$ s in the shot gathers in Figure 6-6, this deviation appears to happen at the intersection between the head wave and event 4 (i.e. the top-taper reflection). Event 4 is not present in the true simulation since there is no taper, so it is not surprising that the black curve differs from the blue and red at this particular location.

In the flat two-layer example the arrival time of reflection event 2 is known exactly. We use a 0.15s window and measure the phase of the reflection in the true simulation, the full domain tapered simulation and the local simulation. Figure 6-8 shows that the phase of the true simulation is quite accurately matched by that of the local domain simulation. On the horizontal axis, we convert the horizontal coordinate to incidence angle. This conversion is trivial in this flat 2-layered example. The three colors correspond to those in Figure 6-7. The solid curves represent the measured phase of the windowed reflection. They start at a phase of $\approx 315^\circ$ degrees, or equivalently $\approx -45^\circ$ due to wrapping. This $\approx -45^\circ$ for the zero offset reflection is the well-known phase shift in 2D simulation where effectively an infinitely long line-source is

simulated. From these three solid curves the dashed phase shift curves are obtained by subtracting the zero incidence angle phase and removing the 360° phase wrapping. The tapered full domain simulation and the local domain simulation differ by at most 1 degree, which is not surprising due to the almost perfect match of the shot gathers in Figures 6-6b and 6-6c and the red and blue traces in Figure 6-7. The phase of the true simulation is also matched quite accurately with a maximum error of four degrees and average error of two degrees. We used the same material properties in our two layer study as Zhu and McMechan (2012) and our results show good agreement with the simulations in Figure 1b of their paper.

The size of the local elastic solver grid compared to the full elastic grid in this example is significant. The computational cost of a local elastic wavefield simulation is approximately half that of the elastic full domain simulation. This does not justify the cost of the 186 acoustic full domain simulations for the required Green's functions and the associated storage cost. We simply chose this large domain to demonstrate the principles behind the local solver. In the next section we use a smaller local domain solver in a more complex geometry. By no longer tapering to CDA material properties we can reduce the size of the taper and improve the accuracy of the phase of the modeled reflection.

6.3.2 Synthetic North Sea model

The synthetic model we focus on in this section has features similar to what is observed in the North Sea. A chalk layer with velocities sometimes exceeding 5000 m/s (Japsen, 1998) generates post-critical reflections. In this section we investigate whether we can correctly model these post-critical reflections with acceptable accuracy. Figure 6-9 shows the P-velocity, density and S-velocity whose combination we consider to be the true elastic model for this study. The high velocity layer at a depth slightly below 3.0 km represents the chalk layer. This chalk layer is composed of two regions, each with its own uniform P-velocity, density and S-velocity. One region is the bulk chalk and the other region centered around $x = 5.75$ km is a small (potentially hydrocarbon bearing) anomaly. The top 80.0 m of the model is sea with zero S-

velocity. The density is derived from the P-velocity model using the relation from Gardner et al. (1974). The S-velocity is obtained from the P-velocity assuming a Poisson solid. The only deviations from these relations are present in the ocean (no solid), top 700 m sediment (low V_s , unconsolidated) and the potential hydrocarbon bearing anomaly. For the latter we reduce the P-velocity and S-velocity by 300.0 m/s and 200.0 m/s and the density by 200.0 kg/m³ compared to the bulk chalk. The grid spacing used in this study is 10.0 m in both coordinate directions. We use a relatively low frequency wavelet of 6.0 Hz to prevent any appreciable dispersion. In practice an elastic solver with higher order accuracy would be used and the frequency of the modeled wavelet can then be increased. PML boundary conditions are placed on all sides of the model for simplicity. Sources and receivers are both placed at a depth of 10.0 m. We model six sources with locations 2.0 km, 2.5 km, 3.0 km, 3.5 km, 4.0 km and 4.5 km. Receivers are continuously placed between $x = 5.0$ km and $x = 12.0$ km. With this geometry for the sources and receivers, any chalk reflection will happen within the fully elastic blue box of Figure 6-9. Figure 6-10 shows the true shot gather simulated on the full elastic model of Figure 6-9 for a shot at $x = 2.5$ km. The blue lines window the top of reservoir reflection whose phase we want to model using the local solver. The red box highlights the region we zoom in on in later Figures for more detailed comparisons.

We first investigate whether the local solver can still accurately model the local reflection, with emphasis on phase modeling. We investigate multiple taper configurations to determine which one gives the most accurate response. Then as a demonstration of the value of post-critical reflections we perform a simple phase inversion using the local solver. The different local solver configurations 'Local A'-'Local D' are introduced gradually in this section. Table 6.3 serves as summary of their different taper configurations. The first configuration 'Local A' is the same configuration as in the two-layer example, where we taper to constant density and zero S-velocity. We use the amplitude compensation factors from the theory section for all four local solver configurations, but the values will differ for each configuration. In the two-layer case of Table 6.2 almost all of them evaluated to 1.0 for for Local A, the same is true

for Local A on the North Sea model of Figure 6-9. In the two-layer example only $l_{V_s}^{src}(\mathbf{x}_s)$ was unequal to one because the S-velocity at the source pixel was nonzero. In the North Sea example this final compensation factor evaluates to 1.0 since the source is in the sea with zero S-velocity.

Local A: Tapering the local model to CDA

In this section we investigate configuration Local A. Just as in the two-layer experiment, the background velocity model is CDA. The perturbation to this CDA background model is the elastic model in the blue box of Figure 6-9 including the elastic taper towards CDA properties along the outer red box, which is also the boundary of the local solver in this experiment. The boundary integral for the acoustic scattered field (6.15) outside of the outer red box (e.g. receiver locations) is again evaluated using the scattered field on the same outer red box. Since the model perturbation is entirely on the inside of the boundary integral, (6.15) is exact provided the scattered field on the boundary is so as well. We expect that the primary reflection of the wavefield on the chalk is modeled exactly the same using a full domain solver on the tapered model and using local solver configuration Local A on the tapered model with the boundary integral (6.15), because we satisfy all of the assumptions of the local solver for these events.

In Figure 6-10 we show the full domain elastic simulation on the true elastic model. The region in the red box is expanded in Figure 6-11a. Figure 6-11 shows shot gathers for different simulations for source position at $x = 2.5$ km. Figure 6-11b shows the full domain elastic simulation on the model which is fully elastic in the blue box and is tapered towards CDA on the outer red box. Figure 6-11c shows the coupled CDA-EL local solver on the same model within the outer red box. Figure 6-11d shows the full domain elastic simulation on a CDA model using the true P-velocity from Figure 6-9a with constant density and shear velocity. Arrow 1 shows the position of the top of reservoir reflection. Arrow 2 shows the location where the head-wave and the reflection separate. Arrow 3 shows the horizontal coordinate along which traces are extracted for comparison. Figure 6-12 plots traces extracted

from Figure 6-11 at $x = 8.0$ km and displays them in the same order. Notice how, as expected, the full domain tapered result is the same as the 'Local A' result, just as in the 2 layer scenario. The 'Local A' simulation quite accurately models the top of reservoir reflection, although some small differences are visible, particularly in amplitude. Comparing with the CDA simulation in green it is clear that using the elastic properties in the region of interests improves the accuracy of the modeled reflection

Heuristical approach

For the Local A simulation we used the region between the blue box and the outer red box in Figure 6-9a to taper the elastic model to the CDA model. The size of the taper substantially adds to the size of the local domain, reducing the computational benefits. Also in Local A the S-velocity is tapered from approximately 1300 m/s for the overlying sediment in the blue box to 0.0 m/s for the CDA properties at the outer red boundary. While we expect that the effect of transmission on phase is small, it is not zero especially when the taper contrast is large. In this section we investigate the three local solvers with different taper configurations to address these issues, Local B, Local C and Local D. None of these configurations taper to CDA zero S-velocity anymore, reducing the contrast across the taper. The background model on which the Green's functions are computed is still CDA. This means that the elastic model perturbations to the CDA background now intersect with the boundary of the local solver. This causes inaccuracies both when injecting the pressure wavefield from the CDA Green's function and when extrapolating the scattered pressure field to the physical receivers using the boundary integral (6.15). When injecting the wavefield in the local solver the constants $l_{V_s}^{inj}(\mathbf{x})$ and $l_{\rho}^{inj}(\mathbf{x})$ are no longer constants but vary in space. These scaling factors approximately compensate for smooth variations of their respective material properties. This scaling is not perfect, in particular when there are lateral variations in these constants along the injection boundary. In addition there are some inherent inaccuracies when transforming the pressure field stored at 9 layers around the boundary of the local solver into a velocity field and stress tensor with

with $\tau_{xx} \neq \tau_{zz}$ and $\tau_{xz} \neq 0$ when $V_s \neq 0$. By no longer tapering to CDA the model perturbation to the CDA background model is no longer limited to the interior of the boundary integral (6.15) as is required by theory. In addition to inaccuracies in the injected wavefield there is also an inaccuracy in propagating the scattered pressure wavefield of the chalk PP-reflection from the local solver to the receiver locations. By modeling the chalk PP-reflection in configurations Local B, Local C and Local D we investigate the impact for the model in Figure 6-9. Further theoretical developments are needed to quantify the errors in injection and extrapolation in general models.

First, the only difference between Local B and Local A is that Local B tapers to $V_s = 1300$ m/s instead of 0.0 m/s. Using a fixed nonzero V_s and varying V_p along the injection boundary results in spatially varying $l_{V_s}^{inj}(\mathbf{x})$ and $l_{V_s}^{rec}(\mathbf{x})$ from (D.2) and (D.4). Configuration Local C uses no taper at all and the blue box in Figure 6-9a is the local solver. It is worth noting that since the Earth in Figure 6-9 is a Poisson solid along the boundary of the blue box, the amplitude compensation factors $l_{V_s}^{inj}(\mathbf{x}) = \frac{2}{3}$ and $l_{V_s}^{rec}(\mathbf{x}) = \frac{3}{2}$ are constant. Configuration Local D only uses the thin horizontal taper on either sides of the blue box in Figure 6-9a and tapers to the same material properties as Local B. So the factors $l_{V_s}^{inj}(\mathbf{x}) = \frac{2}{3}$ and $l_{V_s}^{rec}(\mathbf{x}) = \frac{3}{2}$ along the blue portion of the top boundary, while they spatially vary along the red portions of the boundary due to the tapering of the S-velocity. The density amplitude compensation factors again vary less than those for the S-velocity.

Figure 6-13 shows the shot gathers for this heuristic approach, plotted in the same window as those in Figure 6-11. The three simulations in Figure 6-13 look similar both to one another and to the Local A simulation in Figure 6-11c. One noticeable difference is an artificial high amplitude scatter event for the Local C simulation in Figure 6-13b, marked with a red arrow. At larger x coordinates outside the Figure it asymptotically merges with the chalk reflection. A similar event with apex at $x = 3.0$ km is present as well outside the plotting window. The location of these events coincides with the x coordinates of the left and right boundaries of the blue box, which is the boundary of the taperless solver Local C. Later we investigate this observation in more detail.

Figure 6-14 shows traces extracted at $x = 8.0$ km from the simulations in Figure 6-13. The vertical scale is different than in Figure 6-12, but the true trace in black is exactly the same. We see that at $x = 8.0$ km the top of reservoir reflection is computed almost identically in the Local C and Local D configurations. This agreement is observed everywhere, except for x much smaller than 3.0 km or much larger than 8.0 km when the high amplitude scatter event asymptotically merges with chalk reflection. Similar to Local A, the three traces in 6-14 also match the chalk reflection quite accurately. In order to quantify the match we measure the phase of the reflection within a 0.2 s window, marked by the blue lines in Figure 6-10. The five extracted gathers are plotted in Figure 6-15. Notice how at $x \approx 9.5$ km for Local C the high amplitude scatter event comes into the window and merges with the reflection.

For each trace in the windowed shot gathers of Figure 6-15 the phase is determined by using the algorithm of Fomel et al. (2010), summarized by Zhu et al. (2011). Figure 6-16 shows the correlation coefficients, with the solid red line following the maximum correlation in the individual plots for each receiver coordinate. Similar to Figure 6-8 the receivers nearest to the source at $x = 2.5$ km have measured phase close to 315° . This -45° shift for near offset is the result of 2D effects. At certain offsets, events intersect with the chalk reflection and as a result the solid red line oscillates rapidly. See the region between $x = 8.25$ km and $x = 9.75$ km for instance in all five plots. At these locations the combination of the direct wave and ocean bottom reflection intersects with the chalk reflection. The ocean bottom reflection is different in the true simulation where it is affected by the elastic contrast at the ocean bottom and in the local simulations where the ocean bottom reflection is CDA. The superposition of this event with the chalk reflection will have a different picked phase than that of the chalk reflection alone. But additionally this change will not be reproduced by the local solver since the ocean bottom reflection has different physics than in the true simulation. Another example is the region between $x = 5.0$ km and $x = 5.5$ km for the true simulation. Here a surface wave intersects the chalk reflection, perturbing the picked phase. One approach to remove some of these oscillations is to enforce

smoothness on the picked curve using regularization (see Fomel et al. (2010) for more details). Instead we decide to mute certain positions where events crossing the chalk reflection perturb its phase measurement. The mute windows for the source at $x = 2.5$ km are shown with gray shading in Figure 6-17a. A more desirable option would be to use processing to remove these events, but this is outside the scope of this chapter.

All of the five curves of Figure 6-16 are merged into Figure 6-17a. In order to get the phase shift of the reflection as a function of position, approximately 45° needs to be added to correct for 2D effects. Including this shift, at large offsets (large incidence angles) the phase shift approaches 180° . At intermediate angles the shift is more than 180 degrees. This very large phase shift is caused by the presence of two critical angles. The first critical angle originates from the P-velocity in the chalk which is larger than the that overlying sediment. In addition the S-velocity in the chalk is also larger than the P-velocity in the overburden. This causes the second critical angle, now for the PS converted wave. This angle also influences the phase of the PP reflection we focus on. Figure 6-17b shows the absolute (wrap corrected) phase error with respect to the true phase for the different simulations. Notice that Local C becomes worse at offsets larger than approximately 9.5 km because the scatter artifact merges with the real postcritical reflection due to the relatively long dominant period of the wavelet.

Table 6.4 contains the measured error for the several source locations. The chalk reflection window and the mute windows differ for each source location but are the same for all simulation methods. The average is the arithmetic average of the phase error at all the source locations. The average does not take into account that the number of acceptable (i.e. non-muted) phase picks varies with source location. Based on this average the ranking of the methods, in ascending order, is Local B, C, A and then Local D. While Local C performs slightly worse than Local A, the computational domain for Local C is much smaller than for Local A which makes it more desirable. But the high amplitude artificial reflection event in Local C negatively impacts the measured phase. This event is investigated in Appendix F. Local D has a slightly larger computational domain than Local C, but avoids the artificial scatter event and

has the overall highest accuracy. Local D is also computationally more efficient than the large taper tests of Local A and Local B. Because of the high observed accuracy and small size of the computational domain we prefer Local D for the particular model we study here. This is the local solver we use in the inversion experiment.

6.3.3 Inversion

We now investigate whether the phase of the reflection event, generated by the local solver, can be used to invert for material properties at a reflector of interest. We do this by minimizing the least squares misfit

$$\chi(\mathbf{m}) = \frac{1}{2} \langle \mathbf{d} - \mathbf{u}(\mathbf{m}), \mathbf{d} - \mathbf{u}(\mathbf{m}) \rangle = \frac{1}{2} \langle \mathbf{r}(\mathbf{m}), \mathbf{r}(\mathbf{m}) \rangle \quad (6.16)$$

between observed phase vector \mathbf{d} and simulated phase vector \mathbf{u} . The simulated phase \mathbf{u} is a function of the model parameters \mathbf{m} . The phase residual $\mathbf{r}(\mathbf{m})$ is corrected for phase wrap to get absolute phase differences. Figure 6-17b shows the phase residual for a source at $x = 2.5$ km where the model \mathbf{m} on which \mathbf{u} is simulated is the true model. Because none of the local solvers models the reflections exactly, there is a phase residual even on the true model. Phase residuals in the muted regions are not included in the vector $\mathbf{r}(\mathbf{m})$. The non-muted phase residuals for all sources are concatenated to form the vector $\mathbf{r}(\mathbf{m})$.

The question of how to parametrize the model is important. Zhu and McMechan (2012) show that in general the amplitude and phase of a reflection depend on 4 ratios of material properties at a reflector. Since the full elastic material contrast at a reflector is defined by 6 quantities (P- and S-velocity and density on both sides of the reflector), at least two of them need to be fixed. Zhu and McMechan (2012) fix the P-velocity and density in the upper layer and demonstrate that the method is sensitive to bias in these picks. In this study, we assume the overburden is known and invert only for the material properties of the bulk chalk and the potentially hydrocarbon bearing anomaly at $x = 5.75$ km. Both these regions are defined by P-velocity, density and S-velocity, so the model vector \mathbf{m} contains six elements. In each iteration i we

form the normal equations

$$\mathbf{J}_i^T \mathbf{J}_i \mathbf{p}_i = -\mathbf{J}_i^T \mathbf{r}_i \quad (6.17)$$

for current model estimate \mathbf{m}_i in order to minimize (6.16). Here p_i is the search direction we want to obtain. The Jacobian matrix \mathbf{J}_i is evaluated discretely using 6 forward models. In each forward model, we perturb one material parameter by a small amount (e.g. we find 25.0 works well both for velocities in m/s and density in kg/m³) and then obtain one of the six columns of \mathbf{J}_i . The search direction is then obtained by inverting the small six by six matrix $\mathbf{J}_i^T \mathbf{J}_i$

$$\mathbf{p}_i = -(\mathbf{J}_i^T \mathbf{J}_i)^{-1} \mathbf{J}_i^T \mathbf{r}_i \quad (6.18)$$

With this search direction a line-search is initiated where we update model \mathbf{m}_i to \mathbf{m}_{i+1}

$$\mathbf{m}_{i+1} = \mathbf{m}_i + \alpha_i \mathbf{p}_i \quad (6.19)$$

Since a Newton method is used to compute the search direction \mathbf{p}_i in (6.18), we first attempt step length $\alpha_i = 1.0$. If this does not decrease objective function (6.16), then the step length is reduced and (6.16) is evaluated again. This continues until an improved model \mathbf{m}_{i+1} is found. Based on the performance observations in Table 6.4 we use the Local D solver in the inversion. Local D generates the local wavefields from which we extract the simulated phases $\mathbf{u}(\mathbf{m})$. Six sources are used spaced equally between 2.0 and 4.5 km, each with their respective mute windows.

Table 6.5 shows the initial, inverted and true model parameters for the potentially hydrocarbon bearing anomaly and the bulk chalk. The evolution of the model \mathbf{m}_i as a function of iteration number i is plotted in Figure 6-18. Each curve is normalized by the true value, so all curves should approach 1.0 in case of a perfect inversion. The subscript 1 refers to the material properties of the potentially hydrocarbon bearing anomaly in the chalk anticline at $x \approx 5.5$ km and subscript 2 refers to the bulk chalk. The first 5 iterations show large and sometimes oscillatory updates. In particular the density appears to diverge at the start. But during later iterations the convergence

behavior becomes very smooth and incremental.

Figure 6-19a shows the evolution of the phase for a collection of selected iteration numbers. Figure 6-19b shows the wrap corrected phase residuals of Figure 6-19a for the same iterations. These figures also show the phase curve for the Local D simulation on the true local model. The simulation on the true local model is the same as in Figure 6-17a and 6-17b. We see in Figure 6-19b the phase residual progressively gets smaller and smaller as the inversion progresses. We do not expect to get a perfect model recovery, since we have not perfectly matched the phase when using the acoustic/elastic coupled solver. As a result of this mismatch, we find models that fit our modeled data better than the true model. This is seen in Figure 6-20 where we plot the objective function during inversion and mark the objective function on the true model with the dashed black line. Continuing the inversion beyond 23 iterations will therefore never result in the true model, since this has a higher objective value associated with it. Despite these inaccuracies we see in Table 6.5 that all the model parameter estimates at iteration 23 have errors of less than 4%.

6.4 Discussion

The local solver introduced in this chapter is based on an elastic solver which is fourth order accurate in space. Such a low order stencil is not effective at preventing grid dispersion. As a result we used more points per wavelength than is customary in practice. For instance for the two layer example the peak frequency of the Ricker wavelet is 25 Hz. The P-velocity in the top layer is 2000 m/s so the peak wavelength is 80.0 m. With the used grid spacing of 3.25 m the number of points per wavelength is approximately 25. This small grid spacing increases computation time of the solver and also increases the storage cost for the Green's functions. With higher order accuracy stencils or other dispersion optimized stencils the number of points per wavelength can be reduced, but stencil evaluation may be more expensive. Depending on the particular implementation this may reduce the time required for computing the full domain Green's functions and performing the local solves. The increase in grid

spacing and time step can will both reduce the storage cost for the Green's functions, but more than 9 layers of CDA Green's functions will need to be stored around the boundary of the local solver. Whether the overall approach is more efficient will depend on the stencil size with respect to its dispersion behavior.

In its Local A configuration the local solver generates exactly the same results as a full domain elastic solver on the same tapered elastic-CDA model. The amplitude compensation factors are all constants and the boundary integral in equation 6.15 is just a scalar multiplied version of the exact CDA integral in equation 6.14. The amplitude compensation factors only scale the resulting wavefield to approximately compensate for elastic properties at the source and receiver locations in the true model, since those affect the pressure through the smooth scaling factors given in equations D.2 and D.3. The S-velocity scaling factor in equation D.2 naturally follows from the governing equations of the elastic solver in equation D.1. We derive the density scaling factor (D.3) in Appendix E by making the plane-wave approximation. Empirically we observe the same scaling when using point-sources. The wavefields generated by Local B, C and D no longer correspond to equivalent full domain elastic simulations on some tapered model. They manipulate the injected wavefield on a pixel by pixel basis through the amplitude compensation factors in equations D.2 and D.3. When extrapolating the wavefield back to the receivers they use similar spatially varying amplitude compensation factors within the boundary integral of (6.15). Both the scaled injection and extrapolation are not exact. For the particular North Sea model of Figure 6-9 we observe a good result for Local D, but without a rigorous theoretical framework it is hard to predict whether the Local D configuration transfers well to other models. Based on observations in this chapter we do have preliminary empirical guidelines in the absence of theoretical justification.

Based on the observations of Local C it appears that strong contrasts in elastic properties along the boundary generate artificial scatter events with diffraction-like behavior. The artificial scatter event happens when the downgoing wave hits the intersection of the local solver boundary and the reflector. Since the artificial scatter event is generated on the reflector of interest it will merge with the real reflection

event at a particular position, x . This merging with the target reflection event is what should be avoided since it affects the measured phase. Therefore it is advisable to taper the density and S-velocity in the direction of the reflector. With this taper the reflector no longer has a contrast in density and S-velocity at the boundary and the artificial scatter event is prevented in Local D in contrast to Local C. While we expect that other untapered elastic reflectors which intersect the boundary of the local solver will also cause diffraction events, those should not merge with the reflector of interest since they originate from a different reflector. While these events will not merge with the reflection of interest, they can still cross it and affect the phase at that crossing. In general it is important to avoid jumps in elastic material parameters along the boundary. We choose to prevent strong elastic contrasts by a suitable choice of tapers. An alternative approach is to pick the local solver boundary along a contour where it does not intersect layers with strong elastic material contrasts. Perhaps a more generally applicable routine would be to smooth the entire P-velocity, S-velocity and density models except for the reflector of interest.

It should be noted that the chalk reflection is in fact the combination of the top- and bottom chalk reflection with the 6 Hz Ricker we use. From a practical point of view this superposition does not matter to our approach. Both the true data and the simulated data contain this superposition. When minimizing the misfit in phase in the inversion we observe in Figure 6-18 that the P-velocity and density are not as well recovered as the S-velocity. This good recovery of the S-velocity may be helped by the chalk layer having a PS-critical angle. Zhu and McMechan (2014) observes something similar and show that a combined inversion using both amplitude and phase information gives slightly better results. Intuitively it makes sense that amplitude information helps constrain the P-velocity and density through the impedance. Using amplitude information also makes better use of near-offset reflections, whose negligible phase shift is hard to use in a phase only inversion. It is hard to accurately model the amplitude of a reflection since you need to model the (elastic) transmission losses in the entire overburden. But Zhu and McMechan (2014) show that despite these inaccuracies the addition of amplitude to the inversion

does give a better model estimate. In Figure 6-14 we see that the modeled amplitude of the Local D trace is close to that of the true trace. It should be noted that the overburden sediment in this chapter includes several layers, but that the contrast at these layers is weak. Most variation in material parameters happens smoothly within these sediments. Future studies focusing more on the amplitude should investigate whether the amplitude of the local reflection remains relatively accurately modeled in a more heterogeneous overburden. Hobro et al. (2014) introduce a method for compensating the amplitude in acoustic simulations for elastic effects like transmission loss, which may help with amplitude inversions. While outside the scope of this chapter, it would be interesting to include the amplitude modeled by the local solver to the current phase-only inversion. Another effect worth studying is when high-contrast layers in the overburden become post-critical. This may limit the incidence angles that are able to pass this reflector and will consequently influence the incidence angles at the reflector of interest. The effects of a potentially reduced range of incidence angles on the stability of the inversion is a subject of future work.

In this chapter we have investigated coupling a CDA background model to a local elastic model. The reason for doing this is to reduce the cost of precomputing the Green's functions by leveraging highly efficient CDA wavesolvers. The price we pay for this is that the tapers make the local domain slightly larger than the region of interest (i.e. the blue box in Figure 6-9), which makes the local solves more expensive. In addition there is a small reduction in accuracy. An intermediate approach would use Variable Density Acoustic (VDA) Green's functions, which increases computational cost but potentially also the accuracy because no density compensation factors are required anymore. If precomputing a large set of full-domain elastic Green's functions is not a problem then the alternative is to implement the original local solver introduced by Robertsson and Chapman (2000). Since the background model and the local (potentially perturbed) model are both elastic they can be seamlessly connected. As a result the local solver will generate the primary reflection of interest with the same accuracy as the full domain solver. Since no tapers are needed the local simulations are also slightly faster. This approach may be less beneficial if no detailed elastic

overburden model is available. If many small scale perturbations with respect to the wavelength are present in the true Earth, then the effective anisotropy this induces cannot be accurately modeled using an acoustic solver (Cance and Capdeville, 2015). In those cases the Green's functions should be computed using an elastic solver. If the true Earth has inherent anisotropy parameters as well then an elastic solver may also be mandated. Whether it is possible to link anisotropic pseudo-acoustic Green's functions to a local elastic solver has not been investigated yet.

The computational cost of advancing the local solver one time-step is cheaper than advancing the full domain solver. The speedup factor is proportional to the difference in the size of the full domain and the local domain. An additional cost saving is available to the local solver since the reflection of interest won't enter the local domain until some time $t_0 > 0$. In theory we only need to model between this time t_0 and the time t_1 that the wavefield leaves the local domain. While this is not done in this chapter, this approach would improve the computational benefits of the local solver even more. The cost of extrapolating the wavefield to the receivers is determined by the cost of evaluating the convolutions through the FFT in the boundary integral.

More theoretical work is needed to understand the range of conditions to which the empirical amplitude compensation factors apply. We provide empirical guidelines for how to taper the local model to avoid artificial scatter artifacts in the boundary integral. But theoretical developments would give more confidence about to what degree the observations on the models in this chapter translate to arbitrary models. If finite frequency effects in the overburden are insignificant it would be interesting to investigate whether the local elastic solver can be coupled to the acquisition using a ray based approach. Not only would this drastically reduce the precomputation cost, but it would also reduce the storage cost significantly. Also it would be interesting to use phase measurements in time-lapse applications. In this chapter we use only phase to invert for material parameters in one state. But in a time-lapse setting we could use the observed phase at two or more moments in time and then invert for associated time-lapse reflector change. One way of doing this would be by using

a double difference approach to potentially reduce the effect of inaccuracies in the baseline reflector on the inversion. Using amplitude together with phase will further constrain the inversion. Finally other versions of the weighted boundary integral can be investigated as well, for instance by first scaling the wavefield before evaluating the normal derivatives.

6.5 Conclusions

In this chapter we introduce a mixed local solver where CDA Green's functions are used to couple a local elastic solver to the rest of a larger model. We first investigate this method on a simple two-layer model where this mixed local solver matches the full domain elastic reflection response almost perfectly. After this we try the local solver in several different configurations, with different tapers and associated accuracies and computational costs, on a more complicated heterogeneous velocity model. The Local A configuration directly corresponds to an equivalent tapered elastic full domain simulation. The other three configurations have no full domain equivalent and are heuristic in nature. For these configurations we use spatially varying empirical scaling factors to make the reflection response from the mixed local solver approximate the full domain elastic response. We observe that the Local D configuration, in which we use tapers only on the vertical boundaries of the local domain, is computationally efficient and that it also models the reflection of interest with higher accuracy than others. By studying the artifacts introduced in Local C we observe that the presence of elastic contrasts along the interface introduces artificial scatter events, which reduces the accuracy of this local solver. In Local C this event is caused by the intersection of the untapered reflector of interest with the boundary of the local solver. This scatter event merges with the real reflection we want to model, impacting the measured phase adversely. We therefore suggest tapering the density and S-velocity along the reflector of interest. We show that for a phase-only inversion we are able to quite accurately retrieve the model parameters at the reflector of interest. In combination with the computational efficiency of the local solver this demonstrates the potential of this

approach.

6.6 acknowledgements

This chapter of the thesis is based on a submitted paper (Willemsen, Malcolm ?? ...). I therefore acknowledge the constructive and insightful interactions with Alison Malcolm. We thank Xinfu Zhu and Filippo Brogini for insightful discussions. This work is supported by Chevron and with grants from the Natural Sciences and Engineering Research Council of Canada Industrial Research Chair Program (IRCPJ 491051-14) and the Research and Development Corporation of Newfoundland and Labrador (5405.1085.104) and by the Hibernia Management and Development Corporation. We are grateful to Fons ten Kroode for providing us with a North Sea velocity model, which we modified for the example in this chapter. We would also thank MIT and the ERL consortium members for their support.

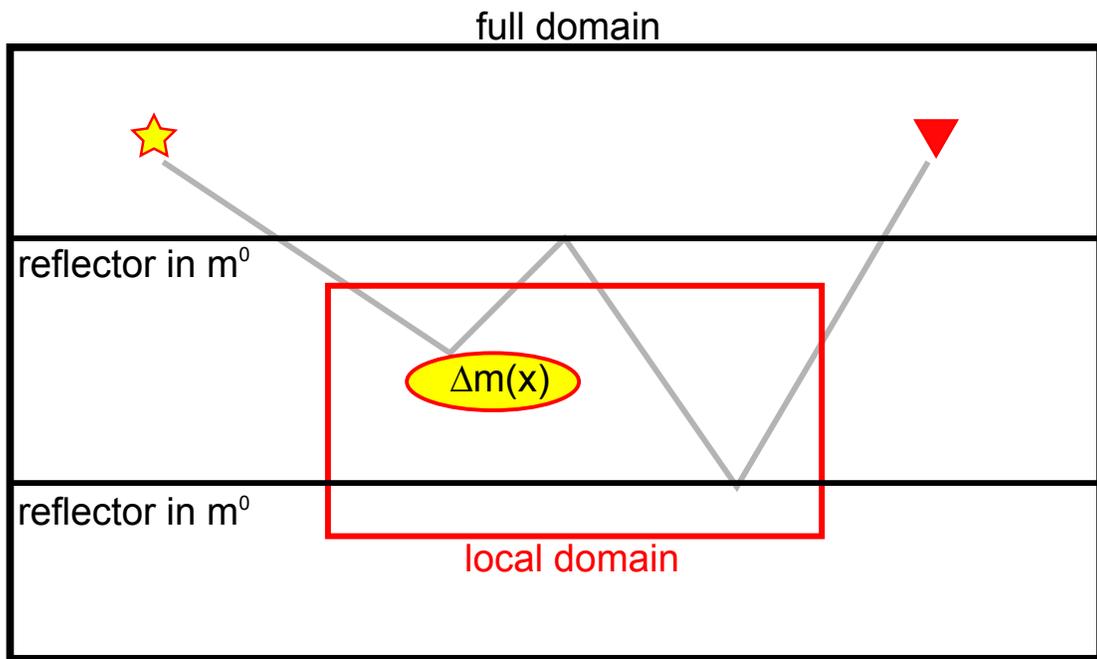


Figure 6-1: Schematic drawing of a local solver (red box) within a full domain solver (black box). The local solver used in this study is a modification of the local solver introduced by Robertsson and Chapman (2000). It discards waves scattering on local model updates $\Delta m(x)$, which then exit and subsequently reenter the local domain, like the gray event.

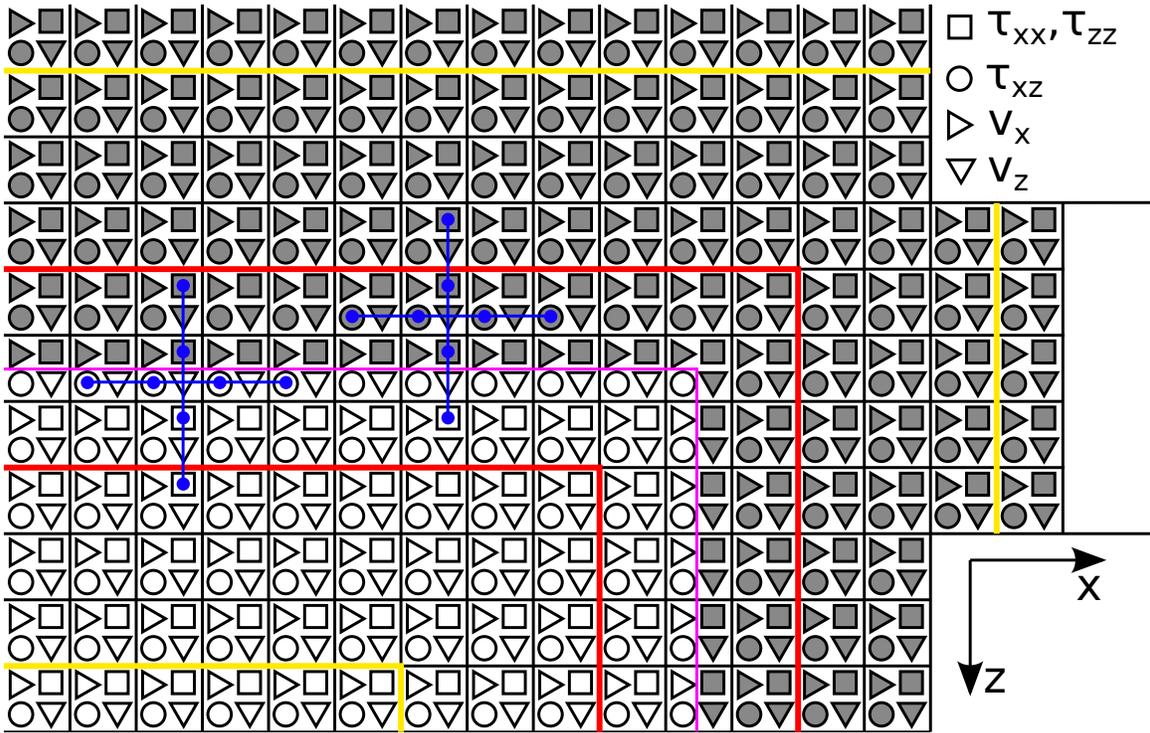


Figure 6-2: Staggered elastic grid around the top-right boundary of the elastic local solver. The various colored lines are used at different points in the chapter, where their purposes are also described.

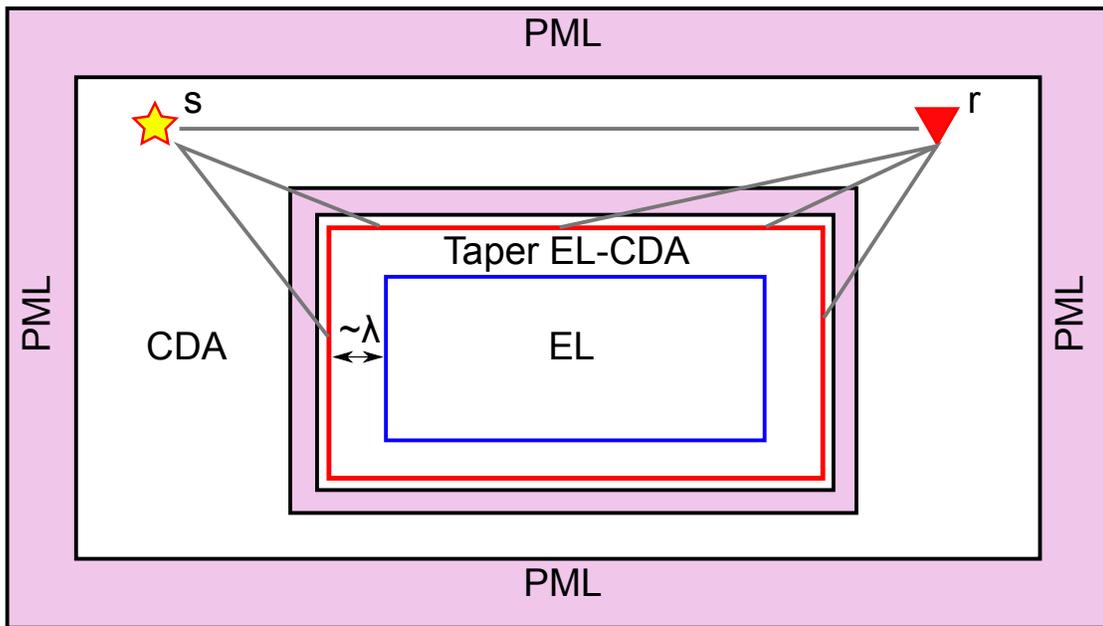


Figure 6-3: Schematic of the local solver. The gray lines represent the required background CDA Green's functions between the sources, receivers and boundary of the local solver. The model update within the local solver is elastic, with the region between the red and the blue line representing a ρ and V_s taper. The blue box contains exactly the part of the elastic model we want to model.

	Top layer	Bottom layer
V_p (m/s)	2000.0	4000.0
ρ (kg/m ³)	2000.0	2300.0
V_s (m/s)	880.0	1540.0

Table 6.1: The elastic material parameters for the two-layer example.

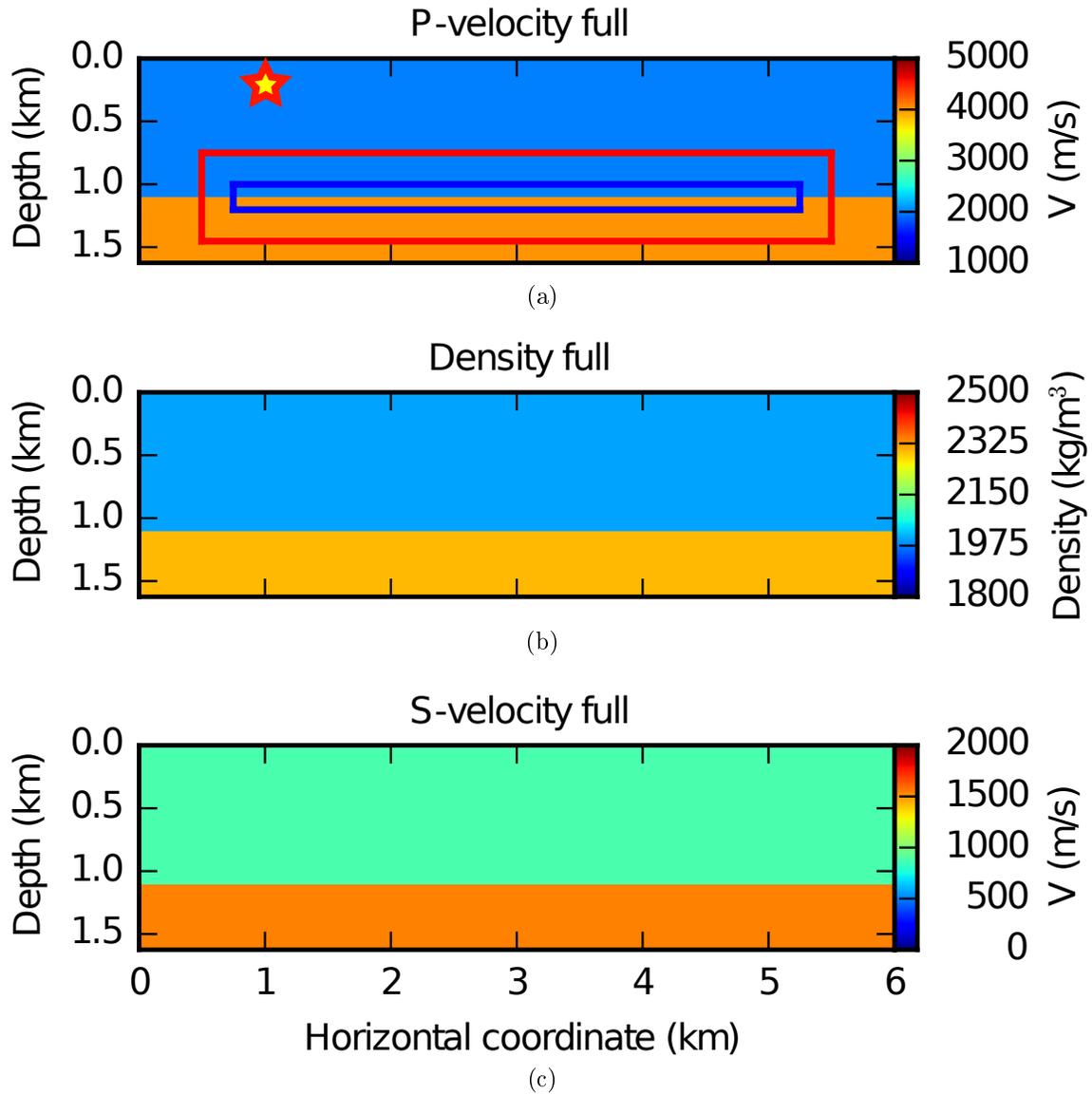


Figure 6-4: True elastic model, with the source at $x = 1.0$ km and $z = 0.1$ km. Density and S-velocity models are tapered from fully elastic inside the blue box to CDA at the red box in Figure 6-5. Material properties of the two layers are summarized in Table 6.1.

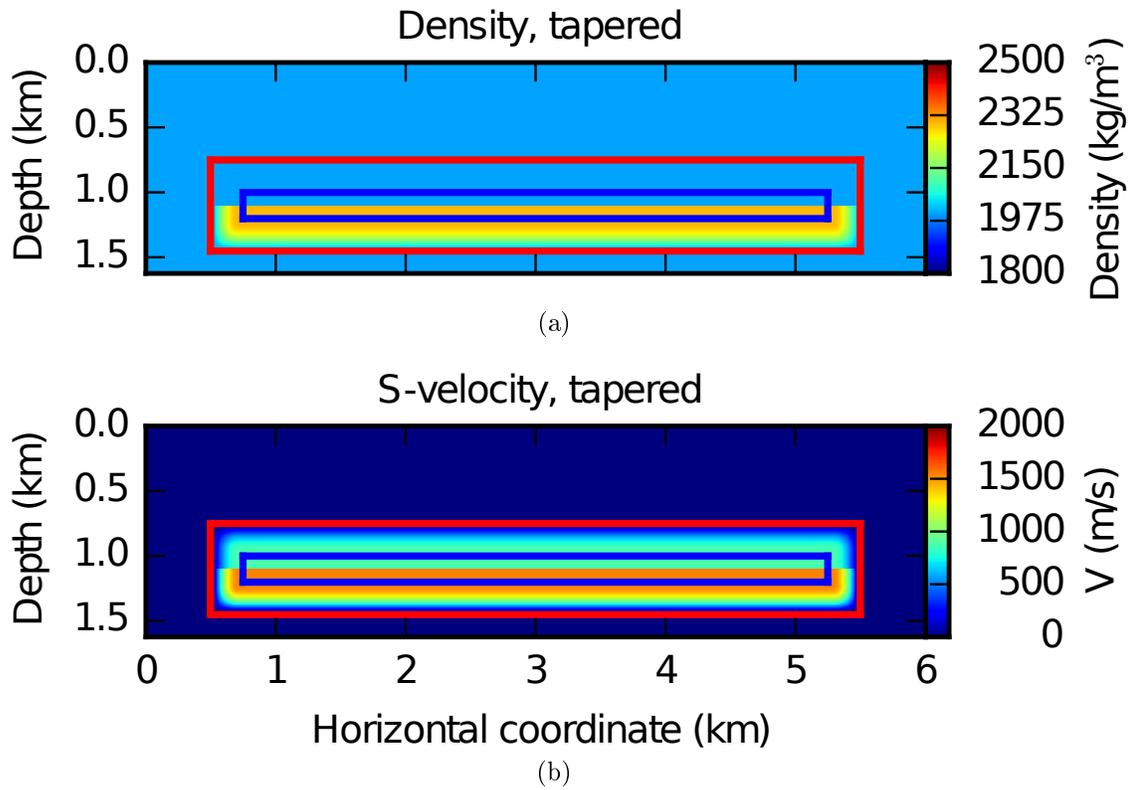


Figure 6-5: Tapered density and shear velocity models from Figure 6-4. Inside the blue box the models are fully elastic and the same as in 6-4. Between the two boxes the model is tapered towards CDA material properties, which implies the V_s taper value is 0.0 m/s. We choose the constant density taper value $\rho = 2000.0 \text{ kg/m}^3$, which is equal to the density of the top layer in the full elastic model of Figure 6-4.

Scaling terms for two layer example	
Scaling factor	Value
$l_{\rho}^{inj}(\mathbf{x})$	1.0
$l_{V_s}^{inj}(\mathbf{x})$	1.0
$l_{\rho}^{rec}(\mathbf{x})$	1.0
$l_{V_s}^{rec}(\mathbf{x})$	1.0
$l^0(\mathbf{x}_s, \mathbf{x}_r)$	1.0
$l_{V_s}^{inj}(\mathbf{x}_s)$	$\frac{2000^2 - 880^2}{2000^2} = 0.8064$

Table 6.2: All scaling factors used in the experiment defined by Figures 6-4 and 6-5. For this particular experiment they reduce to constants.

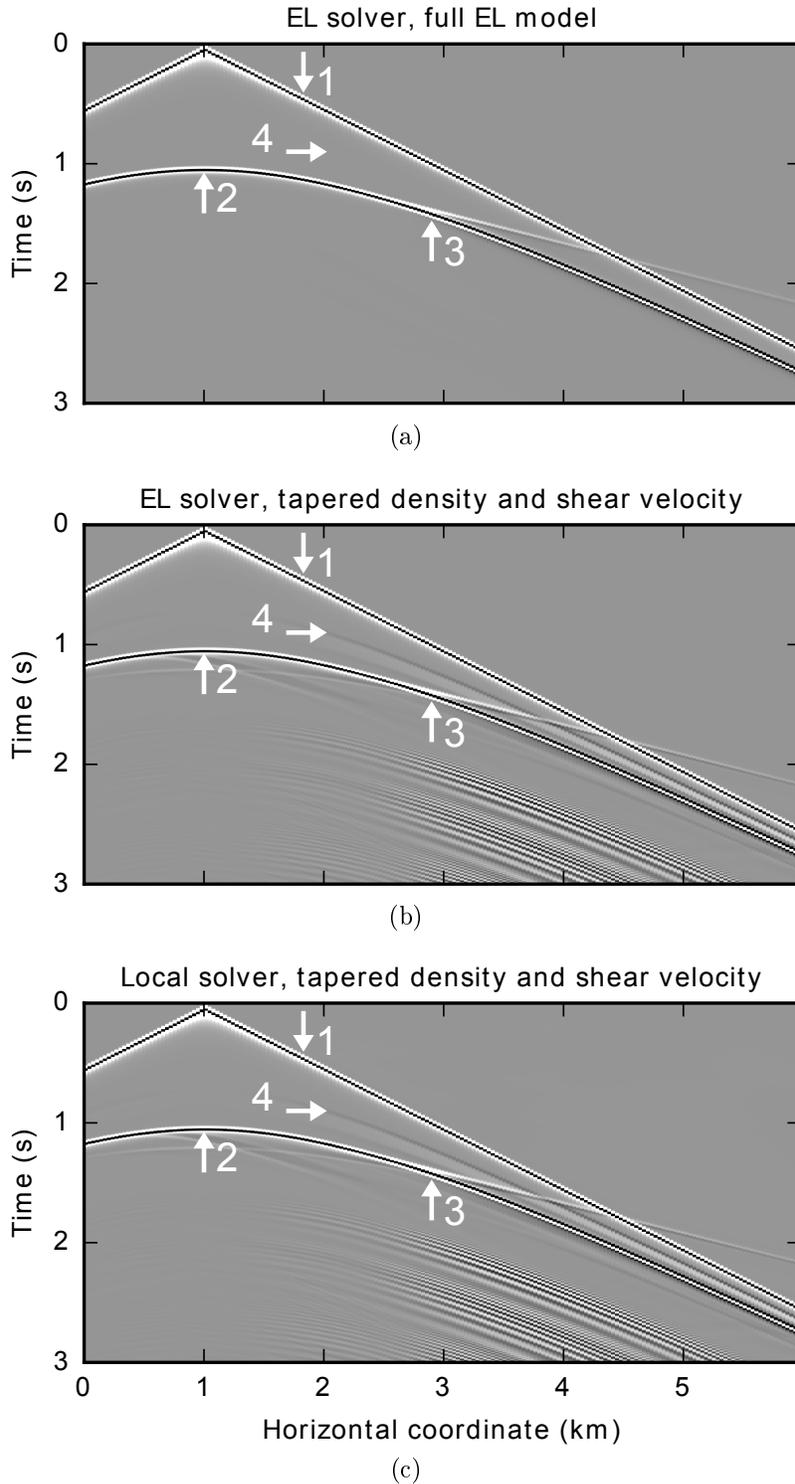


Figure 6-6: Gathers. The number 1 marks the direct wave. The number 2 marks the reflection. The number 3 marks the location where the headwave peels off and where we clearly enter the postcritical regime (to the right, larger offset). The number 4 marks the reflection of the incident pressure wavefield on the top taper boundary. This event is not present in the true simulation because there is no taper. The same amplitude clipping is applied to all three simulations so the color scales correspond.

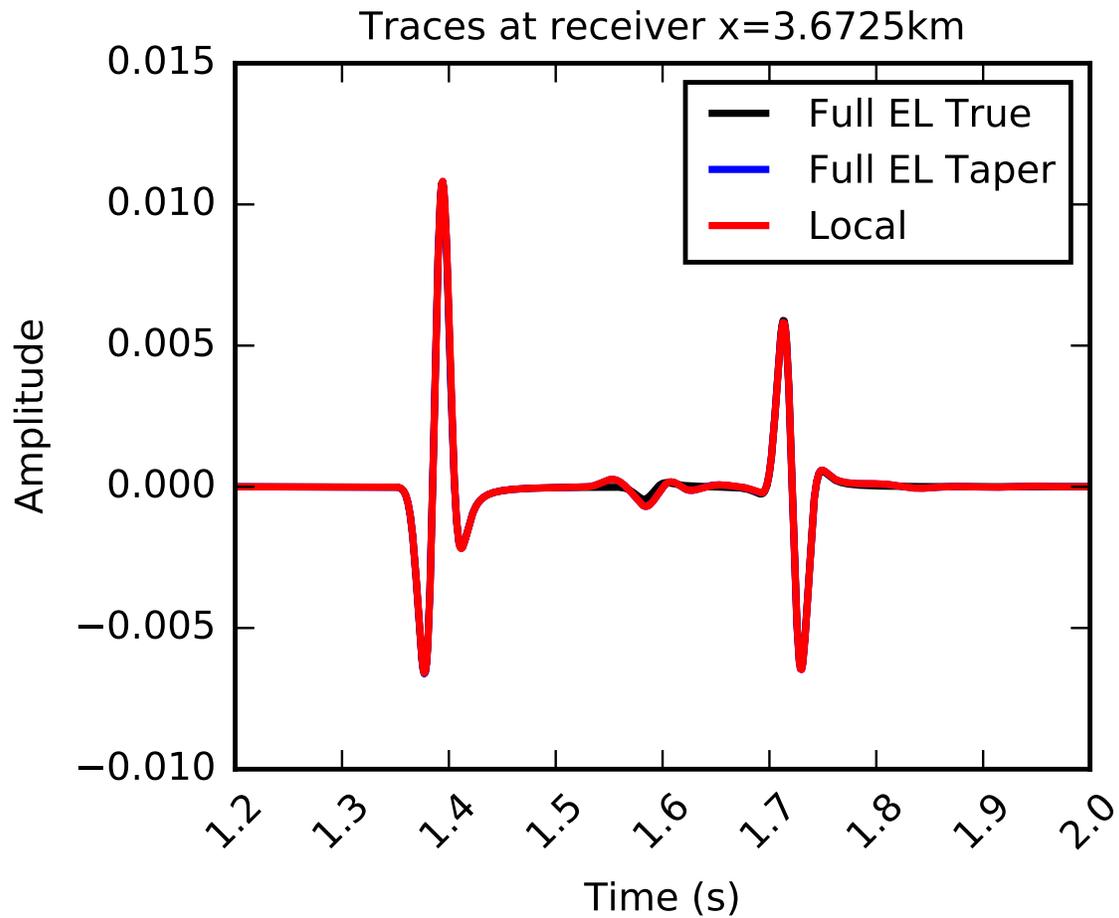


Figure 6-7: Traces at $x = 3672.5$ m for a shot at $x = 1.0$ km. Applying the described single scalar amplitude correction to the full domain taper and local shot gathers. Notice the excellent agreement between the different models. Only at the intersection of events 3 and 4 (between 1.50 s and 1.65 s) do we see some differences. The simulation on the true model does not have the taper influence.

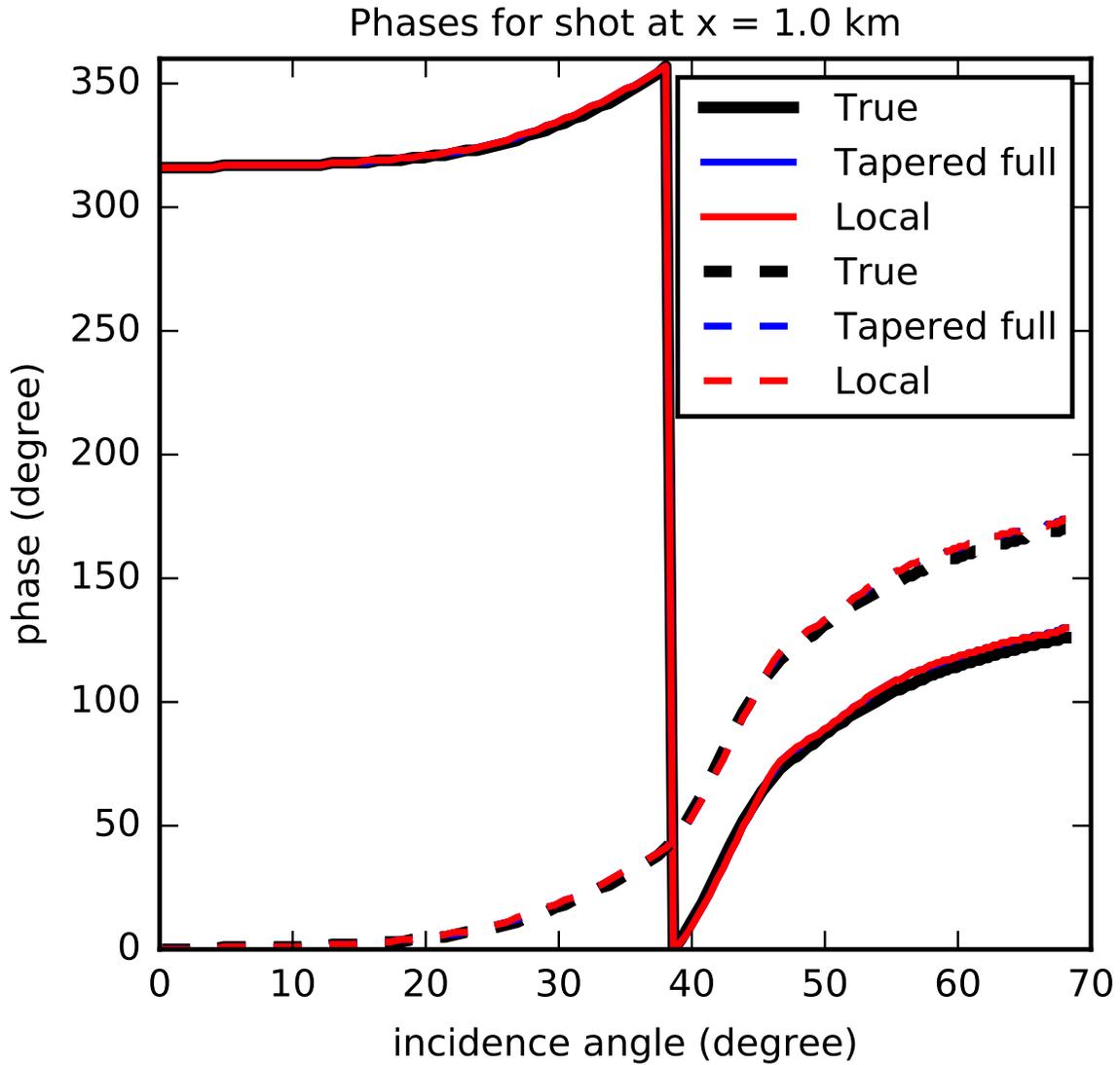


Figure 6-8: Phase shift with respect to incidence angle at the reflector. The dashed lines are obtained by subtracting the zero offset reflection phase (which is approximately shifted by -45 degrees because we are using a 2D model). The phase error between the estimated phase in the true and tapered/local result is four degrees at most.

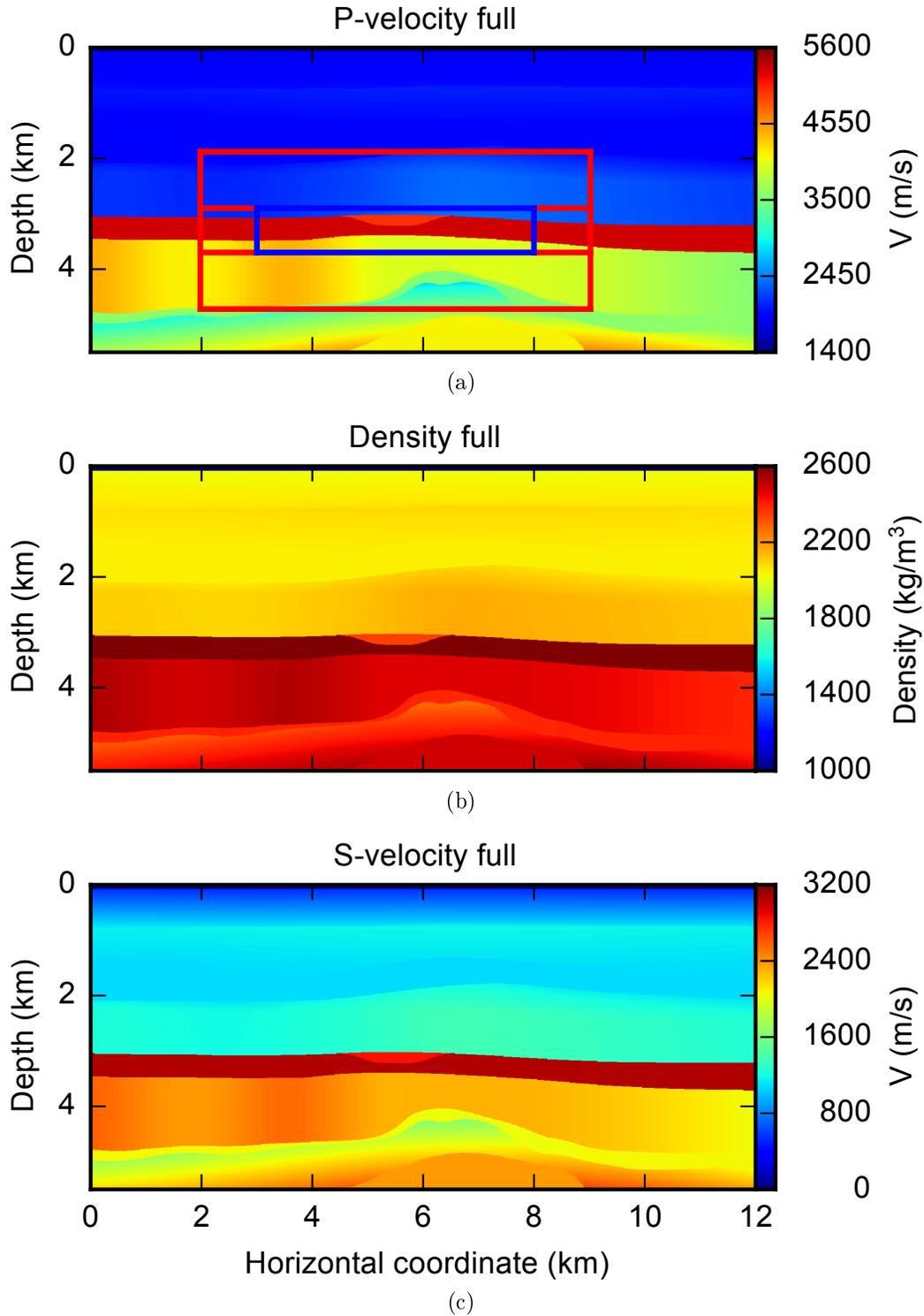


Figure 6-9: True V_p , ρ and V_s for the North Sea model. The image of V_p is the background velocity model on which the CDA Green's functions are computed. The outer red box and inner blue box correspond with those in Figure 6-3. The lines connecting the outer red box and inner blue box are used in subsequent tests where parts of the taper are eliminated.

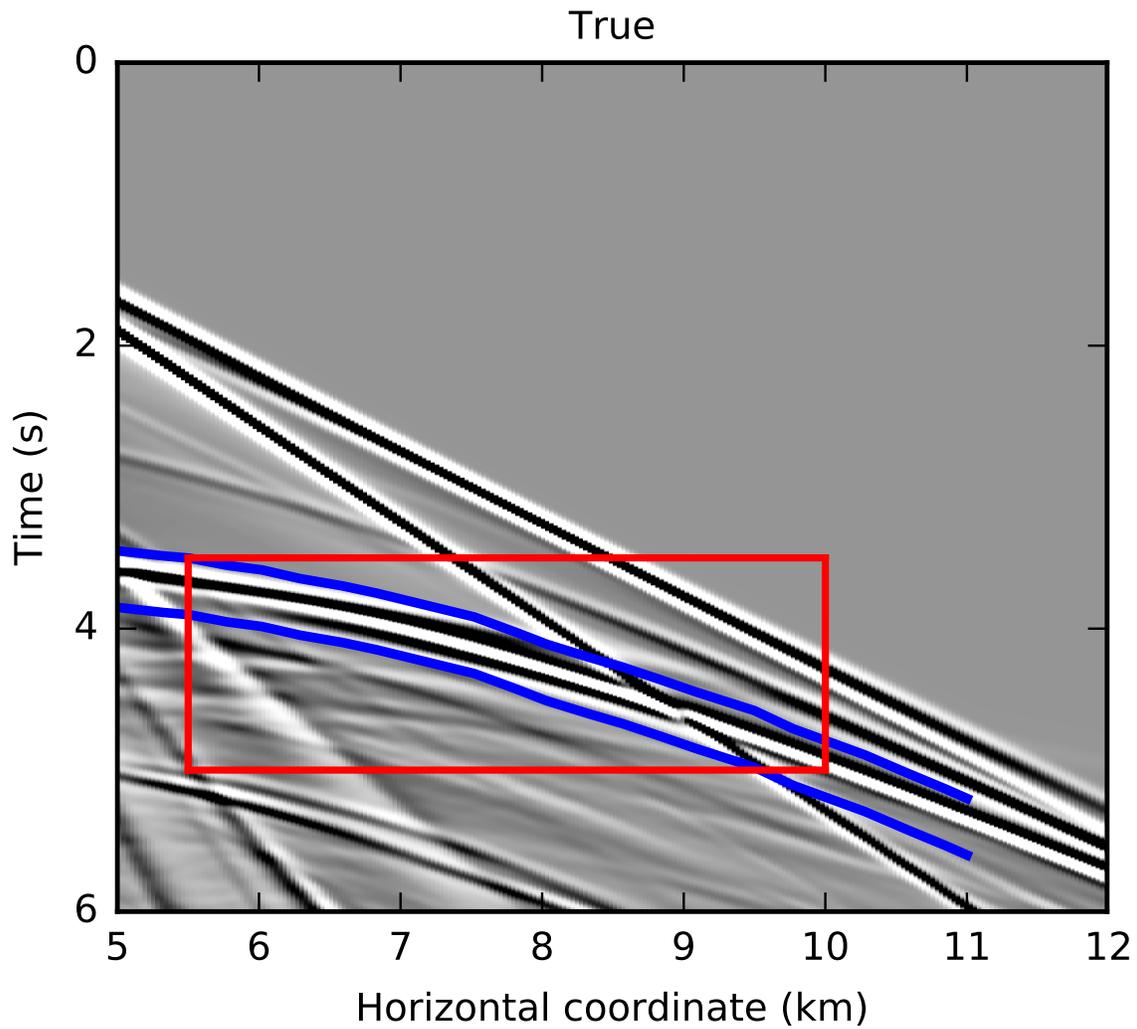


Figure 6-10: Shot gather for a source at $x = 2.5$ km with the red box indicating the zoomed region shown for the local solvers in Figure 6-11. This gather is generated on the true elastic model. The blue line represents the window later used to measure the phase of the reflection.

	Taper Sides			
Solver	Top	Bottom	Left	Right
Local A	ρ : 2100.0 kg/m ³ V_s : 0.0 m/s	ρ : 2100.0 kg/m ³ V_s : 0.0 m/s	ρ : 2100.0 kg/m ³ V_s : 0.0 m/s	ρ : 2100.0 kg/m ³ V_s : 0.0 m/s
Local B	ρ : 2100.0 kg/m ³ V_s : 1300.0 m/s	ρ : 2100.0 kg/m ³ V_s : 1300.0 m/s	ρ : 2100.0 kg/m ³ V_s : 1300.0 m/s	ρ : 2100.0 kg/m ³ V_s : 1300.0 m/s
Local C	N/A	N/A	N/A	N/A
Local D	N/A	N/A	ρ : 2200.0 kg/m ³ V_s : 1300.0 m/s	ρ : 2200.0 kg/m ³ V_s : 1300.0 m/s

Table 6.3: Table summarizing the taper properties for the four local solver configurations. Entries for ρ and V_s indicate the taper end value at the boundary. The entry 'N/A' means that the particular taper is not present in this configuration. Local A and Local B taper to the outer red boundary in Figure 6-9. Local C directly injects on the blue box with $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$. Local D only retains the horizontal tapers. Graphically this corresponds to the thin red box connected extending sideways from the blue box in Figure 6-9.

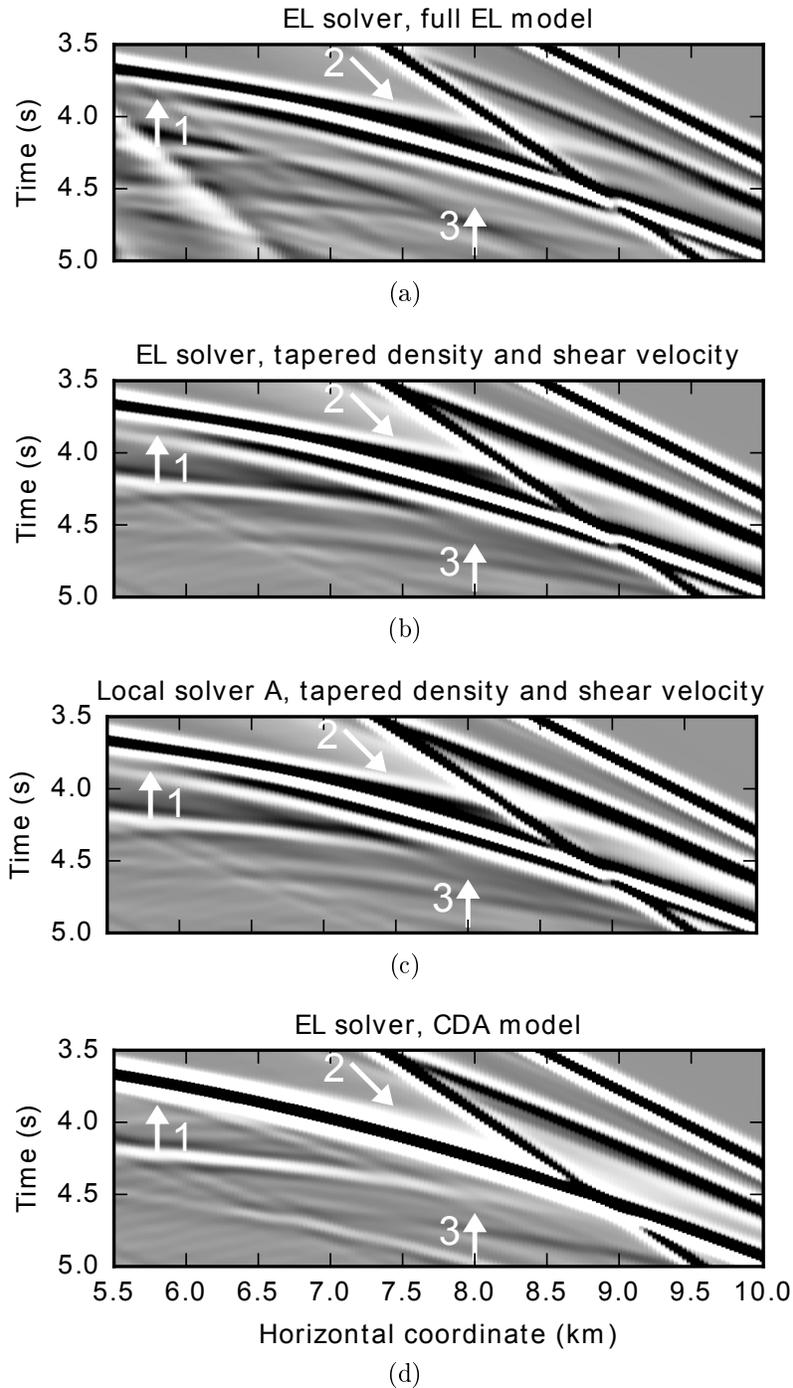


Figure 6-11: Sections of shot gathers for a source at $x = 2.5$ km in four different scenarios, within the region marked with the red box in Figure 6-10. The horizontal axis is the receiver coordinate. Three different locations of interest are marked. 1: The top of reservoir reflection. 2: The head-wave separating from this reflection, marking the transition to the post-critical regime. 3: Location where the traces are plotted in Figure 6-12. The phase has obviously rotated with respect the nearer offset reflections before the critical angle. The amplitude scale is the same in all Figures.

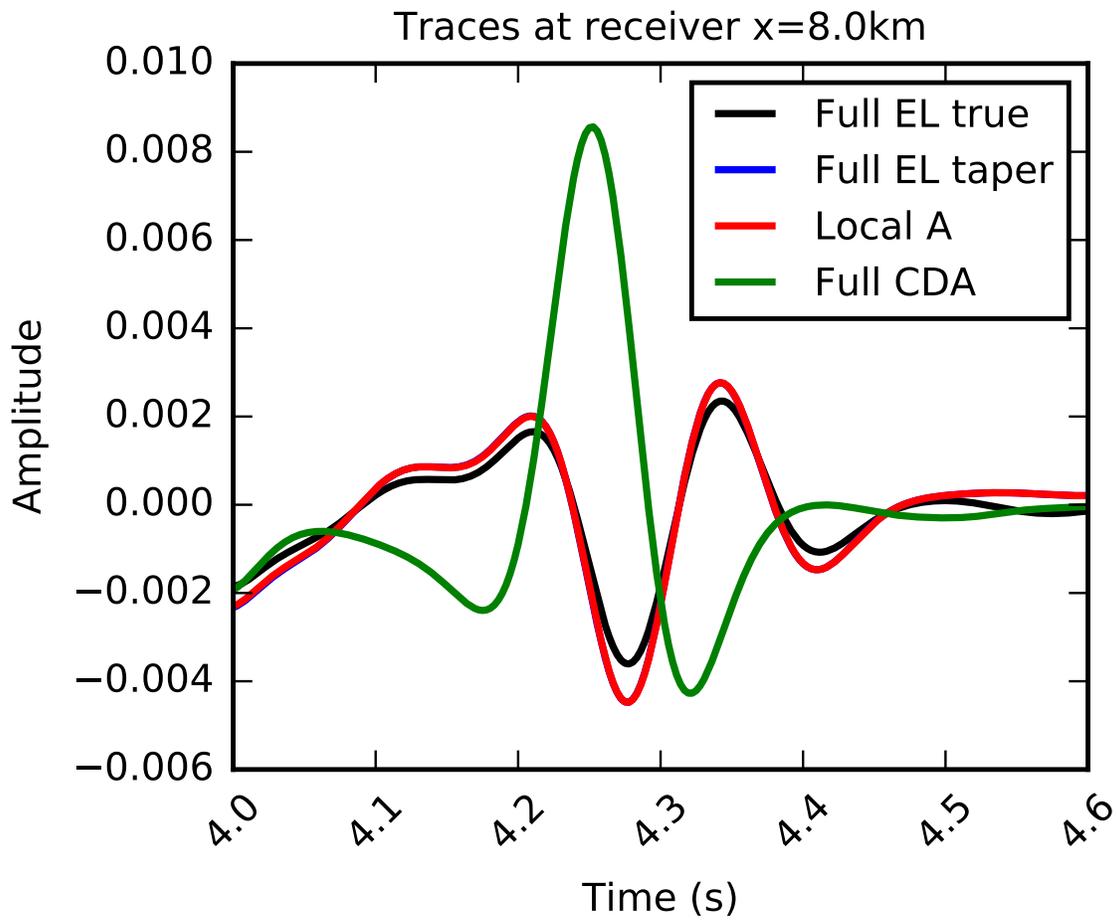


Figure 6-12: Traces corresponding to the four scenarios in Figures 6-11a - 6-11d in the same order at $x = 8.0$ km marked by the number 3 in the first panel. The red line almost perfectly matches the blue line.

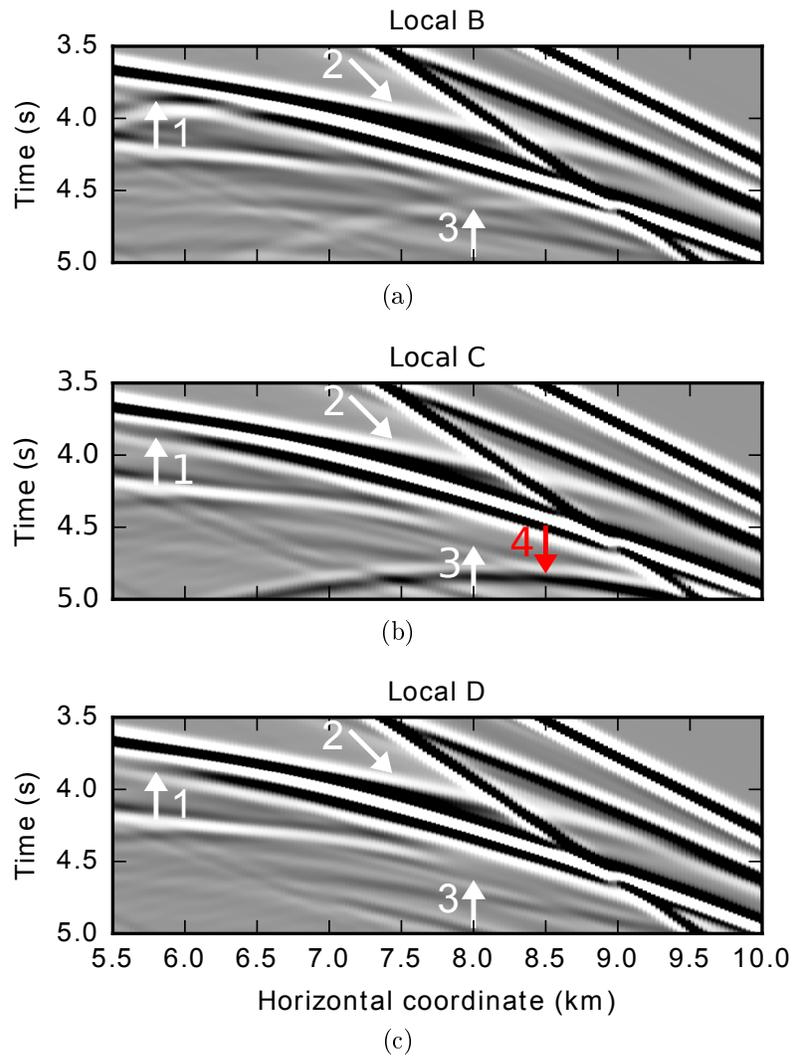


Figure 6-13: This figure plots the wavefield simulations for configurations Local B, Local C and Local D respectively. Similar to Figure 6-11 the source is placed at $x = 2.5$ km. The red arrow in Figure 6-13b points to an artificial high amplitude scatter event. The amplitude scale is the same in all Figures.

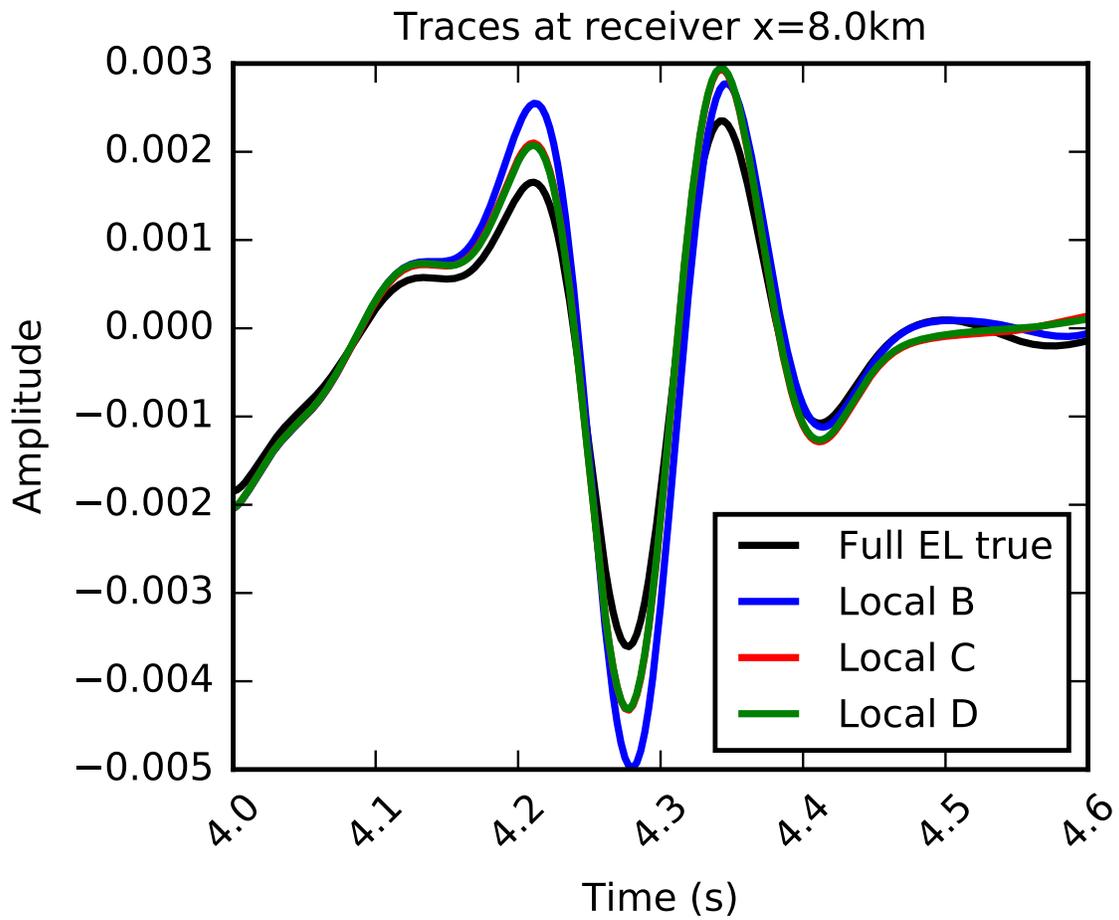


Figure 6-14: The black trace is the same true trace as in Figure 6-12. The other three traces correspond to the solver configurations in Figure 6-13. The red and green line nearly overly one another.

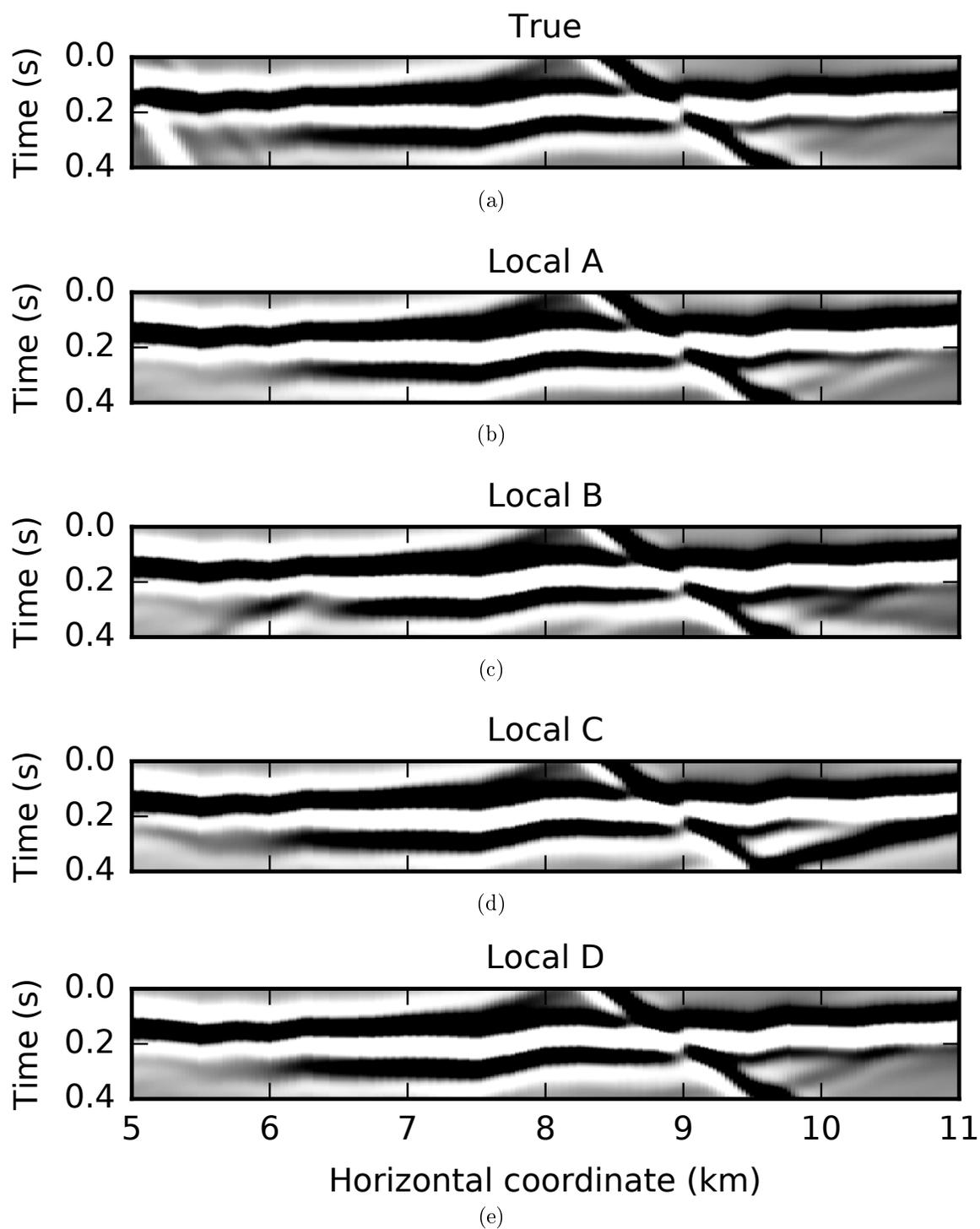


Figure 6-15: The extracted gathers within the window indicated by blue lines in Figure 6-10. Plot (a) shows the true gather and plots (b)-(e) show the four local results. The amplitude scale is the same in all Figures.

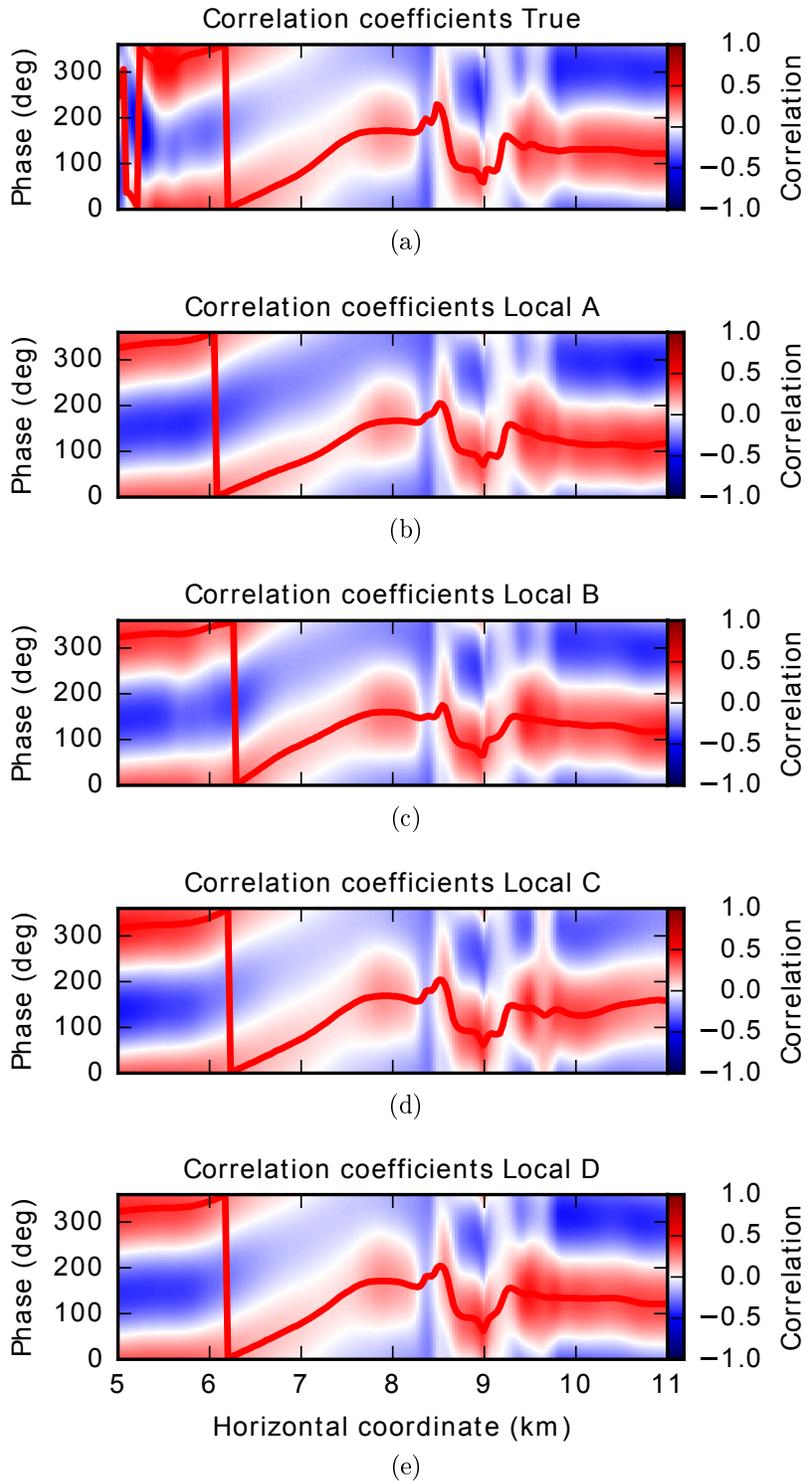
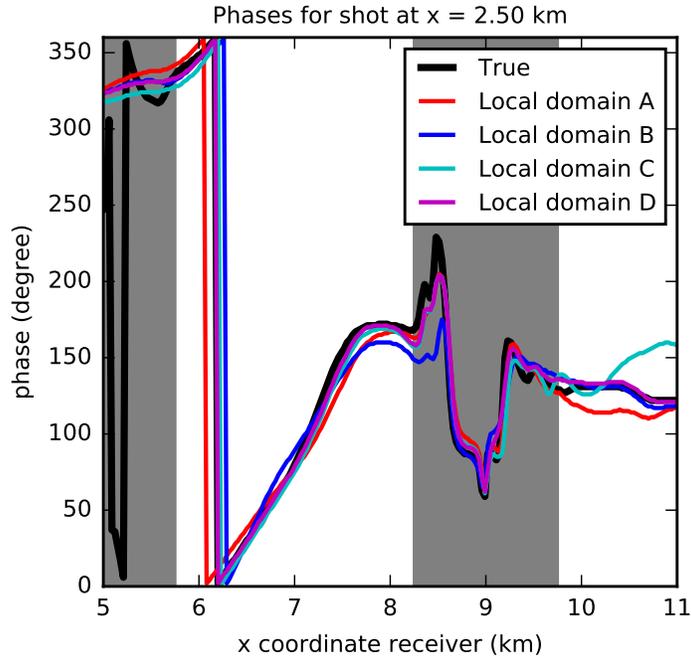
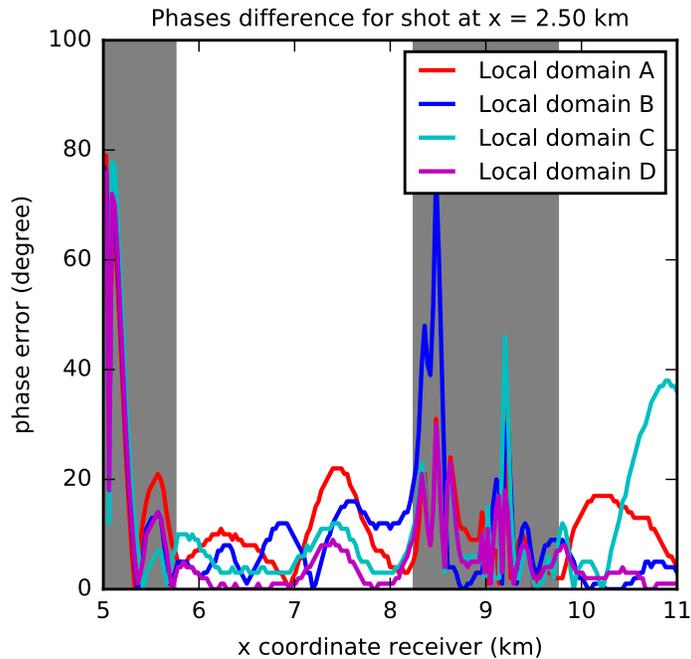


Figure 6-16: Correlation values obtained when rotating each windowed trace and then correlating it with its envelope for a source at $x = 2.5$ km. This is done at each receiver coordinate for the windowed gathers in Figure 6-15. The peak of this correlation is plotted with the red line and this is used as the phase of the reflection. At some receiver coordinates we observe overall lower correlation values .



(a)



(b)

Figure 6-17: a): Comparing the selected phases from Figure 6-16. Intersecting events cause abrupt deviations from a smooth phase transition. We therefore choose to mute the gray regions. b): Plotting the difference with the 'true' phase for all simulations in Figure 6-17a. The phase difference is corrected for phase wrapping. Average misfits in the non-muted regions: Local A: 10.07° , Local B: 6.69° , Local C: 9.84° and Local D: 2.67° .

Table of average phase error				
source x (km)	<i>A</i>	<i>B</i>	<i>C</i>	<i>D</i>
2.0	7.19°	9.41°	8.48°	6.38°
2.5	10.07°	6.69°	9.84°	2.67°
3.0	10.26°	14.65°	11.83°	4.04°
3.5	7.18°	15.78°	6.94°	3.55°
4.0	9.45°	8.47°	3.77°	3.10°
4.5	9.43°	5.03°	12.53°	4.80°
Average	8.93°	10.01°	8.90°	4.09°

Table 6.4: Average phase error (degrees) for each of the four local solver modeling cases for each of the selected sources, not including muted regions due to interacting events. Errors for each source are influenced by the picking of the window and mute ranges. The same windows and mutes for each source are used by all the local solver configurations. The average in the bottom row is just the arithmetic average of the numbers for the individual shots. This does not take into account that not every source has the same number of non-muted phase measurements.

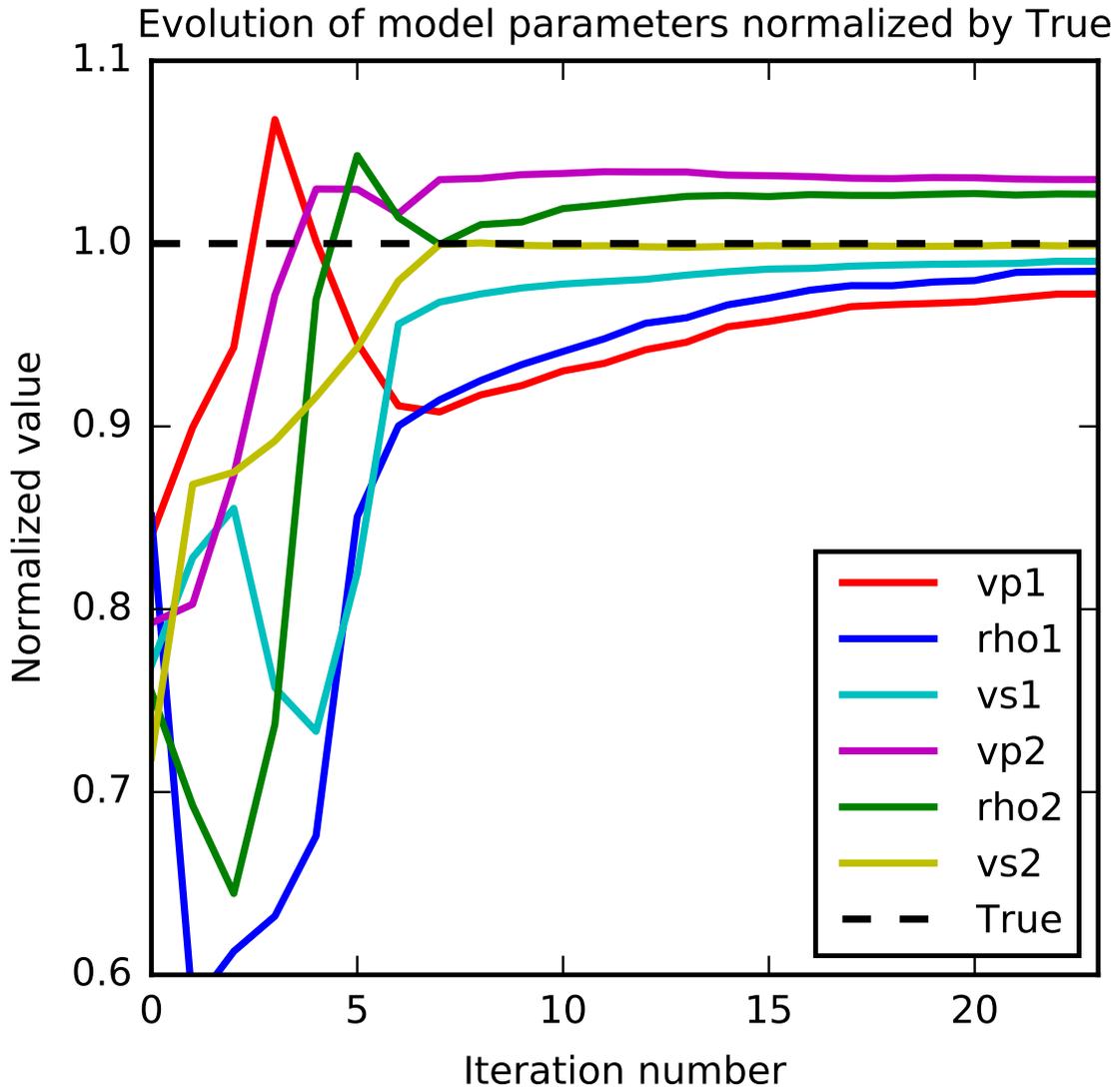


Figure 6-18: Normalized model evolution during inversion. '1' refers to the potentially hydrocarbon bearing anomaly. '2' refers to the bulk chalk.

Hydrocarbon anomaly	True	Initial		Inverted	
		Value	Error	Value	Error
V_p (m/s)	5000.0	4200.0	16.0%	4862.1	2.8%
ρ (kg/m ³)	2350.0	2000.0	14.9%	2314.4	1.5%
V_s (m/s)	2860.0	2200.0	23.1%	2832.3	1.0%
Bulk chalk	True	Initial		Inverted	
		Value	Error	Value	Error
V_p (m/s)	5300.0	4200.0	20.8%	5485.9	3.5%
ρ (kg/m ³)	2650.0	2000.0	24.5%	2721.6	2.7%
V_s (m/s)	3060.0	2200.0	28.1%	3056.4	0.1%

Table 6.5: The P-velocity, density and S-velocity for the potentially hydrocarbon bearing anomaly and the bulk chalk. The initial guess is improved significantly during this phase-only inversion.

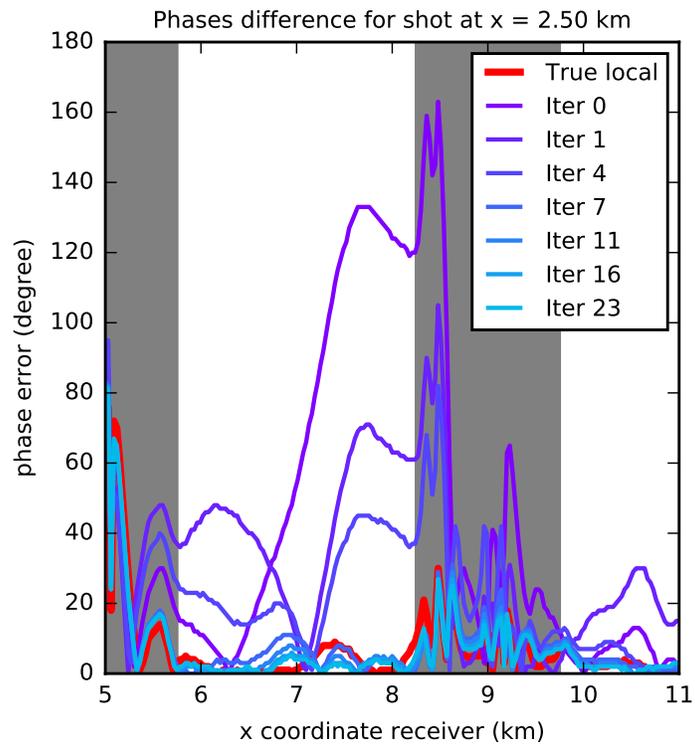
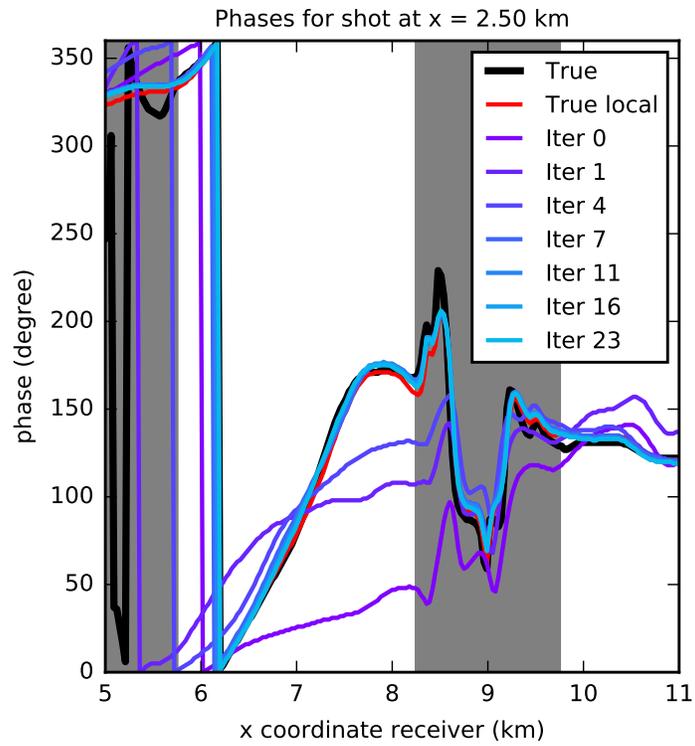


Figure 6-19: Phase for source at $x = 2.5$ km for selected inversion iterations. The grayed out region is muted and the phase does not influence the inversion. Figure a) shows phase and b) shows the difference (i.e. residual).

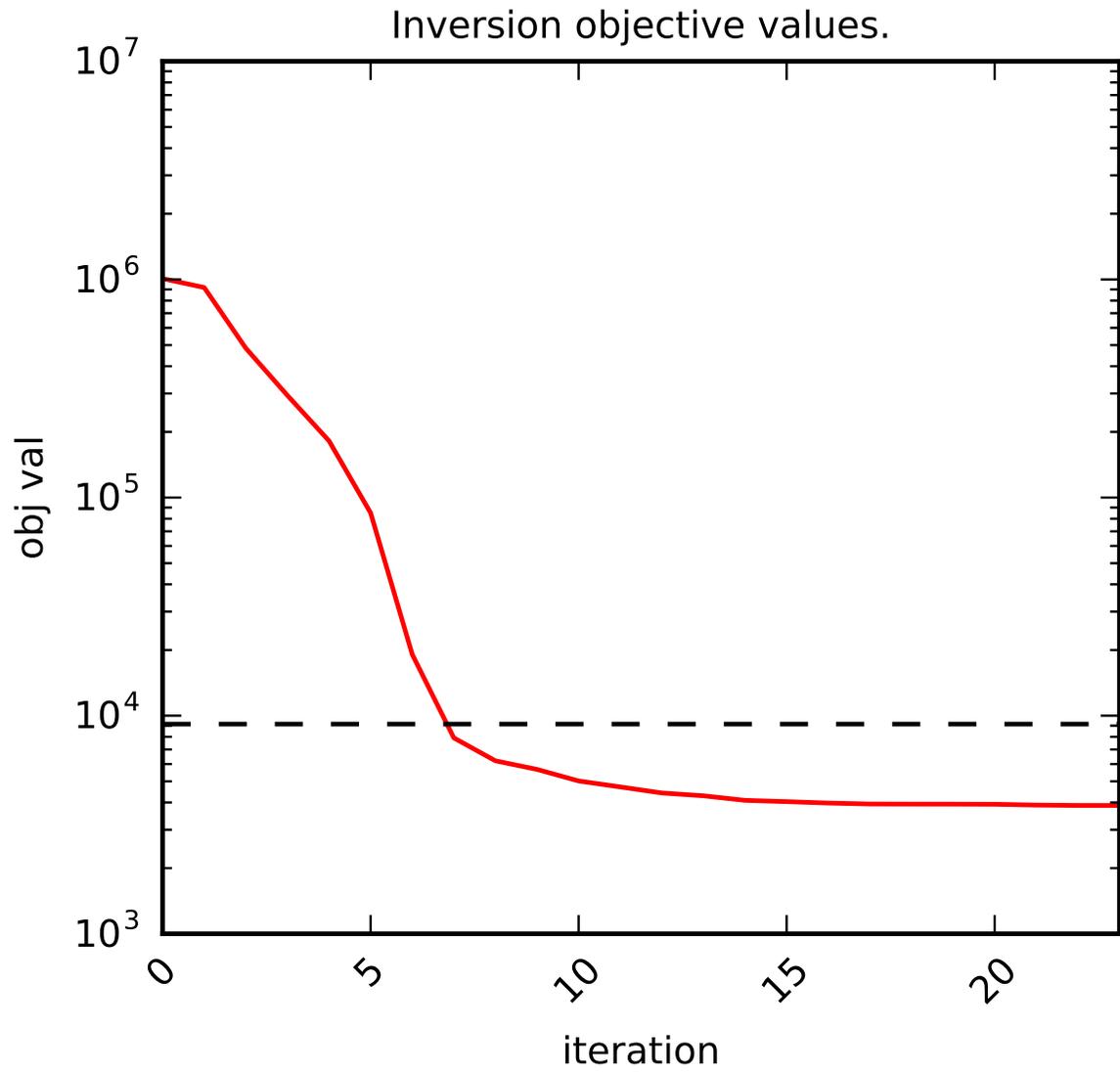


Figure 6-20: Log plot of objective value summing contributions from residual curves like Figures 6-19b for all sources to get objective value (6.16). The black dashed line is the objective value from the true model using Local D.

Chapter 7

Conclusions and Future Directions

7.1 Summary of main contributions

In this thesis we investigated multiple seismic inverse problems with a local nature. For this reason we developed the numerically exact local solver in **Chapter 2**. We obtain large computational savings by restricting the generation of wavefields to the small region of interest. Since both the forward and adjoint wavefields are computed without error, the local gradient is numerically exact. This exact local gradient is then available for a local optimization routine of choice, such as for instance nonlinear CG or L-BFGS. Before the local solver can be used, a number of full domain Green's functions are computed in the offline stage. After these full domain Green's functions are computed, no further full domain simulations are needed. The local solver significantly accelerates the inversion procedure, compared to the conventional approach of using a full domain solver. This reduces the computational time of an inversion, making the procedure more interactive.

In **Chapter 3** we demonstrate the functionality of the local solver by applying it to two local inverse problems. We first use it to update the boundary of a salt body, using FWI and a level set parametrization of the salt boundary. Accurate estimation of salt shape in the velocity model is very important for subsalt imaging. The local solver is efficient for this task because the salt body occupies only a subset of the full computational model. We demonstrated how the local solver updated the salt bound-

ary and how this resulted in an improved subsalt image. The second local problem we investigated was time-lapse inversion. In the absence of significant geomechanical effects, we expect the time-lapse changes to be primarily in the neighborhood of the reservoir. We use a multi-domain local solver and DDFWI to efficiently estimate the time-lapse change in multiple reservoirs. In addition to the faster forward model, we also observe a reduction in the number of required FWI iterations when restricting the model updates to the reservoir neighborhoods. In both local inversion examples we demonstrate that the local solver efficiently updates the velocity model.

We then propose a new time-lapse algorithm in **Chapter 4**. Similar to the AFWI algorithm introduced by Yang (2014), we also compute a confidence map in a pre-processing step. In this step we invert for both the baseline and monitor data using a single model. Since this model has to fit both baseline and monitor data, we suspect that, at least for small time-lapse perturbations, this model has values between those of the true baseline and monitor models. While updating the intermediate model we record the baseline and monitor gradients at each pixel. At regions of true time-lapse change, we suspect that the baseline and monitor gradients have relatively large amplitude and want to update the intermediate model in opposite directions. We therefore interpret regions where the baseline and monitor gradients have opposite sign and large amplitude to be more likely to contain time-lapse change. At each pixel we therefore compute a heuristic confidence value based on the gradient history. The resulting confidence map is used to regularize a subsequent joint inversion. We demonstrate that this confidence map is good at limiting the effect of noise on the time-lapse estimate.

At this point there are not many published examples of time-lapse FWI applied to field data. The few published studies make the acoustic approximation when inverting for this field data. It is not well understood what the impact of the acoustic approximation is on the time-lapse estimate. In **Chapter 5** we investigate this question on a realistic elastic synthetic North Sea model. We demonstrate that the phase-shifts of post-critical reflections in the elastic data negatively impact the acoustic time-lapse estimate. When removing these post-critical reflections from the elastic data, the

acoustic time-lapse estimate improves. In addition we demonstrate the importance of the initial model. Finally we include the time-lapse regularization algorithm from Chapter 4.

In **Chapter 5** we decided to remove the post-critical PP reflections because their phase shifts cannot be modeled accurately using an acoustic forward model. These phase shifts depend on the elastic material contrast at the reflector and therefore contain useful information to constrain an inversion. Instead of removing these post-critical reflections, we introduce in **Chapter 6** a new algorithm for modeling them efficiently. We make use of the observation that the reflector of interest usually occupies a subset of the full computational domain. We therefore introduce an efficient elastic local solver to model local PP reflections and their phase shifts. Zhu and McMechan (2012) demonstrate that phase of a reflection is largely unaffected by propagation through the overburden. Instead of using expensive elastic Green's functions we therefore use cheap constant density acoustic Green's functions. Both types propagate the PP reflections through the overburden with only limited impact on the phase. We demonstrate that the coupled acoustic-elastic local solver accurately computes PP reflections on the reflector of interest. We then use the phase of these reflections to invert for material properties.

7.2 Directions for Future Work

The local solver methods we introduce in this paper can be improved in many ways. In order to make the exact local solver of **Chapter 2** applicable to field studies several things need to happen. First of all, the current implementation is second order accurate in space. This low order of accuracy will introduce numerical dispersion effects to the solutions. Extending the approach to higher orders of accuracy is therefore very important. We suspect that numerical exactness can be preserved for higher orders of accuracy as well, but the derivations may be more complicated. One avenue of efficiently extending the local solver to arbitrary orders of accuracy is by better understanding a possible connection with the Schur complement. The Schur

complement provides a general procedure for computing a subset of a solution vector when solving a system of equations. For the Helmholtz problem, which describes wave propagation in the frequency domain, this corresponds to computing the exact local wavefield in potentially multiple local domains. In order to obtain these exact local wavefields, the Schur complement requires a partial inverse of the full domain Helmholtz matrix. The columns of this partial inverse matrix can be expressed in terms of Green’s functions, which were also the central component of the local solver approach we introduced in Chapter 2. The wavefields can naturally be evaluated at the receivers by including their locations in the choice of the local domain. This wavefield at the receivers is required for objective function evaluation, and to turn the receivers into adjoint sources. Since the Schur complement is a general tool, it would naturally describe how to implement a numerically exact local solver for any order of spatial discretization. The exact local solver we introduce is also only valid for constant density acoustic physics. Extending the approach to include more realistic physics is important. Extensions to more complicated physics may also be derived more naturally using the Schur complement. Finally, direct application of the idea to 3D problems introduces unfavorable scaling in the number of required full domain Green’s function computations and associated the storage cost. In **Chapter 2** we quantify this scaling and propose an approach using matrix probing that can potentially alleviate some of these issues. We believe that a computationally efficient extension of the solver to 3D would be a great step forward.

The confidence map we introduce in **Chapter 4** is based on a fundamental assumption. We intuitively suspect that in the preprocessing step the baseline and monitor gradients try to perturb the single model in opposite directions. In the example we show this assumption seems to result in a confidence map which is aligned with the true time-lapse change. We did not mathematically quantify the conditions under which we can expect this behavior. We suspect that there is a threshold for the amplitude and extent of the model perturbations for which the gradient based confidence map will give desirable results. For instance, when the time-lapse change becomes too extensive, the simulated data on a single ‘intermediate’ model will be-

come too far from both (cycle skipping) the true baseline and true monitor data. The gradient based confidence map for such significant time-lapse change may therefore not be accurate. More study is needed to quantify the limitations of the gradient based confidence map.

In **Chapter 6** we make use of amplitude compensation factors to inject a constant density acoustic wavefield directly on a varying elastic local model. The inverse of this procedure is used in the boundary integral. In this step we use amplitude compensation factors to revert the pressure on the varying elastic local model back to constant density acoustic pressure. Both these dynamic rescaling processes are inherently heuristic. By trying different solver configurations and investigating the boundary integrand in Appendix F we build some understanding for when this procedure works. A more rigorous treatment of the error that is introduced in these heuristic steps would help quantify under which conditions we expect the method perform well. In addition, there may be alternative scaling approaches we have not investigated yet. If finite frequency effects are limited in the overburden, it is interesting to investigate whether the local solver can be coupled to the acquisition using a ray-based approach, instead of constant density acoustic Green's functions. We expect the computational savings from such a modification to be large. Not only would this reduce the offline computational burden of the full domain simulations, but the storage cost would also be reduced significantly. Finally it is interesting to use the phase measurements of the post-critical reflections in the time-lapse inverse problem of **Chapter 5**.

In this thesis we demonstrate the potential of a local solver for speeding up the inversion of the salt boundary, time-lapse model change and PVA/AVA. The main benefit of this speedup is that it makes local workflows more interactive by reducing the computational time between start and finish. This has implications for any inverse problem where the cost of the forward model is currently the limiting factor. In addition to the problems we describe in this thesis, we see multiple other avenues where an efficient local solver can transform current workflows. First, the significant reduction in computational cost of the forward model opens up the opportunity for

more continuous time-lapse inversion. This is especially interesting in those locations with permanent monitoring systems, where new data vintages can be acquired at low cost, either by reshooting with just a source vessel or potentially even by cross-correlating noise traces (de Ridder et al., 2014). With the more efficient local forward model it may be feasible to invert for more than two data vintages simultaneously, which opens up new ways of regularizing reservoir change as function of time. Second, local solvers may find applications outside of oil and gas, including for instance the monitoring of structures, environmental engineering, medical imaging and non-destructive testing. An example of an environmental engineering application could be the monitoring of pollutants, since their source is sometimes localized. Depending on the particular pollutant, non-seismic detection methods may be more efficient. For instance, materials with different resistivity than the surrounding soil may be easier to detect using resistivity methods and would therefore warrant development of local electric modeling methods. A critical component for the adoption of local solvers is that they must scale efficiently in 3D. For the local solver of Chapter 2 we have for instance suggested potential pathways to reduce the storage cost and the number of required full-domain Green's functions. Third, if the computational cost of these steps is brought down significantly, it would potentially open up a new approach in which the full computational domain is partitioned (potentially hierarchically) into a continuous collection of adjacent local domains. The wavefield can be computed in any of these local domains or even in a collection of these local domains by using the multi-domain version of the local solver. The model can then be efficiently updated in this collection of local domains. The scattered field from this model update can then potentially be used to update the background Green's functions for the other local domains which were not involved. This gives the user the freedom to change the selection of local domains after any model update. A big advantage of breaking up the full domain in a collection of small domains is that at any moment the user can decide to focus on any subset of the Earth model and speed up the forward model accordingly. The user no longer needs to fix beforehand the boundaries of the local domains, unlike the more rigid approach in for instance chapter 2.

Appendix A

Derivation of numerically matching scattered field

Model perturbations δm act as scattering sources in the Lippmann-Schwinger equation (Keys and Weglein, 1983) for the scattered field u_s . Equation 2.2 shows that the model perturbations δm are limited to domain C . Therefore, outside of C , the model is the background model and the scattered wavefield is sourceless:

$$-m_0(x)\omega^2 u_s(x, \omega) - \Delta_h u_s(x, \omega) = 0, \quad x \in \Omega \setminus C. \quad (\text{A.1})$$

Recall that the model is unperturbed on the boundary ∂C of the truncated domain. It is also known that the numerical Green's function in the background model m_0 satisfies

$$-m_0(x)\omega^2 G_0(x, y, \omega) - \Delta_h G_0(x, y, \omega) = \delta(x - y), \quad x, y \in \Omega. \quad (\text{A.2})$$

Multiplying Equation (A.1) by $G_0(x, y, \omega)$ and subtracting Equation (A.2) multiplied by $u_s(x, \omega)$ gives

$$\begin{aligned} -G_0(x, y, \omega)\Delta_h u_s(x, \omega) + u_s(x, \omega)\Delta_h G_0(x, y, \omega) = \\ -u_s(x, \omega)\delta(x - y), \quad x \in \Omega \setminus C, y \in \Omega. \end{aligned} \quad (\text{A.3})$$

Both sides of Equation A.3 will be summed over the nodes in $B \cup \partial C$ to get a convenient expression for the scattered field u_s . Summation by parts is used to turn Equation (A.3) into an expression resembling a boundary integral.

To make the rest of the derivation easier to follow, a 1D version of Equation (A.3) is first summed along a line. The Laplacian operator Δ_h becomes the three-point second-order derivative stencil in 1D. The left-hand side of equation Equation A.3 centered around node i is $d^{(i)}$ and in 1D becomes:

$$d^{(i)} = -G_0^{(i)} \frac{u_s^{(i+1)} - 2u_s^{(i)} + u_s^{(i-1)}}{h^2} + u_s^{(i)} \frac{G_0^{(i+1)} - 2G_0^{(i)} + G_0^{(i-1)}}{h^2}, \quad (\text{A.4})$$

where for notational convenience the dependence on x, y , and ω has temporarily been dropped. The detour through the 1D case will be useful when finding an expression for the 2D summation of Equation (A.3) with the five-point Laplacian stencil. Figure A-1 shows the 1D setup of the problem. This figure corresponds to the dotted black box in Figure 2-1. Equation A.4 is first written in a more convenient way:

$$d^{(i)} = -G_0^{(i)} (a^{(i+1)} - a^{(i)}) + u_s^{(i)} (b^{(i+1)} - b^{(i)}), \quad (\text{A.5})$$

with the coefficients $a^{(i)}$ and $b^{(i)}$ defined as:

$$a^{(i)} = \frac{u_s^{(i)} - u_s^{(i-1)}}{h^2}, \quad b^{(i)} = \frac{G_0^{(i)} - G_0^{(i-1)}}{h^2} \quad (\text{A.6})$$

Using summation by parts to sum $d^{(i)}$ over its domain of definition (i.e., the red and black nodes in the 1D version of $B \cup \partial C$ in Figure A-1), the following expression is obtained:

$$\begin{aligned}
\sum_{i=m}^n d^{(i)} &= - \left[G_0^{(n+1)} a^{(n+1)} - G_0^{(m)} a^{(m)} \right] \\
&\quad + \sum_{i=m}^n a^{(i+1)} \left(G_0^{(i+1)} - G_0^{(i)} \right) \\
&\quad + \left[u_s^{(n+1)} b^{(n+1)} - u_s^{(m)} b^{(m)} \right] \\
&\quad - \sum_{i=m}^n b^{(i+1)} \left(u_s^{(i+1)} - u_s^{(i)} \right) \tag{A.7}
\end{aligned}$$

If we recognize that $\left(G_0^{(i+1)} - G_0^{(i)} \right) = h^2 b^{(i+1)}$ and $\left(u_s^{(i+1)} - u_s^{(i)} \right) = h^2 a^{(i+1)}$, the sums cancel. With this simplification Equation (A.7) becomes:

$$\begin{aligned}
\sum_{i=m}^n d^{(i)} &= - \left[G_0^{(n+1)} a^{(n+1)} - G_0^{(m)} a^{(m)} \right] \\
&\quad + \left[u_s^{(n+1)} b^{(n+1)} - u_s^{(m)} b^{(m)} \right] \\
&= \frac{1}{h^2} \left[u_s^{(n)} G_0^{(n+1)} - G_0^{(n)} u_s^{(n+1)} \right] + \\
&\quad \frac{1}{h^2} \left[u_s^{(m)} G_0^{(m-1)} - G_0^{(m)} u_s^{(m-1)} \right] \tag{A.8}
\end{aligned}$$

Equation A.8 can be manipulated to show that $\sum_{i=m}^n d^{(i)}$ can be rewritten in terms of weighted normal derivatives on both boundaries:

$$\begin{aligned}
\sum_{i=m}^n d^{(i)} &= \frac{1}{h^2} \left[u_s^{(n)} \left(G_0^{(n+1)} - G_0^{(n)} \right) - G_0^{(n)} \left(u_s^{(n+1)} - u_s^{(n)} \right) \right] \\
&\quad + \frac{1}{h^2} \left[u_s^{(m)} \left(G_0^{(m-1)} - G_0^{(m)} \right) - G_0^{(m)} \left(u_s^{(m-1)} - u_s^{(m)} \right) \right] \tag{A.9}
\end{aligned}$$

We now use Equation (A.8) in the following derivation for the 2D summation of Equation (A.3). The left hand side of Equation (A.3) is called $d^{(i,j)}$ with index i

referring to the grid row and index j referring to the grid column.

$$\begin{aligned}
d^{(i,j)} = & \\
& \left(-G_0^{(i,j)} \frac{u_s^{(i+1,j)} - 2u_s^{(i,j)} + u_s^{(i-1,j)}}{h^2} \right. \\
& \left. + u_s^{(i,j)} \frac{G_0^{(i+1,j)} - 2G_0^{(i,j)} + G_0^{(i-1,j)}}{h^2} \right) \\
& + \left(-G_0^{(i,j)} \frac{u_s^{(i,j+1)} - 2u_s^{(i,j)} + u_s^{(i,j-1)}}{h^2} \right. \\
& \left. + u_s^{(i,j)} \frac{G_0^{(i,j+1)} - 2G_0^{(i,j)} + G_0^{(i,j-1)}}{h^2} \right) \tag{A.10}
\end{aligned}$$

The elements of the five-point Laplacian stencil have been ordered in such a way that the first line of Equation (A.10) contains all the vertical derivatives and the second line contains all the horizontal derivatives. It is important to notice that both lines resemble Equation (A.4). When finding a simplified expression for the 2D summation of Equation (A.10) over $B \cup \partial C$, it is useful to investigate the summation over both lines of Equation (A.10) individually.

First we will focus on the 2D summation of the first line of Equation (A.10) over the domain $B \cup \partial C$ (i.e., the black and red nodes). We compute this 2D summation within a column and then add all of these column sums. The sum over all the elements within a column is expressed in terms of nodal contributions at the top and bottom boundary of the mesh in Figure 2-1. This is visualized in the left blue box in Figure A-2 (a) and follows from the 1D derivation that the first line of (A.10) has the form of (A.4). Summation over columns crossing C is handled using the same logic from the 1D derivation. The column sum is split into two parts, which are represented by the two other blue boxes in Figure A-2 (a). The first part is summation over the top subcolumn extending from the top of the full mesh to the node on the top boundary of the truncated mesh. The second part is summation over the bottom subcolumn extending from the bottom node on the truncated mesh boundary to the bottom of the full mesh. Summation over both subcolumns is also expressed in terms

of weighted normal derivatives at their boundaries. Adding all these weighted normal derivatives, indicated by arrows in Figure A-2 (a), equals a sum over all grid points of the first line of Equation (A.10).

The same idea can be repeated with the second line of Equation (A.10). But now summation is over all the rows. Rows intersecting C are split into two subrows whose contributions are added. The summation over a row or subrow is again expressed in terms of weighted normal derivatives as in the 1D version Equation (A.8). Therefore, summation of the second line of Equation (A.10) over $B \cup \partial C$ results in horizontal normal derivatives that are added to the vertical normal derivatives of Figure A-2 (a). The summation of Equation (A.10) over $B \cup \partial C$ is the combination of both lines of the equation. Therefore, it includes the horizontal and vertical normal derivatives on the boundaries, as is illustrated in Figure A-2 (b). One important observation is that the corner nodes of ∂C do not contribute.

Summing the right hand side of Equation (A.3) over the volume $B \cup \partial C$ gives $-u^s(y, \omega)$ for y in $B \cup \partial C$. Therefore, the summation of all the weighted normal derivatives in Figure A-2 (b) gives the scattered field:

$$\sum \frac{1}{h^2} \left(u_s^{(n)} \left(G_0^{(n+1)} - G_0^{(n)} \right) - G_0^{(n)} \left(u_s^{(n+1)} - u_s^{(n)} \right) \right) = -u_s(y, \omega), \quad y \in B \cup \partial C, \quad (\text{A.11})$$

where the summation is over all non corner boundary nodes and the superscript n is the value on the boundary, and the subscript $n + 1$ one node to the exterior from B . One problem with this boundary summation is that it requires explicit knowledge of the scattered wavefield. There is an analogous derivation for the background field u_0 , which is generated by a source in B . The left-hand side turns into a weighted normal derivative summation with u_0 replacing u_s , and the right-hand side is zero. Adding

that expression to Equation (A.11) and using $u = u_0 + u_s$ gives:

$$\sum \frac{1}{h^2} \left(u^{(n)} \left(G_0^{(n+1)} - G_0^{(n)} \right) - G_0^{(n)} \left(u^{(n+1)} - u^{(n)} \right) \right) = -u_s(y, \omega), \quad y \in B \cup \partial C, \quad (\text{A.12})$$

where the scattered field in $B \cup \partial C$ is now expressed in terms of the total field u at the boundaries. A more detailed investigation would show that the weighted normal derivatives on the outer boundary sum to zero. Therefore, the summation in Equation (A.12) can be limited to the boundary of the truncated domain, where the reader is reminded that the corner nodes are not involved as can be seen in Figure A-2 (b). The derivation in this appendix can easily be extended to 3D.

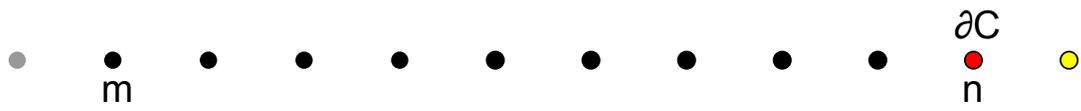


Figure A-1: Zoomed in section of the black dotted box in Figure 2-1.

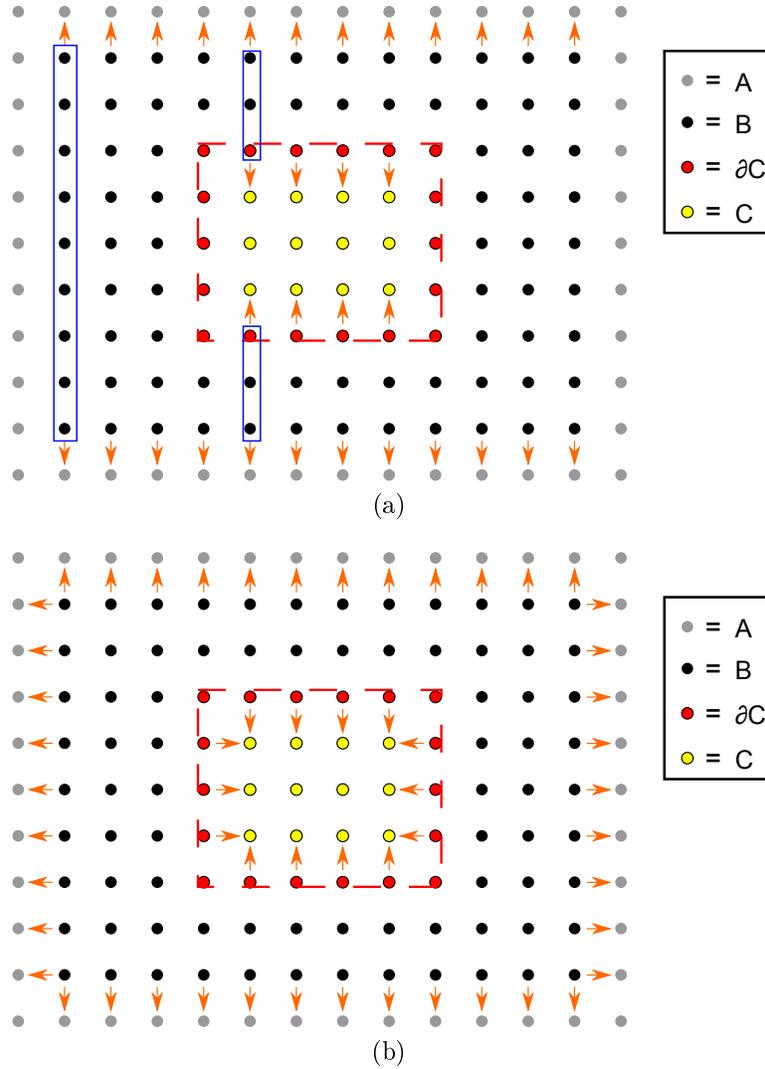


Figure A-2: **(a)** The summation of the first line in (A.10) results in weighted normal derivatives indicated by orange arrows. The left blue box represents a column not intersecting C . Summation over this column results in weighted normal derivatives at the top and bottom of the full mesh. The two smaller blue boxes represent subcolumns whose summation also result in weighted normal derivatives at the endpoints. **(b)** Additional summation over the second line adds the horizontal normal derivatives to the vertical ones of **a**. The contribution on the outer boundary adds up to zero and can therefore be neglected.

Appendix B

Numerically exact multi domain solver

It is possible to solve for the identical wavefield a full domain solver would have generated in an arbitrary number of disjoint truncated domains. The derivation of this multi-domain local solver is analogous to the single truncated domain case. Instead of having a single truncated domain interior C with boundary ∂C , there is now a collection of truncated domain interiors C_1, C_2, \dots, C_N with boundaries $\partial C_1, \partial C_2, \dots, \partial C_N$. The difference in the wavefield due to model perturbations inside the truncated domain interiors is again called u_s . Performing a derivation entirely analogous to the one in Appendix A, the scattered field is expressed using Equation (2.7). In this case, the summation takes place along the boundaries of all the truncated domains, again excluding their corner nodes. This means that Equation 2.7 uses the total wavefield both at the boundaries of the truncated domains and one layer to the interior, and the Green's functions from those nodes to any node not in the interior of the truncated domains to get the scattered field at that location. With this expression a system of equations similar to Equation (2.10) is derived to truncate the computational domain.

As before, a spiraling node numbering scheme is used with node numbering starting in the top left and increasing counterclockwise. The vector $\mathbf{u}_s^{\partial C_j}$ refers to the scattered field on the boundary nodes ∂C_j . The wavefield on the boundary nodes ∂C_j in the background model m_0 is $\mathbf{u}_0^{\partial C_j}$. The total field at the same boundary is

$\mathbf{u}^{\partial C_j}$. The total field k layers to the interior is $\mathbf{u}^{\partial C_{j+k}}$. Just like in the single domain case, a system of equations is assembled.

$$A\mathbf{x} = \mathbf{b} \quad (\text{B.1})$$

with matrix A defined in (B.2)

$$A = \left(\begin{array}{ccc|ccc|cccc}
-I & 0 & 0 & I & 0 & 0 & 0 & 0 & 0 & 0 & \dots \\
0 & \ddots & 0 & 0 & \ddots & 0 & 0 & 0 & 0 & 0 & \dots \\
0 & 0 & -I & 0 & 0 & I & 0 & 0 & 0 & 0 & \dots \\
\hline
I & 0 & 0 & G_{0,\rightarrow\partial C_1}^{\partial C_{1,+1}} & \dots & G_{0,\rightarrow\partial C_1}^{\partial C_{N,+1}} & -G_{0,\rightarrow\partial C_1}^{\partial C_1} & \dots & G_{0,\rightarrow\partial C_1}^{\partial C_N} & 0 & \dots \\
0 & \ddots & 0 & \vdots & & \vdots & \vdots & & \vdots & 0 & \dots \\
0 & 0 & I & G_{0,\rightarrow\partial C_N}^{\partial C_{1,+1}} & \dots & G_{0,\rightarrow\partial C_N}^{\partial C_{N,+1}} & -G_{0,\rightarrow\partial C_N}^{\partial C_1} & \dots & G_{0,\rightarrow\partial C_N}^{\partial C_N} & 0 & \dots \\
\hline
0 & & & & & -\mathbf{m}(x)\omega^2 & & & -\Delta_h & &
\end{array} \right) \quad (\text{B.2})$$

and

$$\mathbf{x} = \begin{pmatrix} \mathbf{u}_s^{\partial C_1} \\ \vdots \\ \mathbf{u}_s^{\partial C_N} \\ \hline \mathbf{u}^{\partial C_1} \\ \vdots \\ \mathbf{u}^{\partial C_N} \\ \hline \mathbf{u}^{\partial C_{1,+1}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+1}} \\ \hline \mathbf{u}^{\partial C_{1,+2}} \\ \vdots \\ \mathbf{u}^{\partial C_{1,+M_1}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+2}} \\ \vdots \\ \mathbf{u}^{\partial C_{N,+M_N}} \end{pmatrix} \quad \mathbf{b} = \begin{pmatrix} \mathbf{u}_0^{\partial C_1} \\ \vdots \\ \mathbf{u}_0^{\partial C_N} \\ \hline 0 \\ \vdots \\ 0 \\ \hline 0 \\ \vdots \\ 0 \end{pmatrix}, \quad (\text{B.3})$$

where the number of layers in truncated domain j is M_j . The Green's block $G_{0,\rightarrow\partial C_j}^{\partial C_i}$ contains the Green's functions from the boundary ∂C_i to the boundary ∂C_j where the scattered field is computed. Similarly the block $G_{0,\rightarrow\partial C_j}^{\partial C_{i,+1}}$ contains Green's functions between nodes one layer to the interior of ∂C_i to the nodes on the boundary of ∂C_j . The 0 subscript emphasizes that the Green's functions correspond to the background velocity state.

The multi-domain matrix in Equation (B.2) has a similar structure as the single domain case of Equation 2.10. In fact, it reduces to Equation (2.10) when N is set to one. The multi-domain analogy of the relations in List 1 is used when constructing Equation (B.2). Horizontal lines in Equation (B.2) and in vector \mathbf{b} in Equation (B.3) divide the matrix in three sections which are not drawn to scale. These sections are not related to the sections in vector x in Equation (B.3). The first section expresses that the difference between the perturbed and the background wavefields is the scat-

tered wavefield on the boundary of every truncated domain. The second section expresses that the scattered field on boundary nodes of any domain is expressed in terms of Equation (2.7), with summation now following the boundary of all domains as explained at the start of this appendix. The third section expresses that in the interior of every domain the regular Helmholtz equation holds.

Solving Equation (B.1) for arbitrary model perturbations in C_1, C_2, \dots, C_N gives in every truncated domain the identical local wavefields a full domain solver gives, with relative differences defined by Equation (2.11) in the order of $1e^{-12}$ to $1e^{-16}$ regardless of the size of the perturbations. Because this local solution is numerically equal to the full domain solution, it means that all interactions between the truncated domains are included, in addition to the interactions with the unperturbed background model. The cost of solving Equation (B.1) is similar to the cost of the single domain case of Equation (2.10) as long as the total number of boundary nodes in the multi-domain case is similar to the number of boundary nodes of the single domain. The single domain case will be slightly slower, because for an equal number of boundary nodes a single domain has more total nodes. Qualitatively, the matrices in equations (B.2) and 2.10 are otherwise the same.

After the local wavefields are obtained through Equation (B.1), the scattered wavefield is propagated to the receiver locations through the multi-domain version of Equation (2.7). The residuals are then obtained and the adjoint wavefields are computed locally. Cross-correlation with the local wavefields gives the local gradients, which exactly match the gradients obtained from a full domain solver. This makes it possible to do simultaneous inversion in multiple disjointed domains.

Appendix C

Parallel FWI vs Joint FWI

In Parallel FWI we invert separately for the baseline and the monitor data by minimizing (5.3) and (5.4) independently. In Joint FWI we minimize the sum of both the baseline and monitor objective functions as is shown in (5.10). After minimizing we subtract the baseline model from the monitor model to obtain our time-lapse estimate.

Both Parallel FWI and Joint FWI minimize the same two objective functions and have the same global optimal models m_0 and m_1 . The only difference between these techniques is that you either minimize the objective functions independently or at the same time. Taking the derivative of the Joint FWI objective function (5.10) with respect to the baseline model shows that this gradient does not depend on the monitor data. The same is true the other way around when taking a derivative with respect to the monitor model. This means that in Joint FWI there is no cross-talk between the two terms of (5.10) from which we might benefit when inverting for them simultaneously. In Figure 5-5d we see that the parallel FWI reconstruction is already quite inadequate for CDA data. Joint FWI on the other hand appears to give very similar results to DDFWI on the EL data as can be seen by comparing Figures 5-9 and 5-7d. This raises the question about why Parallel FWI and Joint FWI have such different characteristics despite their apparent similarities.

To investigate this we run Parallel FWI and Joint FWI on CDA data, both starting from the inverted CDA baseline model m_0 corresponding to Figure 5-4c. The

baseline gradient g_0 is defined as the zero-lag cross-correlation between the second time derivative of the incident wavefield and the adjoint wavefield q :

$$g_0(x) = \int_0^{t_{max}} \frac{\partial^2 u(m_0)}{\partial t^2}(x, t) q_0(x, t) dt. \quad (\text{C.1})$$

The monitor gradient g_1 is defined as

$$g_1(x) = \int_0^{t_{max}} \frac{\partial^2 u(m_0)}{\partial t^2}(x, t) q_1(x, t) dt, \quad (\text{C.2})$$

where I use $u(m_0)$ in the definition of g_1 because the baseline and monitor start from inverted baseline model m_0 . We obtain the adjoint wavefields by solving the backward wave equation with zero final conditions. The adjoint wavefield $q_0(x, t)$ satisfies:

$$\mathbf{L}q_0(x, t) = \left(\mathbf{S}^* (d_0 - \mathbf{S}u_0) \right)(x, t) = \left(\mathbf{S}^* r_0 \right)(x, t) \quad (\text{C.3})$$

where \mathbf{L} is the wave equation operator, \mathbf{S} is again the sampling operator and \mathbf{S}^* is its adjoint which places the traces on the grid as forcing term at the corresponding receiver locations. The baseline residual r_0 is defined in (5.6). Similarly, the adjoint wavefield $q_1(x, t)$ satisfies:

$$\mathbf{L}q_1(x, t) = \left(\mathbf{S}^* (d_1 - \mathbf{S}u_0) \right)(x, t) = \left(\mathbf{S}^* (d_0 + \delta d - \mathbf{S}u_0) \right)(x, t) = \left(\mathbf{S}^* (r_0 + \delta d) \right)(x, t) \quad (\text{C.4})$$

Subtracting (C.3) from (C.4) yields

$$\mathbf{L}(q_1(x, t) - q_0(x, t)) = \mathbf{L}\delta q(x, t) = \left(\mathbf{S}^* \delta d \right)(x, t) \quad (\text{C.5})$$

with this we see that the monitor gradient in (C.2) consists of two parts

$$\begin{aligned} g_1(x) &= \int_0^{t_{max}} \frac{\partial^2 u(m_0)}{\partial t^2}(x, t) (q_0(x, t) + \delta q(x, t)) dt \\ &= g_0(x) + \int_0^{t_{max}} \frac{\partial^2 u(m_0)}{\partial t^2}(x, t) \delta q(x, t) dt = g_0(x) + \delta g(x), \end{aligned} \quad (\text{C.6})$$

where $g_0(x)$ is the gradient contribution of the baseline residual and $\delta g(x)$ is the gradient contribution due to the time-lapse perturbation in the recorded monitor data.

Both the models and the gradients in our simulations are based on squared slowness. In Figure C-1 we convert the square slowness gradients in the first FWI iteration to velocity gradients to aid interpretation for CDA example. We notice that the baseline gradient g_0 induced by the residual r_0 features a large amplitude reflector at depth 3.0km. The starting model m_0 corresponding to Figure 5-4c does not match the very sudden velocity jump at the reservoir boundary, which causes this residual r_0 . We notice approximately the same reflector in the monitor g_1 , because the monitor data d_1 includes the baseline residual as can be seen in equation C.4. The difference between the search directions is displayed in Figure C-1c, and corresponds to the backpropagated data difference δd in our example with perfect source and receiver repeatability, as is seen in equation C.6. A small component of the reflector remains. The time-lapse velocity change delays the boundary reflection, resulting in a δd with recorded time similar to the boundary reflection. We interpret that this is backpropagated as a time-lapse reflector, just like we observe in the ellipse in Figure 5-15.

In Joint FWI the model vector is the concatenation of the baseline and the monitor components. In the first iteration where the equations above are valid (since $m_0 = m_1$), gradient descent in Joint FWI uses the following joint update vector

$$\begin{pmatrix} \Delta m_0^{JFWI} \\ \Delta m_1^{JFWI} \end{pmatrix} = -\alpha \begin{pmatrix} g_{0,i} \\ g_{0,i} + \delta g \end{pmatrix} \quad (\text{C.7})$$

Where the difference between the monitor step Δm_1^{JFWI} and the baseline step Δm_0^{JFWI} can be interpreted as a time-lapse step Δm^{JFWI} .

$$\Delta m^{JFWI} = \Delta m_1^{JFWI} - \Delta m_0^{JFWI} = -\alpha(g_0 + \delta g) + \alpha g_0 = -\alpha \delta g \quad (\text{C.8})$$

We see that in the first iteration JFWI moves down the direction δg , which is the backpropagated data difference δd according to (C.6). In Parallel FWI we invert

independently for the baseline and the monitor step

$$\Delta m_0^{PFWI} = -\alpha_0 g_0 \quad (\text{C.9})$$

$$\Delta m_0^{PFWI} = -\alpha_1 (g_0 + \delta g) \quad (\text{C.10})$$

The independent inversions each choose their own step length which result in a decrease of the objective function. Therefore step lengths α_0 and α_1 are in general different. The difference between the two updates can again be interpreted as a time-lapse step δm^{PFWI}

$$\Delta m^{PFWI} = \Delta m_1^{PFWI} - \Delta m_0^{PFWI} = -(\alpha_1 - \alpha_0)g_0 - \alpha_1 \delta g \quad (\text{C.11})$$

We see in (C.11) that the time-lapse update contains a contribution from the incomplete cancellation of the baseline residual gradient g_0 . The reason is that when doing the inversion for the baseline and the monitor independently, different step lengths can be used. In Joint FWI we concatenate the baseline and monitor components of the models and the gradients into vectors in equation C.7. When doing gradient descent we use a single step length α to add the concatenated gradient to the concatenated model, with α selected to decrease the joint objective function of (5.10). By using the same step length α for the baseline and monitor step we observe in (C.8) that the baseline residual component g_0 cancels out in the time-lapse step Δm^{JFWI} .

Figure C-2a shows the time-lapse update step Δm^{PFWI} in the first iteration of our example. A careful reader may notice that the update step in this first gradient descent iteration is relatively small in amplitude due to design choices made in PySIT. Larger update steps are used in subsequent inversion iterations when L-BFGS is used, although these subsequent iterations are not investigated in this Appendix. In Figure C-2a we see that there is a relatively large time-lapse reflector update which does not correspond to the physical V_p time-lapse in 5-2a. As argued before this time-lapse reflector is the result of the baseline residual r_0 which is not canceled out in parallel FWI due to the different step lengths in the baseline and the monitor

inversions. In the next iterations of FWI leading towards the final parallel estimate of Figure 5-5d, similarly undesirable updates appear to get worse. In the joint FWI update step in Figure C-2b we see that the step taken is just a scalar multiple of δg in Figure C-1c. We completely removed the influence of the backpropagated baseline residual g_0 in this example with perfect source/receiver repeatability. As a result we reduce the undesirable time-lapse reflector update, compared to Δm^{PFWI} in Figure C-2a. Only a small reflector component remains as we explained earlier in this Appendix. This observation illustrates how Joint FWI outperforms parallel FWI despite their apparent similarities by enforcing the same step length for the baseline and the monitor update.

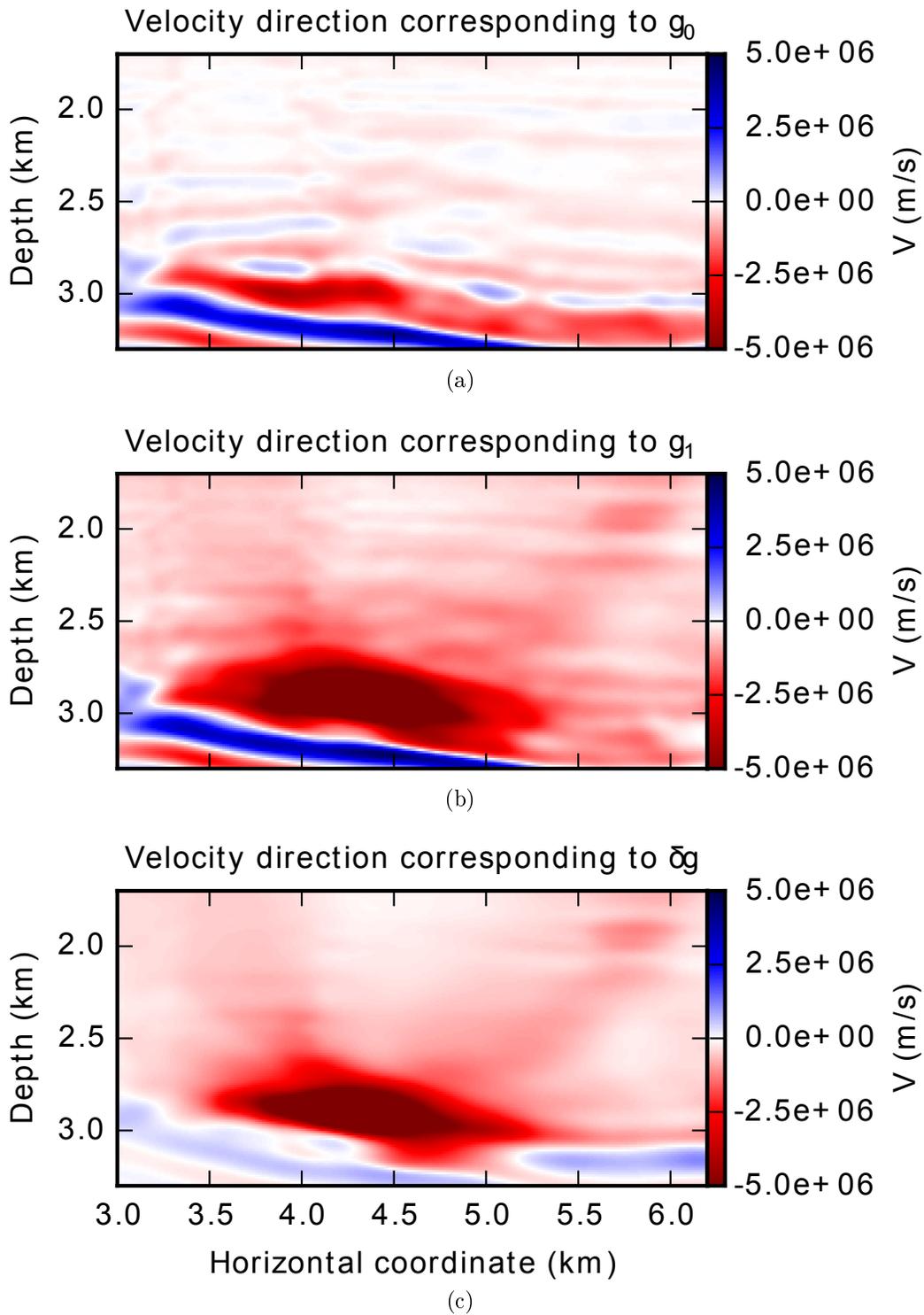


Figure C-1: Velocity update directions corresponding to the squared slowness gradients on the CDA data with m_0 corresponding to Figure 5-4c. Figures C-1a, C-1b and C-1c correspond to the gradients g_0 , g_1 and δg defined in (C.1), (C.2) and (C.6) respectively.

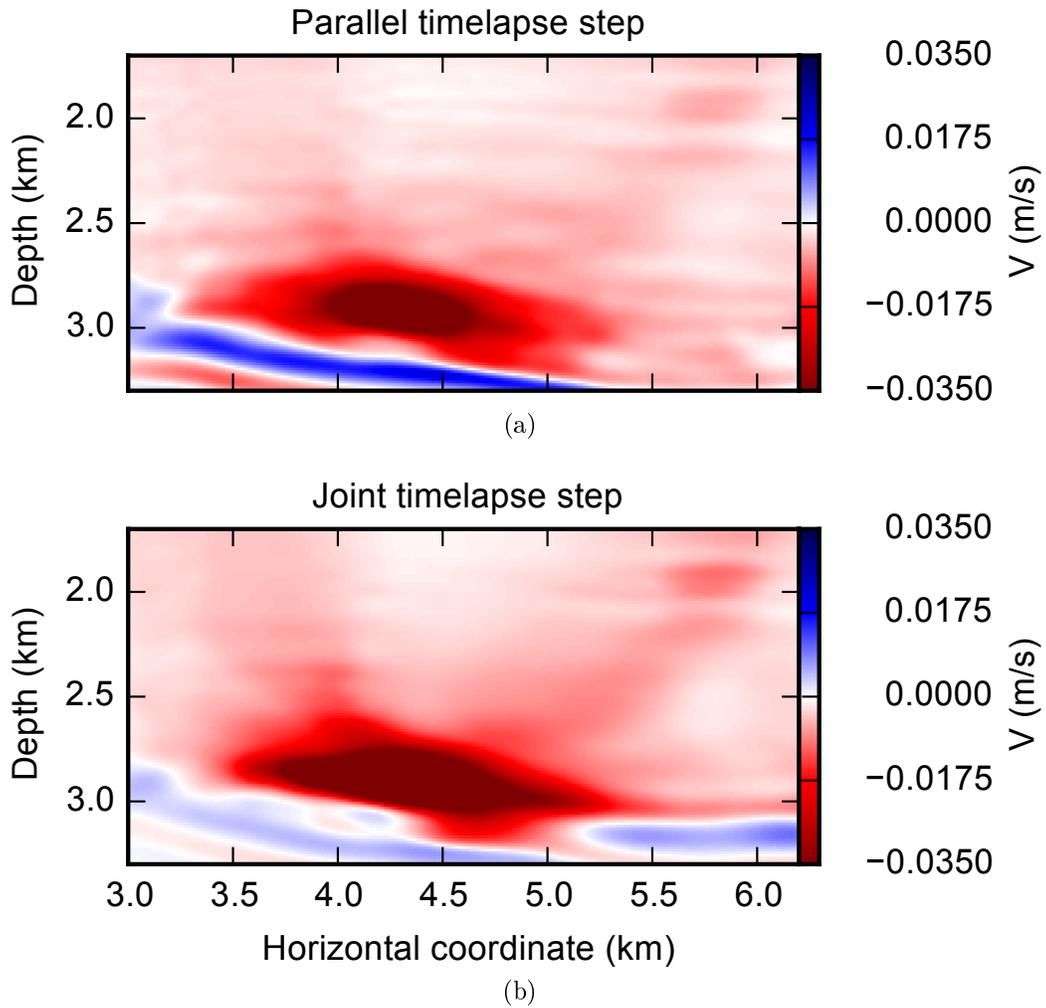


Figure C-2: Velocity steps taken in Parallel inversion and Joint inversion with velocity update directions corresponding to g_0 , g_1 and δg given in Figures C-1a, C-1b and C-1c respectively.

Appendix D

Heuristic amplitude compensation factors

In this section we introduce the heuristic amplitude compensation factors which are required when $\rho(\mathbf{x})$ and $V_s(\mathbf{x})$ are variable along the injection boundary, or between source and receiver locations.

D.1 Injection compensation factor $V_s(\mathbf{x})$

When injecting the wavefield on a model with nonzero S-velocity, the injected wavefield's pressure may be lower than that of the precomputed CDA pressure. This is easily seen by adding (6.3) to (6.4) and using that the average of the normal stresses equals the pressure.

$$\begin{aligned}\frac{\partial p(\mathbf{x}, t)}{\partial t} &= (\lambda(\mathbf{x}) + \mu(\mathbf{x})) \left(\frac{v_x(\mathbf{x}, t)}{\partial x} + \frac{v_z(\mathbf{x}, t)}{\partial z} \right) \\ &= \rho(\mathbf{x}) (V_p^2(\mathbf{x}) - V_s^2(\mathbf{x})) \left(\frac{v_x(\mathbf{x}, t)}{\partial x} + \frac{v_z(\mathbf{x}, t)}{\partial z} \right)\end{aligned}\tag{D.1}$$

Equation (D.1) clearly shows that the amplitude of pressure of the injected P-wave is proportional to $V_p^2(\mathbf{x}) - V_s^2(\mathbf{x})$. It is therefore lower when the shear velocity along the boundary of the local solver is unequal to zero. This proportionality factor will need to be compensated for later so that the background CDA wavefield between source

and receiver and the scattered wavefield from the elastic solver are scaled consistently.

We can use this scaling factor to approximately correct the amplitude of the CDA wavefield for smooth variation in V_s . The amplitude of the P-wave in the elastic simulation scales with $(V_p^2(\mathbf{x}) - V_s^2(\mathbf{x}))$. But a CDA simulation has $V_s = 0$ and this scaling term evaluates to $V_p^2(\mathbf{x})$. The amplitude compensation factor is therefore the division of these terms.

$$l_{V_s}^{inj}(\mathbf{x}) = \frac{V_p^2(\mathbf{x}) - V_s^2(\mathbf{x})}{V_p^2(\mathbf{x})}, \quad (\text{D.2})$$

This factor compensates the CDA simulation where $V_s(\mathbf{x}) = 0$ for the now elastic state where it is nonzero.

D.2 Injection compensation factor $\rho(\mathbf{x})$

In addition to V_s scaling we also observe that in a model with constant V_p but smoothly varying ρ , the amplitude of the pressure wave scales proportionally as

$$l_{\rho}^{inj}(\mathbf{x}) = \sqrt{\frac{\rho(\mathbf{x})}{\rho(\mathbf{x}_s)}} \quad (\text{D.3})$$

A derivation of this scaling is found in Appendix E. The density at the source $\rho(\mathbf{x}_s)$ is obtained from the elastic model where we want to approximate the P wavefield using the CDA model.

D.3 Remaining compensation factors

In order to propagate the scattered wavefield to the receiver locations in an elastic model we use the weighted boundary integral (6.15). In this boundary integral we

use the compensation factors

$$l_{V_s}^{rec}(\mathbf{x}) = \frac{V_p^2(\mathbf{x})}{V_p^2(\mathbf{x}) - V_s^2(\mathbf{x})} \frac{l_{V_s}^{inj}(\mathbf{x}_r)}{l_{V_s}^{inj}(\mathbf{x}_s)}, \quad (\text{D.4})$$

$$l_{\rho}^{rec}(\mathbf{x}) = \sqrt{\frac{\rho(\mathbf{x}_r)}{\rho(\mathbf{x})}} \quad (\text{D.5})$$

The first factor in (D.4) is the inverse of (D.2). It brings back the scattered wavefield to a S-velocity state corresponding to the CDA model. Similarly, factor $l_{\rho}^{rec}(\mathbf{x})$ approximately corrects the pressure from a variable density state to a constant density state. The weighted boundary integral in (6.15) therefore approximately uses a CDA representation of the pressure field along its boundary. We now need to compensate for differences in the V_s compensation factor $l_{V_s}^{inj}$ between source and receiver pixel. This amplitude variation due to interactions between P-velocity and S-velocity is not modeled by the CDA Green's functions. The second fraction in (D.4) takes care of this.

The wavefield in the perturbed model at the receiver locations is the sum of the background wavefield and the scattered wavefield. The scattered wavefield is computed using the procedure above. The same scaling needs to be applied to the background wavefield for this addition to be consistent. This will compensate for the amplitude variation that would be observed in the elastic model if the wavefield would smoothly transition from the source density and shear velocity to the that at the receiver. This compensation factor is,

$$l^0(\mathbf{x}_s, \mathbf{x}_r) = \sqrt{\frac{\rho(\mathbf{x}_r)}{\rho(\mathbf{x}_s)}} \frac{l_{V_s}^{inj}(\mathbf{x}_r)}{l_{V_s}^{inj}(\mathbf{x}_s)}, \quad (\text{D.6})$$

where the density and shear velocity values at the sources and receivers are extracted from the elastic model we want to approximate the pressure wavefield.

So far we have compensated the amplitude of the perturbed (i.e. background + scattered) wavefields for contrasts in density and S-velocity between source and receiver pixels. In the elastic solver we use (Levander, 1988), the amplitude of the

perturbed wavefield scales directly with the value of (D.2) at the source pixel. So the last amplitude compensation multiplies the perturbed gather by $l_{V_s}^{inj}(\mathbf{x}_s)$. The total workflow is plotted in Figure D-1.

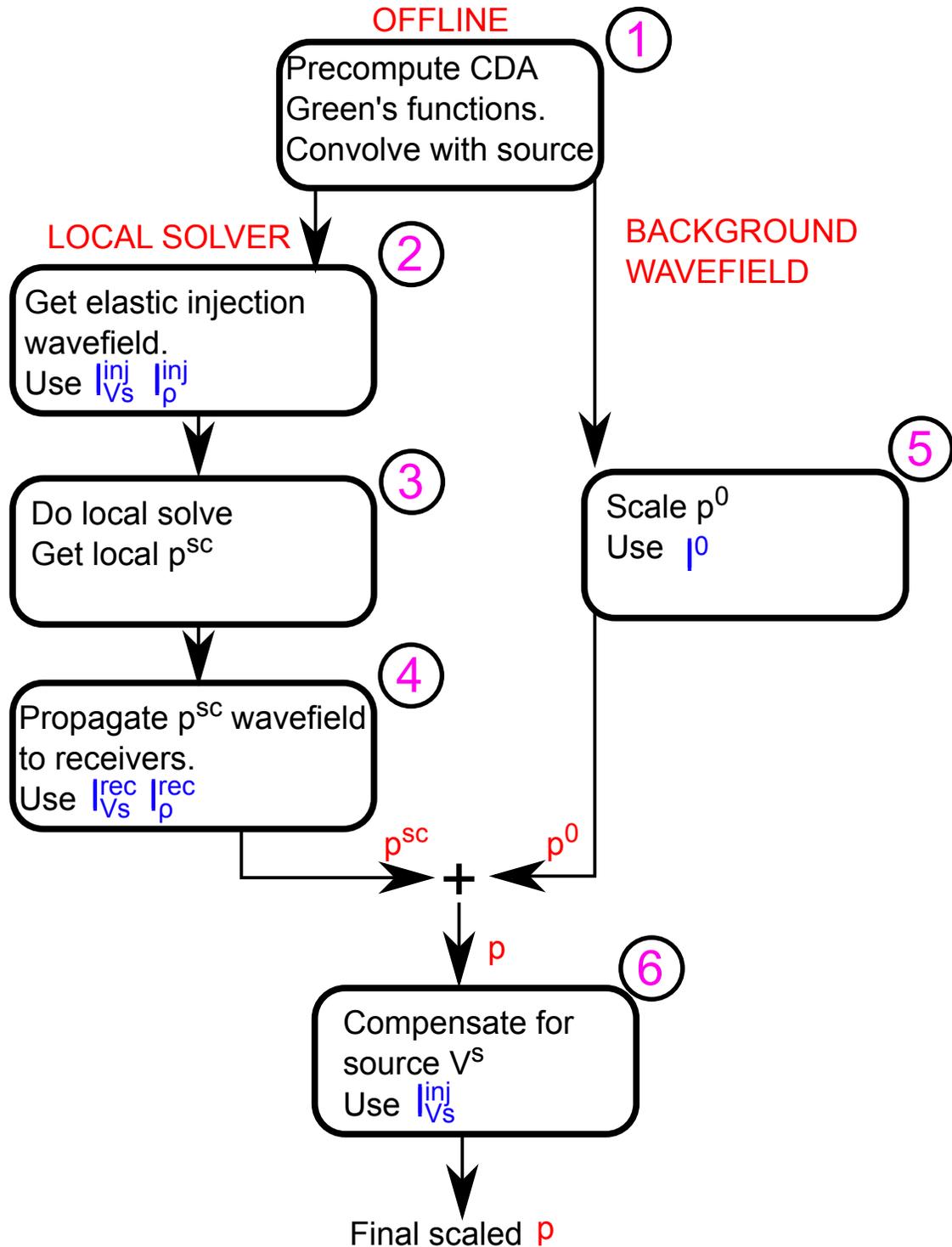


Figure D-1: Workflow of the proposed algorithm. Step one computes CDA Green's functions, which is the only time we use a full domain solver. Step two uses injection scaling factor (D.2) and (D.3). Step four uses (D.4) and (D.5) for propagating the wavefield to the receivers. Step five scales the background wavefield to compensate for differences in V_s and ρ between the source and receiver location, as defined in (D.6). Finally step six scales the wavefield for the shear velocity at the source location with $l_{V_s}^{inj}(\mathbf{x}_s)$.

Appendix E

Wavefield amplitude scaling due to smooth density variation

In this appendix we justify the density scaling term (D.3). We derive the scaling term using the 2D continuous equations of (6.1) - (6.5). To simplify the derivation we investigate the scenario where the model is invariant in the x direction. Furthermore, we investigate the propagation of a plane wave, instead of a cylindrical wavefront originating from a point source. We first investigate a simple scenario with constant material parameters ρ_0, λ_0 and μ_0 . Fourier transforming equations 6.2 and 6.4 and eliminating terms with derivatives in the x direction, we obtain

$$v_z^{(0)}(z, \omega) = \frac{1}{i\omega\rho_0} \frac{\partial\tau_{zz}^{(0)}(z, \omega)}{\partial z} \quad (\text{E.1})$$

$$\tau_{zz}^{(0)}(z, \omega) = \frac{\lambda_0 + 2\mu_0}{i\omega} \frac{\partial v_z^{(0)}(z, \omega)}{\partial z}, \quad (\text{E.2})$$

where the superscript “(0)” is used to make clear this equation is valid on the constant model. A simple second order ODE is obtained by inserting (E.1) into (E.2)

$$\tau_{zz}^{(0)}(z, \omega) = -\frac{\lambda_0 + 2\mu_0}{\omega^2\rho_0} \frac{\partial^2\tau_{zz}^{(0)}(z, \omega)}{\partial z^2} \quad (\text{E.3})$$

A general solution of this equation is

$$\tau_{zz}^{(0)}(z, \omega) = a_1(\omega) \exp \left[i \left(\sqrt{\frac{\omega^2 \rho_0}{\lambda_0 + 2\mu_0}} z + a_2(\omega) \right) \right], \quad (\text{E.4})$$

where a_1 and a_2 are potentially functions of ω . We now investigate a second scenario for which we use the “(1)” superscript. In this scenario density varies linearly with depth

$$\rho(z) = \rho_0 (1 + cz) \quad (\text{E.5})$$

While varying density, we keep the P-velocity and S-velocity fixed. As a result the elastic moduli $\lambda(z)$ and $\mu(z)$ have the same linear scaling as $\rho(z)$. The following equations are obtained for $v_z^{(1)}$ and $\tau_{zz}^{(1)}$

$$v_z^{(1)}(z, \omega) = \frac{1}{i\omega\rho_0(1+cz)} \frac{\partial \tau_{zz}^{(1)}(z, \omega)}{\partial z} \quad (\text{E.6})$$

$$\tau_{zz}^{(1)}(z, \omega) = (1+cz) \frac{\lambda_0 + 2\mu_0}{i\omega} \frac{\partial v_z^{(1)}(z, \omega)}{\partial z}, \quad (\text{E.7})$$

Where the $(1+cz)$ scaling in E.7 originates from the elastic moduli $\lambda(z)$ and $\mu(z)$. Inserting (E.6) into (E.7) we obtain the following ODE

$$\tau_{zz}^{(1)}(z, \omega) = -(1+cz) \frac{\lambda_0 + 2\mu_0}{\omega^2 \rho_0} \left[\frac{1}{(1+cz)} \frac{\partial^2 \tau_{zz}^{(1)}(z, \omega)}{\partial z^2} - \frac{c}{(1+cz)^2} \frac{\partial \tau_{zz}^{(1)}(z, \omega)}{\partial z} \right] \quad (\text{E.8})$$

Reorganizing terms we get

$$\frac{\partial^2 \tau_{zz}^{(1)}(z, \omega)}{\partial z^2} - \frac{c}{(1+cz)} \frac{\partial \tau_{zz}^{(1)}(z, \omega)}{\partial z} + \frac{\omega^2 \rho_0}{\lambda_0 + 2\mu_0} \tau_{zz}^{(1)}(z, \omega) = 0 \quad (\text{E.9})$$

Where we recognize that the first derivative term has a varying coefficient. ODE's with varying coefficients are often solved using power series methods. We choose to use Mathematica to obtain the following solution for $\tau_{zz}^{(1)}(z, \omega)$.

$$\tau_{zz}^{(1)}(z, \omega) = k_1(\omega)(1+cz)J_1 \left(\sqrt{d(\omega)} \frac{(1+cz)}{c} \right) + k_2(\omega)(1+cz)Y_1 \left(\sqrt{d(\omega)} \frac{(1+cz)}{c} \right) \quad (\text{E.10})$$

where J_1 and Y_1 are the first order Bessel functions of the first and second kind respectively. We simplify notation by using

$$d(\omega) = \frac{\omega^2 \rho_0}{\lambda_0 + 2\mu_0}. \quad (\text{E.11})$$

Intuitively it is to be expected that reflection of the incident plane-wave on the sloping density is insignificant when density variation is small over a wavelength. When the density difference across a wavelength is small (i.e. small slope c or large ω) the argument of the Bessel functions in (E.10) becomes large. Using their respective asymptotic expansions we see a complex exponentials arise.

$$\tau_{zz}^{(1)}(z, \omega) \approx \sqrt{\frac{c(1+cz)}{\pi\sqrt{d(\omega)}}} m_1(\omega) \exp \left[i \left(\sqrt{d(\omega)} \frac{(1+cz)}{c} - \frac{3}{4}\pi + m_2(\omega) \right) \right]. \quad (\text{E.12})$$

We now absorb all terms independent of z into $m_1(\omega)$ and $m_2(\omega)$ to get.

$$\tau_{zz}^{(1)}(z, \omega) \approx \sqrt{(1+cz)} m_1(\omega) \exp \left[i \left(\sqrt{\frac{\omega^2 \rho_0}{\lambda_0 + 2\mu_0}} z + m_2(\omega) \right) \right], \quad (\text{E.13})$$

where we also used (E.11). Equation E.13 shows that the amplitude scales directly with the dimensionless contrast cz . Finally we multiply (E.13) by $\sqrt{\frac{\rho_0}{\rho(z)}}$ and use (E.5) to obtain a solution involving density

$$\tau_{zz}^{(1)}(z, \omega) \approx \sqrt{\frac{\rho(z)}{\rho_0}} m_1(\omega) \exp \left[i \left(\sqrt{\frac{\omega^2 \rho_0}{\lambda_0 + 2\mu_0}} z + m_2(\omega) \right) \right]. \quad (\text{E.14})$$

The leading square root term was obtained by using (E.5) on the leftover term $\sqrt{1+cz}$ after absorbing constants into $m_1(\omega)$ and $m_2(\omega)$. We now compare the constant medium plane wave solution $\tau_{zz}^{(0)}(z, \omega)$ of (E.4) with the depth dependent density plane wave solution $\tau_{zz}^{(1)}(z, \omega)$ of (E.14). At depth $z = 0$ the density $\rho(z)$ is equal to ρ_0 and the leading square root term in (E.14) is just 1. At this depth we obtain the same solutions by setting $a_1(\omega) = m_1(\omega)$ and $a_2(\omega) = m_2(\omega)$. If we then divide (E.14) by (E.4), we observe that the difference due to the sloping density and constant P-

velocity and S-velocity is $\sqrt{\frac{\rho(z)}{\rho_0}}$. This is exactly the square root density scaling we use in (D.3). While this density scaling is contingent on the asymptotic approximation of the Bessel functions in (E.12), we numerically observe similar scaling over a wide range of contrasts c . This remains the same in our numerical studies when using a point source instead of the plane-wave used in this Appendix.

Appendix F

The high amplitude artificial scatter event in Local C

The artificial scatter event in Local C originates from the boundary integral. For each receiver location, boundary integral equation 6.15 is evaluated. Figure F-1 plots the boundary integrand for the receiver at $x = 8.0$ km and source at $x = 2.5$ km. Three vertical striped blue lines segment the boundary integrand into four sections. These sections are the top, right, bottom and left boundary of Local C (i.e. blue box in Figure 6-9a). Similar to the ordering of the sides, the nodes are also numbered in a clockwise manner. Figure F-1 zooms in on the red box of Figure F-1. Two vertical arrows indicate the node locations for the top chalk reflector and the bottom chalk reflector on the right side of the local domain respectively. At these locations there are spiky discontinuous features in the boundary integrand. Summing the boundary integrand along the magenta line, also outside the zoom region, and multiplying with the grid spacing performs the integral for $t = 4.85$ s. This gives the scattered field p^{sc} at $t = 4.85$ s at receiver location $x = 8.0$ km. Adding this scattered field to the background field p^0 gives the perturbed field plotted at that time and receiver coordinate in Figure 6-13b. We see that at this location the artificial scatter event is visible. We thus conclude that it originates from integration along the magenta line. In Local C the integrand does not cancel along the horizontal magenta line and this generates this event. Visually we see there is more black polarity than white polarity

along the magenta line. In addition to the mostly black area around the top chalk node, it also goes through a smaller black area at the bottom chalk, all contributing to the amplitude of the artifact by not canceling out.

We now study the boundary integrand of Local C to determine which physical event corresponds to the integrand at the intersection of the magenta line and the top chalk in Figure F-2a. To do so we first look at Figure F-1. Around boundary node 300 and $t \approx 4.15$ s we see the apex of a hyperbolic event, and a smaller event separates towards to top right. The latter has an apex around $t \approx 4.0$ s and boundary node number 380. The p^{sc} trace is formed by summing over the node numbers and multiplying by the grid spacing. Each apex is a 'stationary zone' for the integral because summation is constructive and both appear at the top boundary for a source at $x = 2.5$ km and receiver at 8.0 km. Comparing these times with the perturbed shot gather in Figure 6-13b we see that the thinner event with stationary zone at $t \approx 4.0$ s corresponds to the head-wave and the one with stationary zone at $t \approx 4.15$ s corresponds to the reflection. Following the latter in Figures F-1 and F-2a we see that this event is the one that generates the artifacts. Therefore we believe the large amplitude artificial reflection is caused by the incoming wave reflecting on the chalk right where the boundary integral crosses the fully elastic chalk in Local C.

The scatter event for Local C is not visible in the shot gather for Local D in Figure 6-13c. Figure F-2b shows a zoomed in section of the boundary integrand for Local D, again for source at $x = 2.5$ km and receiver at $x = 8.0$ km. The node numbering scheme is the same as for Local C, but the exact node numbers differ. The reason is that Local D adds a horizontal taper, which changes the number of nodes in the local domain. Just as in Figure F-2a, the left red arrow is aligned with the node at the top chalk and the right red arrow is aligned bottom chalk on the right side of the local domain. In this case the integrand passes beyond the top chalk very smoothly, no spiky phenomena appear. If the same artifact would be generated by Local D then we suspect the apex would be above the right boundary at $x = 9.0$ km in this case. It would also originate at a later time than in Local C since it the right boundary is further away from the source. Since we plot the integrand for a receiver at $x = 8.0$ km,

which is 1.0 km away from the apex of this already delayed hypothetical artifact, the event should arrive at an even later time. In Figure F-2b for Local D we therefore observe that the boundary integrand crosses the chalk layer around $t = 5.4$ s instead of $t = 4.85$ s. Unlike the integrand for Local C, there are no artifacts at the chalk crossings marked by the red arrows in Figure F-2b. Visually we can already see that integration along the magenta line is more balanced than for Local C. Black and white patches are more evenly distributed and no noticeable scatter event is generated for Local D. We emphasize that even though Figure 6-13c only shows Local D up to 5.0 s, no event is visible at later times either. This absence of the scatter event is not simply caused by extending the right boundary outwards. Had we changed Local C by moving the right boundary to $x = 8.0$ km instead of $x = 9.0$ km then a similar scatter event would have been generated, but with an apex close to $x = 9.0$ km. But for Local D with right boundary at $x = 9.0$ km no scatter event is generated. In Local D we have no S-velocity and density contrast along the interface. This apparently makes a big difference here, even though in both cases we technically violate the restriction of not updating the CDA background model at the integration boundary. In Local C, the reflection at the intersection of the chalk and the right boundary encounters a contrast in P-velocity, density and S-velocity. We suspect that the violation of the rule of perturbing the model only on the inside of the boundary integral is too strong here. The fully elastic material contrast fundamentally changes the reflection behavior compared to the CDA background state. And the artifact is generated as a result. While all based on heuristics, we attempt to limit (strong) elastic material contrasts from intersecting the boundary.

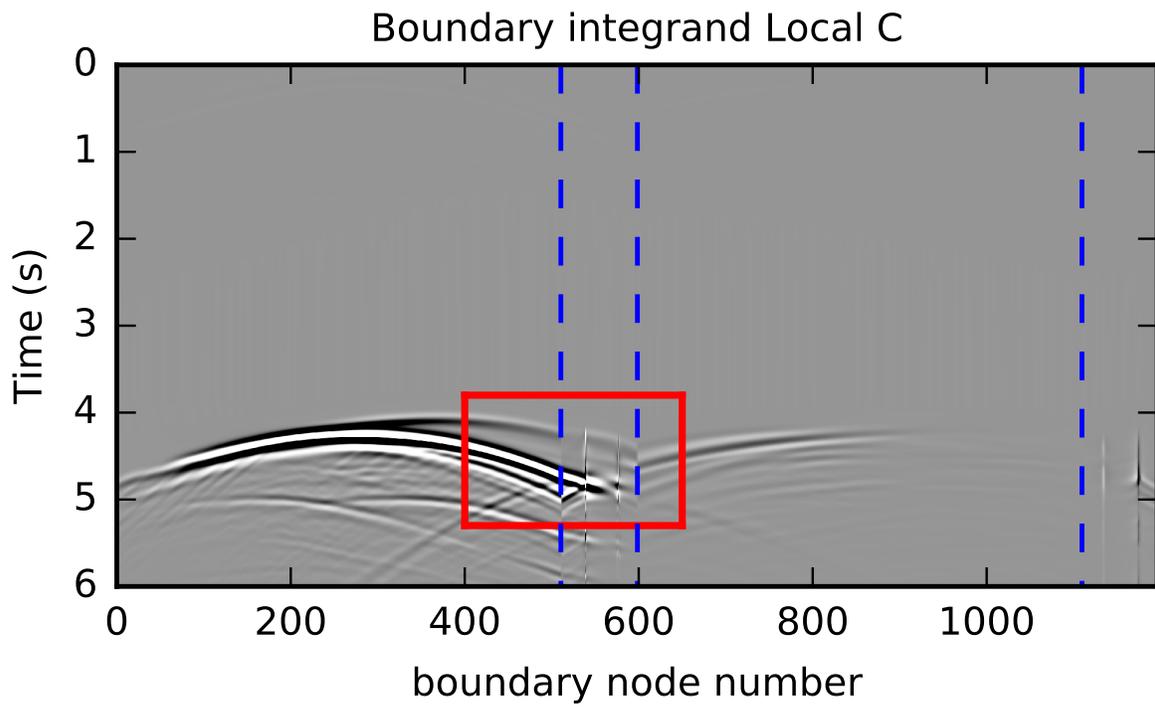


Figure F-1: Boundary integrand of (6.15) for Local C for a source at $x = 2.5$ km and receiver at $x = 8.0$ km. From left to right, the three vertical dashed blue lines represent the transition from the top→right, right→bottom and bottom→left boundary. Node numbering is also clockwise. The red box is a zoom region.

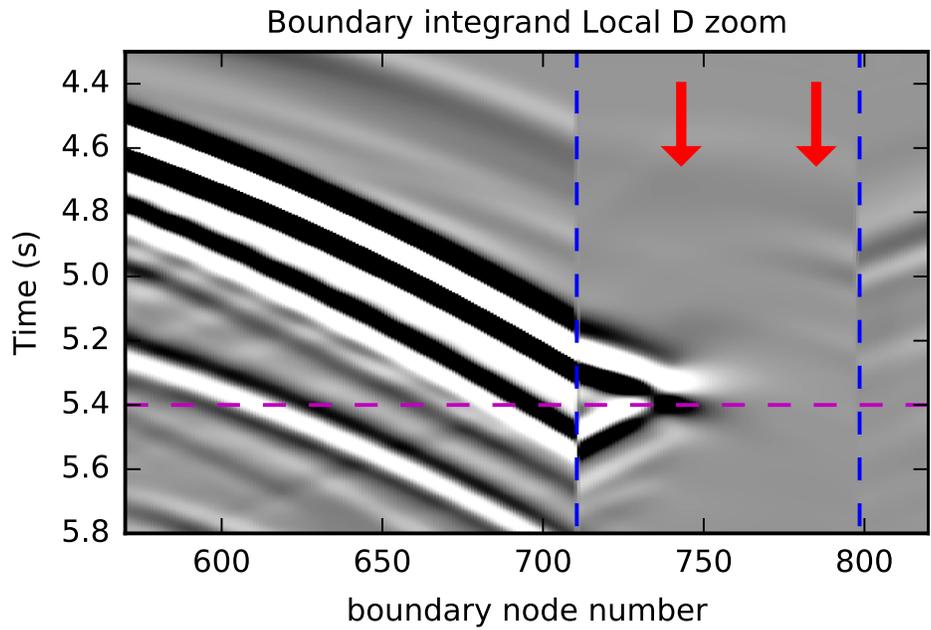
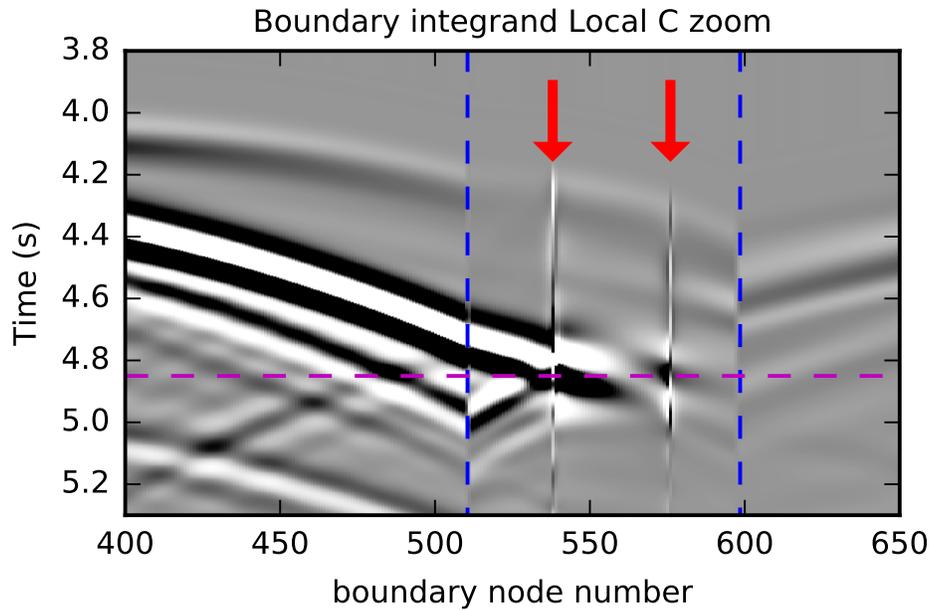


Figure F-2: (a) shows a zoom in on the red box in Figure F-1. The dashed blue lines are the same. The two vertical arrows indicate locations where the boundary integral passes the reflector, which is fully elastic in Local C since there is no taper. Notice the spiky features at the vertical arrows. When summing over all nodes along the horizontal magenta line we obtain the scattered field at $t = 4.85$ s. (b) is similar to (a), but now for Local D. The boundary node numbers are slightly larger because a horizontal taper is now used. When the boundary integral crosses the top of the reflector at the left arrow we observe that the transition in the integrand is very smooth. Because of this no scatter artifact is generated like we saw in Local C.

Bibliography

- Aki, K., and P. G. Richards, 2002, Quantitative seismology: **1**.
- Arts, R., M. Trani, R. Chadwick, O. Eiken, S. Dortland, and L. van der Meer, 2009, Acoustic and elastic modeling of seismic time-lapse data from the sleipner co2 storage operation: State of the science: AAPG Studies in Geology, **59**, 391–403.
- Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, P. Thore, and J. Virieux, 2015, Time-lapse seismic imaging using regularized full-waveform inversion with a prior model: which strategy?: Geophysical prospecting, **63**, 78–98.
- Asnaashari, A., R. Brossier, S. Garambois, F. Audebert, P. Thore, J. Virieux, et al., 2011, Sensitivity analysis of time-lapse images obtained by differential waveform inversion with respect to reference model: Presented at the 2011 SEG Annual Meeting, Society of Exploration Geophysicists.
- Bansal, R., J. Krebs, P. Routh, S. Lee, J. Anderson, A. Baumstein, A. Mullur, S. Lazaratos, I. Chikichev, and D. McAdow, 2013, Simultaneous-source full-wavefield inversion: The Leading Edge, **32**, 1100–1108.
- Berryhill, J., 1984, Wave-equation datuming before stack: Geophysics, **49**, no. 11, 2064–2066.
- Bertrand, A., P. Folstad, A. Grandi, G. Jeangeot, H. Haugvaldstad, B. Lyngnes, R. Midtun, and N. Haller, 2013, The ekofisk life of field seismic (lofs) system-experiences and results after two years in operation: Presented at the 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013.
- Bevc, D., 1995, Imaging under rugged topography and complex velocity structure: PhD thesis, Stanford University, Stanford, California, USA.
- Bishop, T., K. Bube, R. Cutler, R. Langan, P. Love, J. Resnick, R. Shuey, D. Spindler, and H. Wyld, 1985, Tomographic determination of velocity and depth in laterally varying media: Geophysics, **50**, 903–923.
- Borisov, D., S. C. Singh, and N. Fuji, 2015, An efficient method of 3-d elastic full waveform inversion using a finite-difference injection method for time-lapse imaging: Geophysical Journal International, **202**, 1908–1922.
- Brandsberg-Dahl, S., M. V. de Hoop, and B. Ursin, 1999, Velocity analysis in the common scattering-angle/azimuth domain: SEG Technical Program Expanded Abstracts 1999, 1715–1718.
- Broggini, F., R. Snieder, and K. Wapenaar, 2014, Data-driven wavefield focusing and imaging with multidimensional deconvolution: Numerical examples for reflection data with internal multiples: Geophysics, **79**, WA107–WA115.
- Brossier, R., S. Operto, J. Virieux, et al., 2009, Two-dimensional seismic imaging of

- the valhall model from synthetic obc data by frequency domain elastic full-waveform inversion: Presented at the 2009 SEG Annual Meeting, Society of Exploration Geophysicists.
- Buland, A., and H. Omre, 2003, Joint avo inversion, wavelet estimation and noise-level estimation using a spatially coupled hierarchical bayesian model: *Geophysical Prospecting*, **51**, 531–550.
- Byun, J., J. Yu, and S. Seol, 2010, Crosswell monitoring using virtual sources and horizontal wells: *Geophysics*, **75**, SA37–SA43.
- Calvert, M., L. Vagg, K. Lafond, A. Hoover, K. Ooi, and I. Herbert, 2014, Insights into sweep efficiency using 4d seismic at halfdan field in the north sea: *The Leading Edge*, **33**, 182–187.
- Cance, P., and Y. Capdeville, 2015, Validity of the acoustic approximation for elastic waves in heterogeneous media: *Geophysics*, **80**, T161–T173.
- Capdeville, Y., E. Chaljub, and J. Montagner, 2003, Coupling the spectral element method with a modal solution for elastic wave propagation in global earth models: *Geophysical Journal International*, **152**, 34–67.
- Červený, V., 1959, On the reflection of spherical waves at a plane interface with refractive index near to one. i: *Studia Geophysica et Geodaetica*, **3**, 116–134.
- Červený, V., and F. Hron, 1961, Reflection coefficients for spherical waves: *Studia Geophysica et Geodaetica*, **5**, 122–132.
- Chapman, C., J. Hobro, and J. Robertsson, 2014, Correcting an acoustic wavefield for elastic effects: *Geophysical Journal International*, **197**, 1196–1214.
- Colton, D., and R. Kress, 1998, *Inverse acoustic and electromagnetic scattering theory*, 2 ed.: Springer Science & Business Media.
- Davis, T., 2004, Algorithm 832: Umfpack v4.3 - an unsymmetric-pattern multifrontal method: *ACM Transactions on Mathematical Software (TOMS)*, **30**, no. 2, 196–199.
- , 2015, Time-lapse multi-component seismic monitoring, delhi field, louisiana: *First Break*, **33**, 65–70.
- Davis, T. L., M. J. Terrell, R. D. Benson, R. Cardona, R. R. Kendall, and R. Winarsky, 2003, Multicomponent seismic characterization and monitoring of the co2 flood at weyburn field, saskatchewan: *The Leading Edge*, **22**, 696–697.
- de Ridder, S., B. Biondi, B. Clapp, et al., 2014, Time-lapse ambient-seismic-noise tomography at valhall: Presented at the 2014 SEG Annual Meeting, Society of Exploration Geophysicists.
- Denli, H., L. Huang, et al., 2009, Double-difference elastic waveform tomography in the time domain: Presented at the 2009 SEG Annual Meeting, Society of Exploration Geophysicists.
- Diaz, E., and P. Sava, 2013, Data-domain and image-domain wavefield tomography: *The Leading Edge*, **32**, 1064–1071.
- Dix, C., 1955, Seismic velocities from surface measurements: *Geophysics*, **20**, 68–86.
- Dong, S., Y. Luo, X. Xiao, S. Chávez-Pérez, and G. T. Schuster, 2009, Fast 3d target-oriented reverse-time datuming: *Geophysics*, **74**, WCA141–WCA151.
- Dorn, O., E. Miller, and C. Rappaport, 2000, A shape reconstruction method for electromagnetic tomography using adjoint fields and level sets: *Inverse Problems*,

- 16**, 1119–1156.
- Downton, J. E., and C. Ursenbach, 2006, Linearized amplitude variation with offset (avo) inversion with supercritical angles: *Geophysics*, **71**, E49–E55.
- Eggenberger, K., P. Christie, D.-J. van Manen, and M. Vassallo, 2014, Multisensor streamer recording and its implications for time-lapse seismic and repeatability: *The Leading Edge*, **33**, 150–162.
- Etgen, J. T., C. Chu, T. Yang, and M. Vyas, 2014, Adaptive image focusing: SEG Technical Program Expanded Abstracts 2014, 3774–3778.
- Fichtner, A., 2010, Full seismic waveform modelling and inversion: Springer. *Advances in Geophysical and Environmental Mechanics and Mathematics*.
- Fokkema, J., and P. van den Berg, 2013, *Seismic applications of acoustic reciprocity*: Elsevier Science.
- Fomel, S., M. van der Baan, et al., 2010, Local similarity with the envelope as a seismic phase detector: 80th Ann. Internat. Mtg: Soc. of Expl. Geophys, 1555–1559.
- Gardner, G., L. Gardner, and A. Gregory, 1974, Formation velocity and density—the diagnostic basics for stratigraphic traps: *Geophysics*, **39**, 770–780.
- Gassmann, F., 1951, Über die elastizität poröser medien.
- Geertsma, J., et al., 1973, Land subsidence above compacting oil and gas reservoirs: *Journal of Petroleum Technology*, **25**, 734–744.
- Ghaderi, A., and M. Landrø, 2009, Estimation of thickness and velocity changes of injected carbon dioxide layers from prestack time-lapse seismic data: *Geophysics*, **74**, O17–O28.
- Gillman, A., A. Barnett, and P.-G. Martinsson, 2013, A spectrally accurate direct solution technique for frequency-domain scattering problems with variable media: arXiv preprint arXiv:1308.5998.
- Gong, T., and G. A. McMechan, 2016, Target-oriented linear least squares and non-linear, trust-region newton inversions of plane waves using ava and pva data for elastic model parameters: *Geophysics*, **81**, R325–R338.
- Grote, M., and I. Sim, 2011, Local nonreflecting boundary condition for time-dependent multiple scattering: *Journal of Computational Physics*, **230**, 3135–3154.
- Guenther, R., and J. Lee, 1988, *Partial differential equations of mathematical physics and integral equations*: Dover Publications, Inc. Dover books on mathematics.
- Haffinger, P., 2012, *Seismic broadband full waveform inversion by shot/receiver refocusing*: PhD thesis, TU Delft, Delft, The Netherlands.
- Han, D.-h., M. Batzle, et al., 2001, Velocity, density and modulus of hydrocarbon fluids—data measurement: Presented at the Society of Exploration Geophysicist (SEG) Annual Meeting.
- Hatchell, P., and S. Bourne, 2006, Measuring reservoir compaction using time-lapse timeshifts: Presented at the 68th EAGE Conference & Exhibition.
- Herrmann, F., I. Hanlon, T. van Leeuwen, X. Li, B. Smithyman, and H. Wason, 2013, Frugal full-waveform inversion: From theory to a practical algorithm: *The Leading Edge*, **32**.
- Hewett, R., L. Demanet, and the PySIT Team, 2013, PySIT: Python seismic imaging toolbox v0.5. (Release 0.6).
- Hobro, J. W., C. H. Chapman, and J. O. Robertsson, 2014, A method for correcting

- acoustic finite-difference amplitudes for elastic effects: *Geophysics*, **79**, T243–T255.
- Hodgson, N., C. MacBeth, L. Duranti, J. Rickett, and K. Nihei, 2007, Inverting for reservoir pressure change using time-lapse time strain: Application to genesis field, gulf of mexico: *The Leading Edge*, **26**, 649–652.
- Japsen, P., 1998, Regional velocity-depth anomalies, north sea chalk: a record of overpressure and neogene uplift and erosion: *AAPG bulletin*, **82**, 2031–2074.
- J.Chiu, and L.Demanet, 2012, Matrix probing and its conditioning: *Siam Journal of Numerical Analysis*, **50**, 171–193.
- Jin, H., G. A. McMechan, and H. Guan, 2014, Comparison of methods for extracting adcigs from rtm: *Geophysics*, **79**, S89–S103.
- Johnston, D. H., 2013, Practical applications of time-lapse seismic data: Distinguished Instructor Series, Society of Exploration Geophysicists.
- Jugnon, V., and L. Demanet, 2013, Interferometric inversion: a robust approach to linear inverse problems: *SEG Technical Program Expanded Abstracts 2013*, 5180–5184.
- Kaipio, J., and E. Somersalo, 2006, Statistical and computational inverse problems: Springer Science & Business Media.
- Keys, R., and A. Weglein, 1983, Generalized linear inversion and the first born theory for acoustic media: *Journal of Mathematical Physics*, **24**, no. 6, 1444–1449.
- Krebs, J., J. Anderson, D. Hinkley, R. Neelamani, S. Lee, A. Baumstein, and M. Lacasse, 2009, Fast full-wavefield seismic inversion using encoded sources: *Geophysics*, **74**, WCC177–WCC188.
- Kumar, D., D. P. Hampson, B. H. Russell, B. Bankhead, S. Bland, P. Griffiths, D. Hodge, A. Ravaglia, H. Raghunath, J. Abraham, et al., 2006, A tutorial on gassmann fluid substitution: formulation, algorithm and matlab code: *matrix*, **2**, 1.
- Lailly, P., 1983, The seismic inverse problem as a sequence of before-stack migrations: Presented at the Conference on Inverse Scattering: Theory and Application, SIAM.
- Landrø, M., and J. Stammeijer, 2004, Quantitative estimation of compaction and velocity changes using 4d impedance and travelttime changes: *Geophysics*, **69**, 949–957.
- Levander, A. R., 1988, Fourth-order finite-difference p-sv seismograms: *Geophysics*, **53**, 1425–1436.
- Lewis, W., B. Starr, and D. Vigh, 2012, A level set approach to salt geometry inversion in full-waveform inversion: *SEG Technical Program Expanded Abstracts 2012*, 1–5.
- Liu, F., L. Guasch, S. A. Morton, M. Warner, A. Umpleby, Z. Meng, S. Fairhead, and S. Checkles, 2012, 3-d time-domain full waveform inversion of a valhall obc dataset: *SEG Technical Program Expanded Abstracts*, 1–5.
- Lytle, R. J., and K. A. Dines, 1980, Iterative ray tracing between boreholes for underground image reconstruction: *IEEE Transactions on Geoscience and Remote Sensing*, 234–240.
- Maharramov, M., and B. Biondi, 2014, Robust joint full-waveform inversion of time-lapse seismic data sets with total-variation regularization: arXiv preprint arXiv:1408.0645.
- Maharramov, M., B. Biondi, et al., 2014, Joint full-waveform inversion of time-lapse

- seismic data sets: Presented at the 2014 SEG Annual Meeting, Society of Exploration Geophysicists.
- Maharramov, M., B. Biondi, and M. Meadows, 2015, Simultaneous tv-regularized time-lapse fwi with application to field data: Presented at the 2015 SEG Annual Meeting, Society of Exploration Geophysicists.
- Malcolm, A., and B. Willemsen, 2016, Rapid 4d FWI using a local wave solver: *The Leading Edge*, **35**, 1053–1059.
- Martin, G. S., S. Larsen, K. Marfurt, et al., 2002, Marmousi-2: An updated model for the investigation of avo in structurally complex areas: Presented at the 2002 SEG Annual Meeting, Society of Exploration Geophysicists.
- Masson, Y., P. Cupillard, Y. Capdeville, and B. Romanowicz, 2013, On the numerical implementation of time-reversal mirrors for tomographic imaging: *Geophysical Journal International*, **196**, no. 3, 1580–1599.
- Métivier, L., R. Brossier, J. Virieux, and S. Operto, 2013, Full waveform inversion and the truncated newton method: *SIAM Journal on Scientific Computing*, **35**, B401–B437.
- Monteiller, V., S. Chevrot, D. Komatitsch, and N. Fuji, 2013, A hybrid method to compute short-period synthetic seismograms of teleseismic body waves in a 3-d regional model: *Geophysical Journal International*, **192**, 230–247.
- Mulder, W., 2005, Rigorous redatuming: *Geophysical Journal International*, **161**, no. 2, 401–415.
- Mulder, W., and R.-E. Plessix, 2008, Exploring some issues in acoustic full waveform inversion: *Geophysical Prospecting*, **56**, 827–841.
- Nocedal, J., and S. Wright, 2006a, Numerical optimization: Springer Science & Business Media.
- Nocedal, J., and S. J. Wright, 2006b, Numerical optimization, 2 ed.: Springer.
- O’Brien, P., 1963, A note on the reflection of seismic pulses with application to second event refraction shooting: *Geophysical prospecting*, **11**, 59–72.
- Plessix, R.-É., 2009, Three-dimensional frequency-domain full-waveform inversion with an iterative solver: *Geophysics*, **74**, WCC149–WCC157.
- Plessix, R.-E., G. Baeten, J. W. de Maag, M. Klaassen, Z. Rujie, T. Zhifei, et al., 2010, Application of acoustic full waveform inversion to a low-frequency large-offset land data set: *SEG Technical Program Expanded Abstracts 2010*, 930–934.
- Pratt, R. G., 1999, Seismic waveform inversion in the frequency domain, part 1: Theory and verification in a physical scale model: *Geophysics*, **64**, 888–901.
- Prieux, V., R. Brossier, S. Operto, and J. Virieux, 2013, Multiparameter full waveform inversion of multicomponent ocean-bottom-cable data from the valhall field. part 1: imaging compressional wave speed, density and attenuation: *Geophysical Journal International*, **194**, no. 3, 1640–1664.
- Queißer, M., S. Singh, et al., 2010, Time lapse seismic monitoring of co2 sequestration at sleipner using time domain 2d full waveform inversion: Presented at the 2010 SEG Annual Meeting, Society of Exploration Geophysicists.
- Rickett, J., L. Duranti, T. Hudson, N. Hodgson, et al., 2006, Compaction and 4-d time strain at the genesis field: Presented at the 2006 SEG Annual Meeting, Society of Exploration Geophysicists.

- Rickett, J., L. Duranti, T. Hudson, B. Regel, and N. Hodgson, 2007, 4d time strain and the seismic signature of geomechanical compaction at genesis: *The Leading Edge*, **26**, 644–647.
- Robertsson, J., and C. Chapman, 2000, An efficient method for calculating finite-difference seismograms after model alterations: *Geophysics*, **65**, 907–918.
- Robertsson, J. O., S. Ryan-Grigor, C. M. Sayers, and C. H. Chapman, 2000, A finite-difference injection approach to modeling seismic fluid flow monitoring: *Geophysics*, **65**, 896–906.
- Røste, T., O. P. Dybvik, and O. K. Søreide, 2015, Overburden 4d time shifts induced by reservoir compaction at snorre field: *The Leading Edge*, **34**, 1366–1374.
- Røste, T., A. Stovas, and M. Landrø, 2006, Estimation of layer thickness and velocity changes using 4d prestack seismic data: *Geophysics*, **71**, S219–S234.
- Routh, P., G. Palacharla, I. Chikichev, S. Lazaratos, et al., 2012, Full wavefield inversion of time-lapse data for improved imaging and reservoir characterization: Presented at the 2012 SEG Annual Meeting, Society of Exploration Geophysicists.
- Sava, P., B. Biondi, S. Fomel, et al., 2001, Amplitude-preserved common image gathers by wave-equation migration: 71st Ann. Internat. Mtg: Soc. of Expl. Geophys, 296–299.
- Sava, P., and B. Biondo, 2004, Wave-equation migration velocity analysis. i. theory: *Geophysical Prospecting*, **52**, 593–606.
- Schuster, G., 2009, *Seismic interferometry*: Cambridge University Press.
- Schuster, G., and M. Zhou, 2006, A theoretical overview of model-based and correlation-based redatuming methods: *Geophysics*, **71**, SI103–SI110.
- Shin, C., S. Jang, and D.-J. min, 2001, Improved amplitude preservation for prestack depth migration by inverse scattering theory: *Geophysical Prospecting*, **49**, 592–606.
- Sirgue, L., and R. G. Pratt, 2004, Efficient waveform inversion and imaging: A strategy for selecting temporal frequencies: *Geophysics*, **69**, 231–248.
- Smith, G., and P. Gidlow, 1987, Weighted stacking for rock property estimation and detection of gas: *Geophysical Prospecting*, **35**, 993–1014.
- Stork, C., and R. Clayton, 1991, Linear aspects of tomographic velocity analysis: *Geophysics*, **56**, no. 4, 483–495.
- Tang, Y., and B. Biondi, 2011, Target-oriented wavefield tomography using synthesized born data: *Geophysics*, **49**, no. 5.
- Tarantola, A., 1984, Inversion of seismic reflection data in the acoustic approximation: *Geophysics*, **49**, 1259–1266.
- Teng, Z.-H., 2003, Exact boundary condition for time-dependent wave equation based on boundary integral: *Journal of Computational Physics*, **190**, 398–418.
- Tura, A., T. Barker, P. Cattermole, C. Collins, J. Davis, P. Hatchell, K. Koster, P. Schutjens, and P. Wills, 2005, Monitoring primary depletion reservoirs using amplitudes and time shifts from high-repeat seismic surveys: *The Leading Edge*, **24**, 1214–1221.
- van Leeuwen, T., and W. A. Mulder, 2007, 364, *in* Data correlation for velocity inversion: Society of Exploration Geophysicists, 1800–1804.
- van Manen, D.-J., A. Curtis, and J. O. Robertsson, 2006, Interferometric model-

- ing of wave propagation in inhomogeneous elastic media using time reversal and reciprocity: *Geophysics*, **71**, SI47–SI60.
- van Manen, D.-J., J. O. Robertsson, and A. Curtis, 2005, Modeling of wave propagation in inhomogeneous media: *Physical Review Letters*, **94**, 164301.
- van Manen, D.-J., J. O. A. Robertsson, and A. Curtis, 2007, Exact wave field simulation for finite-volume scattering problems: *The Journal of the Acoustical Society of America*, **122**, EL115–EL121.
- Vasmel, M., J. O. A. Robertsson, D.-J. van Manen, and A. Curtis, 2013, Immersive experimentation in a wave propagation laboratory: *The Journal of the Acoustical Society of America*, **134**, EL492–EL498.
- Vigh, D., B. Starr, J. Kapoor, H. Li, et al., 2010, 3d full waveform inversion on a gulf of mexico waz data set: Presented at the 2010 SEG Annual Meeting, Society of Exploration Geophysicists.
- Virieux, J., and S. Operto, 2009, An overview of full-waveform inversion in exploration geophysics: *Geophysics*, **74**, WCC1–WCC26.
- Wang, S., M. V. de Hoop, and J. Xia, 2011, On 3d modeling of seismic wave propagation via a structured parallel multifrontal direct Helmholtz solver: *Geophysical Prospecting*, **59**, 857–873.
- Watanabe, T., S. Shimizu, E. Asakawa, T. Matsuoka, et al., 2004, Differential waveform tomography for time-lapse crosswell seismic data with application to gas hydrate production monitoring: Presented at the 2004 SEG Annual Meeting, Society of Exploration Geophysicists.
- White, D., 2009, Monitoring co2 storage during eor at the weyburn-midale field: *The Leading Edge*, **28**, 838–842.
- Wiggins, J., 1984, Kirchhoff integral extrapolation and migration of nonplanar data: *Geophysics*, **49**, no. 8, 1239–1248.
- Willemsen, B., J. Cao, and B. Roy, 2016a, The impact of the acoustic approximation on time-lapse fwi, *in* SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 5435–5440.
- Willemsen, B., W. Lewis, A. Malcolm, et al., 2015a, Toward target-oriented fwi: An exact local wave solver applied to salt boundary inversion: Presented at the 2015 SEG Annual Meeting, Society of Exploration Geophysicists.
- Willemsen, B., and A. Malcolm, 2016, Coupling a local elastic solver to a background acoustic model to estimate phase variation, *in* SEG Technical Program Expanded Abstracts 2016: Society of Exploration Geophysicists, 3945–3949.
- Willemsen, B., A. Malcolm, et al., 2015b, Regularizing velocity differences in time-lapse fwi using gradient mismatch information: Presented at the 2015 SEG Annual Meeting, Society of Exploration Geophysicists.
- Willemsen, B., A. Malcolm, and W. Lewis, 2016b, A numerically exact local solver applied to salt boundary inversion in seismic full-waveform inversion: *Geophysical Journal International*, **204**, 1703–1720.
- Yang, D., 2014, Full wavefield inversion methods for monitoring time-lapse subsurface velocity changes: PhD thesis, Massachusetts Institute of Technology.
- Yang, D., M. Meadows, P. Inderwiesen, J. Landa, A. Malcolm, and M. Fehler, 2015, Double-difference waveform inversion: Feasibility and robustness study with pres-

- sure data: *Geophysics*, **80**, M129–M141.
- Yang, D., Y. Zheng, M. Fehler, and A. Malcolm, 2012, Target-oriented time-lapse waveform inversion using virtual survey: *SEG Technical Program Expanded Abstracts 2012*, 1–5.
- Yu, P., J. Geng, X. Li, and C. Wang, 2016, Acoustic-elastic coupled equation for ocean bottom seismic data elastic reverse time migration: *Geophysics*, **81**, S333–S345.
- Zamanian, S. A., D. Yang, W. L. Rodi, and M. C. Fehler, 2014, Hierarchical bayesian inversion of time-lapse seismic data with different acquisition geometries: Presented at the 76th EAGE Conference and Exhibition 2014.
- Zepeda-Núñez, L., and L. Demanet, 2016, The method of polarized traces for the 2d helmholtz equation: *Journal of Computational Physics*, **308**, 347–388.
- Zhang, Z., and L. Huang, 2013, Double-difference elastic-waveform inversion with prior information for time-lapse monitoring: *Geophysics*, **78**, R259–R273.
- Zhu, X., G. McMechan, et al., 2011, Comparison of methods for modeling phase variation with angle: 2011 SEG Annual Meeting.
- , 2012a, Amplitude and phase in angle-domain common-image gathers: Presented at the 2012 SEG Annual Meeting, Society of Exploration Geophysicists.
- , 2012b, Analysis of phase variation with ray parameter in the τ -p domain: Presented at the 2012 SEG Annual Meeting, Society of Exploration Geophysicists.
- Zhu, X., and G. A. McMechan, 2012, Elastic inversion of near-and postcritical reflections using phase variation with angle: *Geophysics*, **77**, R149–R159.
- , 2014, Amplitude and phase versus angle for elastic wide-angle reflections in the τ -p domain: *Geophysics*, **80**, N1–N9.
- Zoeppritz, K., 1919, Vii b. über reflexion und durchgang seismischer wellen durch unstetigkeitsflächen: *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, **1919**, 66–84.