NTNU
Norwegian University of
Science and Technology

Inverse Scattering Series internal multiple prediction through utilization of sparse transforms

Including high-performance
implementational aspects

Ole Edvard Aaker

Petroleum Geoscience and Engineering
Submission date: June 2018
Supervisor: Børge Arntsen, IGP

Norwegian University of Science and Technology
Department of Geoscience and Petroleum

Freedom from the desire for an answer is essential to the understanding of a problem.
J.Krishnamurti

Summary

Industry standard imaging algorithms do not treat internal multiples correctly. 'Cross-talk' with primary reflections may therefore occur in the imaging procedure, potentially causing significant artifacts in the image of the subsurface.

The Inverse Scattering Series (abbr. ISS) internal multiple prediction algorithms enable completely data-driven computation of internal multiple models that are kinematically accurate with well-approximated amplitudes. The ISS prediction algorithms are therefore key candidates for computing internal multiple models suitable for use in adaptive subtraction procedures. This thesis work has been centered around implementation of the Inverse Scattering Series predictors in the coupled plane wave domain in 2D, and its 1.5D variant in the plane wave domain.

The prediction codes implemented employ a vast range of optimizations, including algorithmic optimizations and code optimizations, such as vectorization and parallelization. In the case of restricted dips of interfaces related to internal multiple generation the coupled plane wave domain easily facilitates a reduction of the number of required computations.

Linear Radon transforms constitute the mapping of the input data to the coupled plane wave domain. Due to the non-orthogonality of the forward and inverse transform pairs, certain aperture artifacts may arise in the input data after mapping to the prediction domain. Experimental results demonstrated that this may yield some undesired artifacts in the calculated multiple model. It should be noted that, compared to the 2D predictor, the 1.5D prediction algorithm does not appear as sensitive to artifacts present in the plane wave domain.

Motivated by the possibility to minimize artifacts and, in general, to improve the prediction output lead to the implementation of a so-called high-resolution or sparse linear RT. By virtue of its design, where sparseness constraints are imposed in $\tau - p$ domain directly, it is effective at both reducing aperture artifacts and as well as compressing the temporal support of the signal.

The usage of high-resolution RTs for transformation to the prediction domain resulted in internal multiple predictions with less artifacts and improved (apparent) waveform matches. Results from adaptive subtraction demonstrated improved internal multiple attenuation, compared to using standard Radon transforms. Automatic regularization routines for the sparse Radon transform can give further benefits, especially for prediction in the coupled plane wave domain.

A previously proposed procedure for enabling multidimensional internal multiple prediction using migrated datasets has been demonstrated with a simple proof of concept. The

procedure is considered suitable for application to real data.

Sammendrag

Migrasjonsalgoritmer som benyttes i anvendt geofysikk behandler ikke interne multipler på en korrekt måte. På grunn av dette kan de 'kryss-snakke' med primære refleksjoner i migrasjonsprosedyren, noe som kan gi betydelige artefakter i avbildningen av undergrunnen.

Prediksjonsalgoritmer for interne multipler som kommer fra *Inverse Scattering Series* (forkortet ISS) muligjør utregning av modeller for interne multipler som er kinematisk korrekte og med amplituder som er gode approksimasjoner. Dette gjøres på en fullstendig datadrevet måte, og de kalkulerte modellene for interne multipler er godt egnet for bruk i adaptive subtraksjonsprosedyrer. Arbeidet tilknyttet denne mastergraden har vært fokusert rundt implementasjon av ISS prediksjonsalgoritmene i det koblede planbølgedomene i 2D, og dens 1.5D variant i ordinært planbølgedomene.

Prediksjonskodene som har blitt implementert inneholder mange optimaliseringer for å forbedre kjøretiden, f.eks. som optimaliseringer tilknyttet algoritmen og optimaliseringer av koden, slik som vektorisering og parallellisering. Det koblede planbølgedomene muliggjør, på en enkel måte, reduksjon i antall utregninger i situasjoner hvor reflektorer som er involvert i generering av interne multipler har begrensede vinkler (ift horisontale akser).

Lineære Radon transformasjoner benyttes for å dekomponere bølgefeltsdata til sine planbølgekomponenter, som igjen brukes i prediksjonsalgoritmene. Fordi framover- og inverstransformasjonene ikke er ortogonale på hverandre, medfører bruk av denne typen transformasjon ofte enkelte artefakter i transformdomenet. Eksperimentelle resultater viser at disse artifaktene kan medføre andre artefakter i den predikterte modellen av interne multipler. Riktignok virker det som at 1.5D prediksjonsalgoritmen ikke er like sensitiv til artefakter i planbølgedomenet, når en sammenlikner med 2D algoritmen.

Muligheten til å kunne minimere artifakter og forbedre prediksjonsresultatet ble en motivasjon for å implementere høy-oppløsnings eller sparsom linære RT. Fordi kravene til sparsomhet settes direkte i planbølgedomene er implementasjonen effektiv både i å redusere artefakter såvel som å komprimere signalets støtte i tid.

Bruken av høy-oppløselige Radon transformasjoner til planbølgedekomposisjon resulterte i prediksjoner av interne multipler med mindre artefakter og tilsynelatende bedre gjengivelse av bølgeform. Resultater fra adaptiv subtraksjon demonstrerte at dette også gav bedre fjerning og attenuasjon av interne multipler, når en sammenlikner med bruk av standardformulasjoner av lineære Radon transformasjoner. Implementasjon av automatiske regulariseringsprosedyrer for den høy-oppløselige Radon transformasjonen kan potensielt sett gi enda bedre resultater, spesielt for 2D prediksjon i koblet planbølgedomene.

En tidligere foreslått prosedyre for å kunne gjennomføre multidimensjonell prediksjon av interne multipler ved bruk av migrerte datasett har blitt demonstrert til å fungere. Prosedyren vurderes som passende for bruk på reelle feltdata.

Preface

This Master's thesis has been written as a collaboration between the Norwegian University of Science and Technology and Equinor ASA (previously Statoil ASA). The thesis follows as a natural continuation of the project work in Aaker (2017) performed in the course *TPG4570 - Petroleum Geosciences, Specialization Project*.

For the sake of completeness and due to a wish of the supervisor at NTNU, professor Børge Arntsen, the first three chapters include some of the work presented in Aaker (2017). In the introduction to each of the relevant chapters I have carefully disclaimed which parts of the text originate from the previous project work. It is my personal view that the work presented in this thesis, excluding what originates from Aaker (2017), is more than satisfactory to comply with a workload corresponding to 30 ECTS.

Trondheim, June 2018
Ole Edvard Aaker

Acknowledgements

I gratefully acknowledge the support of Equinor ASA to perform this work. I thank the partners Petoro AS, Equinor ASA, Eni Norge AS, Total EP Norge AS and ExxonMobil Exploration and Production Norway AS for permission to use and publish data from the Åsgard license.

I would like to thank Adriana Citlali Ramírez for a very interesting and rewarding cooperation. Your ambition and knowledge have been key factors in realizing this work. The countinuous encouragement that you have extended to me and the drive for improvement that you exhibit have been very important for lifting the quality of this academic work.

I extend my thanks to professor Børge Arntsen whom has been my supervisor at NTNU. You have motivated me to master challenging, but highly rewarding, skills within geophysics and beyond. Discussions with you are always entertaining.

I would like to thank Lill-Tove Wetjen Sigernes. During my work for Equinor you have placed confidence in me from day one. The two internships and the collaboration on the project work and master's thesis have all been very valuable experiences for me. I hope that you will continue to give other students the same opportunities.

Emin Sadikhov and Shruti Gupta have been key contributors to this thesis, thank you for the pleasant cooperation. I would also like to thank Anh Kiet Nguyen for fruitful discussions.

Anne-Elisabeth and Harald, my parents, deserve special thanks for their constant support.

During these five years of study I have been fortunate to make friendships with many persons whom are all special to me. I hope that our paths may cross frequently.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

Disclaimer: Section 1.2, *History of the inverse scattering series*, in this chapter is directly derived from the work in Aaker (2017) performed in the course *TPG4570 - Petroleum Geosciences, Specialization Project*.

1.1 On internal multiples

Internal multiples constitute a complex, higher order scattering phenomenon. In contrast to so-called primary reflections, internal multiples are defined as the subpart of the wavefield which has experienced more than one reflection-like scattering within the subsurface.

Industry standard wavefield redatuming and imaging algorithms do not treat internal multiples correctly. If they are considerably present in the recorded seismic data, they may ’cross-talk’ with primary reflections in the imaging phase. This has the potential to cause significant distortions in the resulting seismic images.

One way to prevent the unwanted effects related to internal multiples is to predict and remove the multiples prior to performing the imaging procedure. This particular part of the wavefield is however highly sensitive to medium heterogenities and travel along complex wavepaths. In practice, the medium of interest is usually not known to the specifications and resolution needed to faithfully reproduce internal multiples in a model-driven fashion.

Data driven prediction algorithms are therefore exceedingly interesting for use in internal multiple prediction and removal. The Inverse Scattering Series (abbr. ISS) (Weglein et al., 1997) provides the ability to exactly reproduce the kinematics and in an approximate sense recover the amplitudes. The method requires virtually no knowledge of the medium and in general avoids many assumptions and restrictions imposed on the subsurface.

1.2 History of the inverse scattering series

The origin of the Inverse Scattering Series traces back to a result obtained by Moses (1956). Razavy (1975) applied the latter work to the problem of determining the velocity of acoustic waves from reflection data originating in a 1D medium. These works were transformed for application in a multi-dimensional earth by Weglein et al. (1981) and Stolt and Jacobs (1980). Carvalho (1992) presented a way to use the inverse scattering series to remove free-surface multiples in marine seismic data. Matson and Weglein (1996) presented algorithms for free-surface multiple removal in an elastic formulation. Weglein et al. (1997) gave a thorough description of an ISS derived algorithm for internal multiple prediction. This was based on the earlier works of Araújo (1994) and Araújo et al. (1994). A review paper on the ISS in the context of seismic exploration is found in Weglein et al. (2003).

Ramírez (2007) presented a rigorous amplitude and phase analysis of the inverse scattering derived internal multiple predictor. Furthermore, it included a detailed derivation of the algorithm. The first mapping of the internal multiple predictor from the wavenumber-pseudodepth domain to the coupled plane wave domain was given in Coates and Weglein (1996). Nita and Weglein (2009) gave an in-depth analysis of the relation between the two domains. The latter article demonstrated that the psuedo-depth monotonicity condition translates to an intercept time monotonicity condition in the coupled plane wave domain. In practice, the ISS prediction algorithms in the pseudo-depth wavenumber formulation typically suffer from known artifacts (Sun and Innanen, 2015). The results by Sun and Innanen (2015) seemed to indicate that the plane wave domain might yield an improved setting for performing internal multiple prediction. Along with the analysis and results in Sun and Innanen (2016), this has lead to a renewed interest in ISS internal multiple prediction in the plane wave domain.

Chapter 2

Theory of Internal Multiple prediction from the Inverse Scattering series.

Disclaimer: The work presented in this chapter is mainly derived from the work in Aaker (2017) performed in the course *TPG4570 - Petroleum Geosciences, Specialization Project*. Beyond a general correction of errata, section 2.6 has seen extensive re-work, while section 2.9 and beyond also feature certain modifications.

2.1 Scattering theory and nomenclature.

The theory and nomenclature that follows here is largely derived from the review paper Weglein et al. (2003).

Scattering theory is a form of perturbation analysis that describes how perturbations in medium properties relate to perturbations in wavefields that experience said medium. Often, one considers an original, unperturbed medium as a reference medium, and a perturbed medium to be the actual medium of interest. In this notation, the differences between the actual and reference media upon wavefields passing through them is encoded in the perturbation operator. In scattering theory, one distinguishes between two different tasks/models, namely forward and inverse scattering:

- **Forward scattering** is the process that outputs the actual wavefield given the properties of the reference medium, the reference wavefield and the perturbation operator.
- **Inverse scattering** in turn, outputs the perturbation operator, which again encodes the difference between actual and reference medium, when given the reference

> medium, the reference wavefield and measured values of the actual wavefield. In the setting of exploration seismology, the measured values of the actual wavefield are acquired through the seismic acquisition process.

In order to be able to define and exemplify the necessary quantities, this paper will begin with a brief mathematical description of the differential equations that govern wave propagation in media relevant for exploration seismology. The wavefield that is the impulse response, or Green's function, of an actual medium can be notationally described in the space-frequency $(\mathbf{x}, \omega)$ domain as:

$$\boldsymbol{\mathcal{L}}(\mathbf{x}, \omega)\mathbf{G}(\mathbf{x}, \mathbf{x}', \omega) = -\mathbf{I}\delta(\mathbf{x} - \mathbf{x}') \tag{2.1}$$

The location of the impulsive source is at position $\mathbf{x}'$. $\mathbf{I}$ denotes the identity matrix, or unit operator. $\mathbf{G}$ is the Greens' matrix containing Green's function entries. These entries depend on the physical problem at hand as well as the chosen parametrization. $\boldsymbol{\mathcal{L}}(\mathbf{x}, \omega)$ is the differential operator matrix describing wave propagation in this medium. For the problems studied here, $\boldsymbol{\mathcal{L}}$ is a linear differential operator. The coordinate vector is defined as $\mathbf{x} := (x, y, z)$ with horizontal components $\mathbf{x}_H := (x, y)$. The sign convention chosen at the right-hand side of equation (2.1) is arbitrary, and may be chosen otherwise[1].

Similarly, one can define the Green's function in a reference medium, $\mathbf{G}_0$, where wave propagation is described by the reference differential operator $\boldsymbol{\mathcal{L}}_0$:

$$\boldsymbol{\mathcal{L}}_0(\mathbf{x}, \omega)\mathbf{G}_0(\mathbf{x}, \mathbf{x}', \omega) = -\mathbf{I}\delta(\mathbf{x} - \mathbf{x}') \tag{2.2}$$

With these definitions in hand, the perturbation operator $\boldsymbol{\mathcal{V}}$ is defined:

$$\boldsymbol{\mathcal{V}} := \boldsymbol{\mathcal{L}} - \boldsymbol{\mathcal{L}}_0 \tag{2.3}$$

The scattered field is defined as the difference between the actual and reference Green's functions:

$$\boldsymbol{\Psi}_s := \mathbf{G} - \mathbf{G}_0 \tag{2.4}$$

Note that even though $\boldsymbol{\Psi}_s$ is an arithmetic difference between two Green's functions, it is not itself a Green's function[2].

Clearly, the actual Green's function can be described as the sum of a scattered wavefield and a reference Green's wavefield:

$$\mathbf{G} = \boldsymbol{\Psi}_s + \mathbf{G}_0 \tag{2.5}$$

Example of operators and Green's functions in acoustic media

In arbitrarly inhomogenous acoustic media, where we parametrize a Green's function

[1] I.e., the properties of the Green's functions are independent of the sign of the source function.

[2] I.e, the scattered parts of the Green's wavefield does not satisfy the differential equation given in eq. (2.1).

purely in terms of acoustic pressure (rather, deviations from a reference pressure) the matrices containing the Green's functions $\mathbf{G}$ and $\mathbf{G}_0$ can be reduced to scalars. In further discussion, the one scalar entry of the Green's matrix $\mathbf{G}_{(0)}$ is denoted $G_{(0)}$. The impulsive sources are in the following notation of the volume injection type.
The differential operators of the acoustic wave equation can then be written as follows:

$$\boldsymbol{\mathcal{L}}(\mathbf{x},\omega) = \mathcal{L}(\mathbf{x},\omega) = \omega^2\kappa(\mathbf{x}) + \partial_i \frac{1}{\rho(\mathbf{x})}\partial_i \cdot \tag{2.6}$$

$$\boldsymbol{\mathcal{L}}_0(\mathbf{x},\omega) = \mathcal{L}_0(\mathbf{x},\omega) = \omega^2\kappa_0(\mathbf{x}) + \partial_i \frac{1}{\rho_0(\mathbf{x})}\partial_i \cdot \tag{2.7}$$

In the actual and reference media respectively, described by the properties $\kappa_{(0)}(\mathbf{x})$ and $\rho_{(0)}(\mathbf{x})$ corresponding to adiabatic incompressibility and volumetric density. Einstein summation notation is implied on repeated indices. Latin indices take on the values $i \in \{1,2,3\}$ whereas Greek indices take on the values $\alpha \in \{1,2\}$.

In an acoustic medium with actual and reference differential operators in equations (2.6) and (2.7), by the definition of the perturbation operator in equation (2.3), the acoustic perturbation operator becomes:

$$\mathcal{V}(\mathbf{x},\omega) = \omega^2\{\kappa(\mathbf{x}) - \kappa_0(\mathbf{x})\} + \partial_i\Big[(\frac{1}{\rho(\mathbf{x})} - \frac{1}{\rho_0(\mathbf{x})})\partial_i\ \cdot\Big] \tag{2.8}$$

The perturbation operator may only be nonzero in areas $\mathbf{x}$ where the reference and actual media differ in properties.

In media with properties beyond the acoustic case, e.g. elastic, poroelastic and seismoelectric, $\mathbf{G}$, $\boldsymbol{\mathcal{L}}$ and $\boldsymbol{\mathcal{V}}$ are all matrices. Matrix notation thus includes a more general result.

2.2 Forward and inverse scattering series

The scattered wavefield and the two Green's functions in equation (2.4) are related through the operator identity named the Lippmann-Schwinger equation, which reads in the space-frequency domain (Weglein et al., 2003):

$$\boldsymbol{\Psi}_s = \mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G} \tag{2.9}$$

Although usually known specifically from quantum mechanics, the Lippmann-Schwinger integral equation is elegantly derivable from the two-way reciprocity theorems of convolution type (Vasconcelos et al., 2009). Relation (2.9) can be used to generate the forward scattering series in terms of the reference wavefield $\mathbf{G_0}$ and the perturbation operator $\boldsymbol{\mathcal{V}}$. In order to accomplish this, substitute the representation of the actual wavefield $\mathbf{G}$ given in equation (2.5):

$$\boldsymbol{\Psi}_s = \mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0 + \mathbf{G}_0\boldsymbol{\mathcal{V}}\boldsymbol{\Psi}_s \tag{2.10}$$

As such, by substituting equation (2.9) into itself repeatedly and making use of the representation of $\mathbf{G}$, this generates an infinite series in terms of increasing scattering order. The resulting series is termed the forward-scattering, Born or Neumann series.

$$\begin{aligned} \boldsymbol{\Psi}_s &= \mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0 \\ &+\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0 + \mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0 + \cdots + \end{aligned} \tag{2.11}$$

Note that equation (2.11) is written in operator notation. In order to account for all scattering phenomena of a wavefield excited at $\mathbf{x}_s$ and observed at $\mathbf{x}$, the series is written as an integral formulation over all possible locations (of scattering) in the medium of interest:

$$\begin{aligned} \boldsymbol{\Psi}_s(\mathbf{x}, \mathbf{x}_s, \omega) &= \int \mathbf{G}_0(\mathbf{x}, \mathbf{x}_1, \omega)\boldsymbol{\mathcal{V}}(\mathbf{x}_1, \omega)\mathbf{G}_0(\mathbf{x}_1, \mathbf{x}_s, \omega) \\ \times d\mathbf{x}_1 &+ \int \mathbf{G}_0(\mathbf{x}, \mathbf{x}_1, \omega)\boldsymbol{\mathcal{V}}(\mathbf{x}_1, \omega)\mathbf{G}_0(\mathbf{x}_1, \mathbf{x}_2, \omega)\boldsymbol{\mathcal{V}}(\mathbf{x}_2, \omega) \\ &\times \mathbf{G}_0(\mathbf{x}_2, \mathbf{x}_s, \omega) d\mathbf{x}_1 d\mathbf{x}_2 + \cdots + \end{aligned} \tag{2.12}$$

Each term in equation (2.12) is to be interpreted as a sequence of propagations in the reference medium, given by $\mathbf{G}_0$ and scattering from points where actual and reference medium differ, encoded in $\boldsymbol{\mathcal{V}}$.

Rewriting the scattered wavefield explicitly in terms of the order of $\boldsymbol{\mathcal{V}}$ (in operator notation):

$$\boldsymbol{\Psi}_s = (\boldsymbol{\Psi}_s)_1 + (\boldsymbol{\Psi}_s)_2 + (\boldsymbol{\Psi}_s)_3 + \cdots + \tag{2.13}$$

By this notation, the term

$$(\boldsymbol{\Psi}_s)_n := \mathbf{G}_0(\boldsymbol{\mathcal{V}}\mathbf{G}_0)^n \tag{2.14}$$

is $n-$th order in the perturbation operator $\boldsymbol{\mathcal{V}}$.

In the following discussion, the data considered correspond to the marine experiment without a free surface (equivalently without its effects[3]), after subtraction of the reference field and source signature deconvolution. The data, $\boldsymbol{\mathcal{D}}$, can therefore be considered to be the measured values of the scattered wavefield:

$$\boldsymbol{\Psi}_s^m = \boldsymbol{\mathcal{D}} \tag{2.15}$$

Because the scattered wavefield could be written in terms of order in $\boldsymbol{\mathcal{V}}$, eq. (2.13), so can the data (Weglein et al., 1981):

$$\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}_1 + \boldsymbol{\mathcal{D}}_2 + \cdots + = \sum_{i=0}^{\infty} \boldsymbol{\mathcal{D}}_i \tag{2.16}$$

[3] I.e. after common processing procedures such as Surface Related Multiple Eliminiation (SRME) and source and receiver deghosting.

This implies, through the following geometric series argument, that $\boldsymbol{\mathcal{V}}$ can be written as a power series of terms $\boldsymbol{\mathcal{V}}_i$ that are i-th order in (all the) data (Weglein et al., 1997).

Consider the following forward series:

$$x = a\sum_{n=1}^{\infty} y^n = \frac{ay}{1-y} \tag{2.17}$$

This series, has a corresponding inverse series, representing y as powers of x:

$$y = \frac{x}{x+a} = \sum_{n=1}^{\infty} \cos[(n-1)\pi](\frac{x}{a})^n \tag{2.18}$$

Considering $\boldsymbol{\mathcal{D}}$ as an analog to x, then $\boldsymbol{\mathcal{V}}$ is analog to y. This provides the justification for the expansion of $\boldsymbol{\mathcal{V}}$ as a geometric series in $\boldsymbol{\mathcal{D}}$. We expand $\boldsymbol{\mathcal{V}}$ as previously declared:

$$\boldsymbol{\mathcal{V}} = \boldsymbol{\mathcal{V}}_1 + \boldsymbol{\mathcal{V}}_2 + \boldsymbol{\mathcal{V}}_3 + \cdots + \tag{2.19}$$

Inserting equation (2.19) and relation (2.15) into the Born series, eq. (2.11), yields:

$$\begin{aligned} \boldsymbol{\mathcal{D}} &= \mathbf{G}_0(\boldsymbol{\mathcal{V}}_1 + \boldsymbol{\mathcal{V}}_2 + \boldsymbol{\mathcal{V}}_3 + \cdots +)\mathbf{G}_0 \\ &+\mathbf{G}_0(\boldsymbol{\mathcal{V}}_1 + \cdots +)\mathbf{G}_0(\boldsymbol{\mathcal{V}}_1 + \cdots +)\mathbf{G}_0 + \cdots + \end{aligned} \tag{2.20}$$

By equating the terms on the left- and righthandside of this equation in terms of their order in the data $\boldsymbol{\mathcal{D}}$, one gets:

$$\boldsymbol{\mathcal{D}} = \mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0 \tag{2.21}$$

$$0 = \mathbf{G}_0\boldsymbol{\mathcal{V}}_2\mathbf{G}_0 + \mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0 \tag{2.22}$$

$$\begin{aligned} 0 &= \mathbf{G}_0\boldsymbol{\mathcal{V}}_3\mathbf{G}_0 + \mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_2\mathbf{G}_0 + \mathbf{G}_0\boldsymbol{\mathcal{V}}_2\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0 \\ &+\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0 \end{aligned} \tag{2.23}$$

Equations for higher order terms follow the exact same structure. Equations (2.21) through (2.23) and the infinitely many following equations constitute what is named the **inverse scattering series** (Weglein et al., 1981). By first solving for the component of $\boldsymbol{\mathcal{V}}$ that is linear in the data, $\boldsymbol{\mathcal{V}}_1$, by equation (2.21), one can then determine $\boldsymbol{\mathcal{V}}_2$ by equation (2.22) and so forth. Hence, the inverse scattering series allows one to sequentially build the subseries that constitutes the true reflectivity $\boldsymbol{\mathcal{V}}$ of the perturbed medium. The only assumption made so far is that $\boldsymbol{\mathcal{V}}_1$ truly is the portion of $\boldsymbol{\mathcal{V}}$ that is linear in the data[4].

[4] Note: In the formalism of the inverse scattering series the inverse Born approximation is never invoked.

2.3 Leading order primary and internal multiple generators in the forward series

As primaries per definition only have one upward reflection, the leading order contribution for primary reflections is the first term in the forward-scattering series: $\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0$ (Matson and Weglein, 1996). However, do note that all primaries are constructed in the forward series by portions of every term in the series. Extra terms in $\boldsymbol{\mathcal{V}}$ can describe e.g. extra transmission, and self-interaction on the path between source and receiver.

As first order internal multiples have three factors of reflection-like scattering, the leading order contribution in the forward modelling of the internal multiples comes from the term with three factors of the perturbation operator: $\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0$ (Matson and Weglein, 1996). In general, $n-$th order internal multiples have contributions from all terms starting at term $2n+1$, where the term $2n+1$ itself is the leading order contribution.

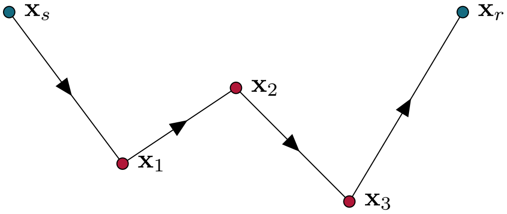
However, as stated, all primaries are constructed in the forward series by portions of every term in the series (Weglein et al., 2003). The term $\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0$ therefore contributes both to generate primaries and first order internal multiples. A separation of this term into the part that solely can create the first-order internal multiple is needed. For this we follow Weglein et al. (2003), invoking a geometrical argument that works well for internal multiple generation in many geological settings. The scattering described in the term $\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0\boldsymbol{\mathcal{V}}\mathbf{G}_0$ occurs at three distinct subsurface locations, $\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3$, respectively. In the sense of forward modelling, many first-order internal multiples will satisfy a scattering-like reflection pattern given by: one upward reflection at $\mathbf{x}_1$ followed by one downward at $\mathbf{x}_2$, and finally one upward at $\mathbf{x}_3$. For this to be possible, the true depth of the second reflector/scatterer must be shallower than that of the other two. Hence, we end up at the condition:

$$\begin{aligned} z_1 &> z_2 \\ z_3 &> z_2 \end{aligned} \tag{2.24}$$

This is termed the Lower-Higher-Lower condition, abbreviated LHL. A Feynman-type diagram for a first order internal multiple satisfying the Lower-Higher-Lower condition is shown in figure 2.1.

Ramirez and Otnes (2008) demonstrated that each order of approximation within the forward series provides the correct wave type. For two-parameter (velocity and density) perturbations the amplitude and traveltime is incorrect at each order contribution. Furthermore, they showed that the forward series requires an infinite number of terms to be corrected. In order to yield a prediction procedure of practical value, the internal multiple prediction sought is instead often from the inverse series. The LHL condition in the forward series, in true depth, gives a hint as to which terms contribute in the sense of the inverse series.

Figure 2.1 First order internal multiple satisfying the lower-higher-lower relationship.



2.4 Linking forward series multiple generators to the inverse series mutiple attenuator

The internal multiple predictor studied in this paper does not originate from the forward series, but rather from the inverse series. The clue to realizing an internal multiple attenuator from the inverse series lies in the hint given in the forward series. Understanding how the forward series creates an event can give information on where the inverse process might be located in the inverse series (Weglein et al., 2003). This is virtue of a symmetry between the two series. The first term in the forward series that constructs the leading order term in the internal multiple and the corresponding first term in the inverse series that predicts its eliminator of a given order are only approximate. However, as noted by Weglein et al. (2003), the efficiency of the first term in the removal subseries is remarkably higher than the first term in the forward creation. The internal multiple attenuator predicts the time precisely and gives an approximate amplitude prediction (Weglein et al., 2003). The efficiency of the first internal multiple subterm of the inverse scattering series is what accounts for its practical value.

In the forward series, the leading order contribution to the first order internal multiple lies in the third order scattering term. Using the link between the two series, one searches for the contributing subterms in the third order term of the inverse scattering series, equation (2.23). As shown by Araújo (1994), the two terms $\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_2\mathbf{G}_0$ and $\mathbf{G}_0\boldsymbol{\mathcal{V}}_2\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0$ always contain a refraction-like scattering component. They do not contribute to the internal multiple predictor (Weglein et al., 2003). The resulting internal multiple predictor derived from the Inverse Scattering Series is in operator notation $\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0\boldsymbol{\mathcal{V}}_1\mathbf{G}_0$ (Weglein et al., 2003). In integral formulation the internal multiple predictor is expressed as:

$$(\mathbf{G}_0\boldsymbol{\mathcal{V}}_3\mathbf{G}_0)^{IM} = \mathbf{d}_3^{IM}(\mathbf{x}_r,\mathbf{x}_s,\omega) := -\int_{\substack{z_1>z_2\\ z_3>z_2}} \mathbf{G}_0(\mathbf{x}_1,\mathbf{x_s})\boldsymbol{\mathcal{V}}_1(\mathbf{x}_1)\mathbf{G}_0(\mathbf{x}_2,\mathbf{x_1}) \times\boldsymbol{\mathcal{V}}_1(\mathbf{x}_2)\mathbf{G}_0(\mathbf{x}_3,\mathbf{x_2})\boldsymbol{\mathcal{V}}_1(\mathbf{x}_3)\mathbf{G}_0(\mathbf{x}_r,\mathbf{x_3})d\mathbf{x}_1d\mathbf{x}_2d\mathbf{x}_3 \tag{2.25}$$

The internal multiple predictor in (2.25) satisfies an LHL condition in pseudodepth[5]. The assumption is that the ordering of the actual and the pseudo depths of two sub-events is

[5] Note: this is in contrast to the forward series LHL condition, which is given in true depth.

preserved (Nita and Weglein, 2009). This is termed the pseudodepth monotonicity condition. Note however that attempting to perform internal multiple prediction directly via equation (2.25) does not actually yield a correct prediction. Rather, the equation needs to be re-written in terms of a quantity that in every dimension would correspond to a scattered field with spike-like events (Weglein et al., 2003). For this, $\mathcal{V}_1$ is re-written in terms of the effective data, $\mathbf{b}_1$. This derivation will be provided in section 2.7. Until we have actually realized a proper internal multiple prediction algorithm by studying and re-writing the terms in (2.25), we will still refer to the latter as the internal multiple predictor, although this is not strictly true.

2.5 A short note on the first order perturbation operator

The first order perturbation operator $\mathcal{V}_1$ is present within the leading order internal multiple predictor. It is briefly studied here in order to investigate why the predictor is purely data-driven.

Under the assumption that $\mathbf{G_0}$ is invertible, i.e.:

$$\mathbf{G_0}^{-1}\mathbf{G_0} = \mathbf{I} \tag{2.26}$$

The solution of equation (2.21) for $\mathcal{V}_1$ is written:

$$\mathcal{V}_1 = \mathbf{G_0}^{-1}\mathcal{D}\mathbf{G_0}^{-1} \tag{2.27}$$

Equation (2.27) represents a migration performed using the Green's functions, and therefore implicitly the medium properties, of the reference medium. The mapping of the data to pseudodepth is provided by this migration. As a strict discrepancy between common migrations used in exploration geophysics, the properties of the reference medium migration in eq. (2.27) are never required to be close to that of the actual medium. Therefore, $\mathcal{V}_1$ should not be seen as close to the true reflectivity $\mathcal{V}$. In other words, the inverse Born approximation is never invoked in the formalism of the inverse scattering series.

By representing $\mathcal{D}$ using the Lippmann-Schwinger equation, equation (2.9), this solution for the first order scattering term becomes:

$$\begin{aligned}\mathcal{V}_1 = \mathbf{G_0}^{-1}(&\mathrm{G}_0\mathcal{V}\mathrm{G}_0 + \mathrm{G}_0\mathcal{V}\mathrm{G}_0\mathcal{V}\mathrm{G}_0 + \\ &\mathrm{G}_0\mathcal{V}\mathrm{G}_0\mathcal{V}\mathrm{G}_0\mathcal{V}\mathrm{G}_0 + \ldots +)\mathbf{G_0}^{-1}\end{aligned} \tag{2.28}$$

Invoking the assumption of invertibility of the reference Green's matrix:

$$\mathcal{V}_1 = \mathcal{V} + \mathcal{V}\mathrm{G}_0\mathcal{V} + \mathcal{V}\mathrm{G}_0\mathcal{V}\mathrm{G}_0\mathcal{V} + \ldots + \tag{2.29}$$

Evidently, $\mathcal{V}_1$ contains the information of the true reflectivity of the medium $\mathcal{V}$. The additional terms $(\mathcal{V}\mathrm{G}_0\mathcal{V} + \mathcal{V}\mathrm{G}_0\mathcal{V}\mathrm{G}_0\mathcal{V} + \ldots +)$ are however not treated correctly. The internal multiple prediction given by (2.25) can be considered a demigration procedure that generates a subset of the data, namely the main contribution to first-order internal multiples, given the first-order reflectivity and the Green's function(s) in the reference medium.

The process of first solving equation (2.27) and then using it to predict first-order internal multiples by a properly re-expressed version equation (2.25)[6], can as such be referred to as a migration-demigration procedure, as is the way Verschuur (2006, p. 183) describes it.

This description of the internal multiple prediction algorithm aids in the interpretation as to why it is indeed data driven, as opposed to model driven. The parameters used in the migration, e.g. velocity of the reference medium, are not of critical importance, because the errors commited in this stage is balanced by the demigration procedure.

2.6 Re-representing the internal multiple predictor in terms of three subevents

Two of the Green's function terms in equation (2.25) can be re-written using representation integrals, so that each of them could have been emitted or recorded on the acquisition surface. For this, we will consider an acoustic reference medium. As such, the matrices in equation (2.25) reduce to scalars.

Consider the Kirchhoff-Helmholtz integral[7] for two states, A and B, of Green's functions satisfying the constant-density Helmholtz equation:

$$G(\mathbf{x}_A, \mathbf{x}_B) = \int_{\partial\mathbb{D}} \{G^*(\mathbf{x}, \mathbf{x}_A)\partial_i G(\mathbf{x}, \mathbf{x}_B) - (\partial_i G^*(\mathbf{x}, \mathbf{x}_A))G(\mathbf{x}, \mathbf{x}_B)\} n_i d^2\mathbf{x} \quad (2.30)$$

The outlines of the domain $\mathbb{D}$ of interest is shown in figure 2.2. $\mathbb{D}$ is a proper subset of the reference medium, which is here chosen to be characterized only by the waterspeed velocity, i.e it is homogenous everywhere. $\partial\mathbb{D}_0$ and $\partial\mathbb{D}_i$ represent the upper and lower horizontal boundaries. They are defined at pseudodepths z_0 and z_i, respectively, and have normal vectors parallel to the z axis. The upper boundary $\partial\mathbb{D}_0$ represents the aquisition surface, while the boundary $\partial\mathbb{D}_i$ is chosen at convenience. For selection of $\partial\mathbb{D}_i$ we will make use of the Lower-Higher-Lower relation in pseudodepth in order to achieve an optimal single-sided representation. S represents the surface of the cylindrical part and is parallel to the z axis. Its area grows proportional to $\propto r_H$. The far-field response[8] of the terms involved in the integral have amplitudes given by $\propto \frac{1}{r_H}$ (Fokkema and van den Berg, 2013). Their products therefore have amplitudes proportional to $\frac{1}{r_H^2}$. For the present choice of the domain $\mathbb{D}$, the contribution over S therefore vanishes.

[6] That is, by the proper formulas derived and showed in sections 2.7 and 2.8, respectively.

[7] The Kirchhoff-Helmholtz integral is properly introduced and derived in appendix A.

[8] Recall that the evaluation on this boundary is performed in the limit $\mathbf{x}_H \rightarrow \pm\infty$

Figure 2.2 Cylindrical domain of interest, $\mathbb{D}$, with non-overlapping boundaries $\partial\mathbb{D} = \partial\mathbb{D}_0 \cup \partial\mathbb{D}_i \cup S$.

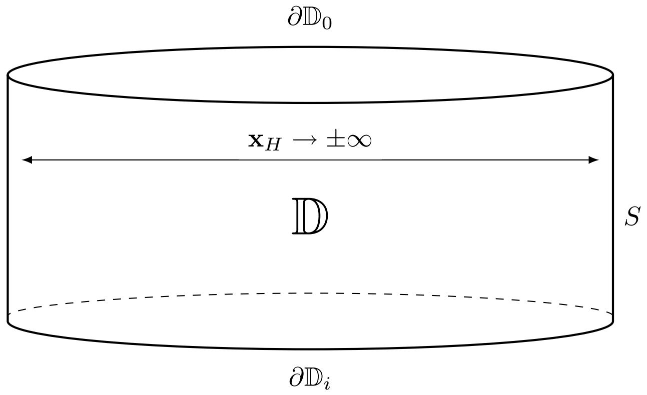


Figure 2.3 Configuration for re-representation of $G_0(\mathbf{x}_2, \mathbf{x}_1, \omega)$.

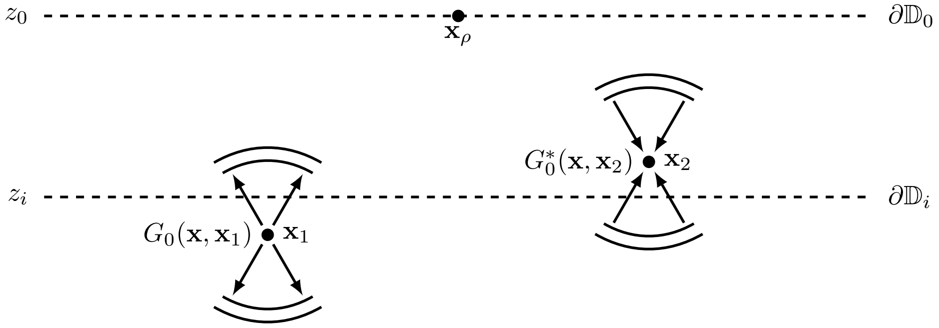
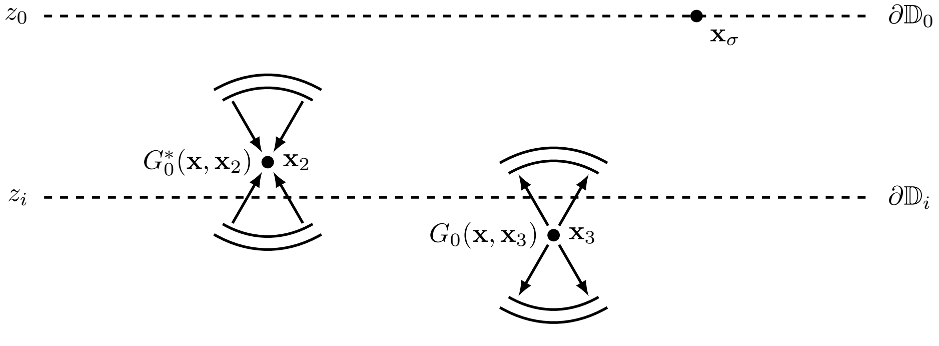


Figure 2.4 Configuration for re-representation of $G_0(\mathbf{x}_3, \mathbf{x}_2, \omega)$, through is reciprocal quantity, $G_0(\mathbf{x}_2, \mathbf{x}_3, \omega)$.



	State A	**State B**
Wavefield	$G_0(\mathbf{x}, \mathbf{x}_1)$	$G_0^*(\mathbf{x}, \mathbf{x}_2)$
Source $s(\mathbf{x}, \omega)$	$\delta(\mathbf{x} - \mathbf{x}_1) \quad \mathbf{x}_1 \notin \mathbb{D}$	$\delta(\mathbf{x} - \mathbf{x}_2) \quad \mathbf{x}_2 \in \mathbb{D}$
Directivity at $\partial\mathbb{D}_0$	Purely outgoing	Purely ingoing
Directivity at $\partial\mathbb{D}_i$	Purely ingoing	Purely ingoing

Table 2.1: Configuration for re-representation of $G_0(\mathbf{x}_2, \mathbf{x}_1, \omega)$

	State A	**State B**
Wavefield	$G_0(\mathbf{x}, \mathbf{x}_3)$	$G_0^*(\mathbf{x}, \mathbf{x}_2)$
Source $s(\mathbf{x}, \omega)$	$\delta(\mathbf{x} - \mathbf{x}_3) \quad \mathbf{x}_3 \notin \mathbb{D}$	$\delta(\mathbf{x} - \mathbf{x}_2) \quad \mathbf{x}_2 \in \mathbb{D}$
Directivity at $\partial\mathbb{D}_0$	Purely outgoing	Purely ingoing
Directivity at $\partial\mathbb{D}_i$	Purely ingoing	Purely ingoing

Table 2.2: Configuration for re-representation of $G_0(\mathbf{x}_3, \mathbf{x}_2, \omega)$ through is reciprocal quantity, $G_0(\mathbf{x}_2, \mathbf{x}_3, \omega)$.

Case I: Representation of $G_0(\mathbf{x}_2, \mathbf{x}_1, \omega)$

The configuration for re-presentation of the term $G_0(\mathbf{x}_2, \mathbf{x}_1, \omega)$ is shown graphically in figure 2.3, and the tabulated format is given in table 2.1. We will consistently use the fact that at the boundaries, only terms that propagate in opposite directions contribute to the boundary integrals (Wapenaar, 2014). Note that because the reference medium is globally homogenous, no scattering can take place in or outside $\mathbb{D}$. The wavefield directivities at the boundaries are therefore extraordinarily simple. Furthermore, we will make use of a far-field approximation of the partial derivatives of the Green's functions, namely we will replace them with $\partial_i G n_i \approx \mp\frac{j\omega}{c}G$ (Wapenaar and Fokkema, 2006). The minus sign denotes an ingoing wave at the boundary with normal vector n_i, while the plus sign denotes an outgoing wave at the same boundary.

Because of these properties we achieve the representation for $G_0(\mathbf{x}_2, \mathbf{x}_1, \omega)$:

$$G_0(\mathbf{x}_2, \mathbf{x}_1, \omega) \approx -\frac{2j\omega}{c_0} \int_{\partial\mathbb{D}_0} G_0(\mathbf{x}_\rho, \mathbf{x}_1) G_0^*(\mathbf{x}_\rho, \mathbf{x}_2) d^2\mathbf{x}_{\rho,H} \tag{2.31}$$

It is evident from the integral representation that $\mathbf{x}_\rho$ represents an auxiliary receiver coordinate. The acquisition boundary varies only in the horizontal coordinates, hence the integration is over $\mathbf{x}_{\rho,H}$ evaluated at $z_\rho = z_0$.

Case II: Representation of $G_0(\mathbf{x}_2, \mathbf{x}_3, \omega)$

Idem, the configuration for re-presentation of the term $G_0(\mathbf{x}_3, \mathbf{x}_2, \omega)$ is shown graphically in figure 2.4 and the tabulated format is given in table 2.2. We will achieve the representation through the reciprocal term $G_0(\mathbf{x}_2, \mathbf{x}_3, \omega)$.

Inserting the properties from table 2.2 yields the formula:

$$G_0(\mathbf{x}_2, \mathbf{x}_3, \omega) \approx -\frac{2j\omega}{c_0} \int_{\partial\mathbb{D}_0} G_0(\mathbf{x}_\sigma, \mathbf{x}_3) G_0^*(\mathbf{x}_\sigma, \mathbf{x}_2) d^2\mathbf{x}_{\sigma,H} \tag{2.32}$$

We make use of acoustic source-receiver reciprocity (Wapenaar, 2014) for all of the three terms involved, in one step, in order to achieve:

$$G_0(\mathbf{x}_3, \mathbf{x}_2, \omega) \approx -\frac{2j\omega}{c_0} \int_{\partial\mathbb{D}_0} G_0(\mathbf{x}_3, \mathbf{x}_\sigma) G_0^*(\mathbf{x}_2, \mathbf{x}_\sigma) d^2\mathbf{x}_{\sigma,H} \tag{2.33}$$

It is evident from the integral representation that $\mathbf{x}_\sigma$ represents an auxiliary source coordinate. Similarly as for the representation in equation (2.31), the integration is over $\mathbf{x}_{\sigma,H}$ evaluated at $z_\sigma = z_0$.

Substitution into the internal multiple predictor

Substituting the Green's function representations in equations (2.31) and (2.33) into the internal multiple predictor, eq. (2.25), and re-arranging the involved terms:

$$\begin{aligned} d_3^{IM}(\mathbf{x}_r, \mathbf{x}_s, \omega) \approx (\frac{2j\omega}{c_0})^2 \int_{\substack{z_1 > z_2 \\ z_3 > z_2}} & [G_0(\mathbf{x}_1, \mathbf{x}_s) \mathcal{V}_1(\mathbf{x}_1) G_0(\mathbf{x}_\rho, \mathbf{x}_1)] \\ & \times [G_0(\mathbf{x}_2, \mathbf{x}_\sigma) \mathcal{V}_1(\mathbf{x}_2) G_0(\mathbf{x}_\rho, \mathbf{x}_2)]^* \\ & \times [G_0(\mathbf{x}_3, \mathbf{x}_\sigma) \mathcal{V}_1(\mathbf{x}_3) G_0(\mathbf{x}_r, \mathbf{x}_3)] \quad d\mathbf{x}_1 d\mathbf{x}_2 d\mathbf{x}_3 d^2\mathbf{x}_{\rho,H} d^2\mathbf{x}_{\sigma,H} \end{aligned} \tag{2.34}$$

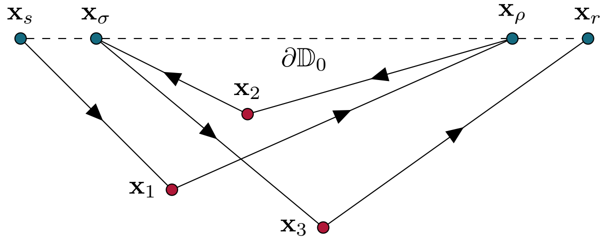
We have used the fact that for cases of lossless true and reference media, $\mathcal{V}_1$ must be a self-adjoint operator. Furthermore, note that $\mathcal{V}_1$ is a scalar not a differential operator. The terms involving Green's functions can therefore be moved around. The representation in equation (2.34) reveals that the internal multiple predictor is constructed from three individual subevents, isolated in the brackets, each emitted and recorded at the acquisition surface. The constant factor outside the integral is real-valued and thus constitutes no phase information. The predicted traveltime of the internal multiple is thus the sum of the traveltimes of the three subevents. The second term is an anticausal, i.e. backwards propagating, term. This leads to the predicted traveltime:

$$\mathcal{T}^{IM} = \mathcal{T}_1 + (-\mathcal{T}_2) + \mathcal{T}_3 = \mathcal{T}_1 - \mathcal{T}_2 + \mathcal{T}_3 \tag{2.35}$$

The three subevents are graphically depicted in figure 2.5.

Again we stress that Weglein et al. (2003, p. R60) do not use $\mathcal{V}_1$ in the internal multiple predictor. They state that taking $\mathcal{V}_1$ through this algorithm does not lead to an attenuation algorithm, based upon empirical evaluation. b_1 is instead used, the effective data generated by a single frequency plane-wave incident field (Weglein et al., 2003). Starting from equation (2.34) we will see how to enable this in the following subsection.

Figure 2.5 The three subevents that construct the internal multiple. Scattering points denoted in red, source and receiver locations along acquisition surface $\partial\mathbb{D}_0$ denoted in blue.



2.7 Novel derivation: Internal multiple attenuator due to 1D wave propagation in a 1D medium

Consider the true medium to vary only in the vertical direction, z. The wave propagation considered is also only in the vertical direction. In this reduced dimensionality the boundary integrals over the additional source and receiver coordinate $\mathbf{x}_\sigma$ and $\mathbf{x}_\rho$ in equation (2.34) vanish[9]. Furthermore, ignore the Lower-Higher-Lower relationship in equation (2.34), such that we first consider d_3, containing both primaries and internal multiples. With these substitutions, one gets:

$$d_3(z_r, z_s, \omega) \approx (\frac{2j\omega}{c_0})^2 \int [G_0(z_1, z_s, \omega)\mathcal{V}_1(z_1, \omega)G_0(z_r, z_1, \omega)] \times[G_0(z_2, z_s, \omega)\mathcal{V}_1(z_2, \omega)G_0(z_r, z_2, \omega)]^* \times[G_0(z_3, z_s, \omega)\mathcal{V}_1(z_3, \omega)G_0(z_r, z_3, \omega)] \quad dz_1dz_2dz_3 \quad (2.36)$$

For simplicity, assume in the following that $z_r = z_s = 0$.
The Green's function $G_0(z, z_s, \omega)$ defined in the one-dimensional reference medium obeys the constant-velocity, constant-density Helmholtz equation:

$$(\frac{\partial^2}{\partial z^2} + \frac{\omega^2}{c_0^2})G_0(z, z_s, \omega) = -\delta(z - z_s) \quad (2.37)$$

From now on define the quantity $k = \frac{\omega}{c_0}$. The analytical solution of (2.37) reads:

$$G_0(z, z_s, \omega) = \frac{-e^{jk|z-z_s|}}{2jk} \quad (2.38)$$

[9] A boundary integral in one dimension is just an evaluation of the integrand, and is therefore implicitly considered.

Making such substitutions into equation (2.36) gives:

$$\begin{aligned} d_3(\omega) = (2jk)^2(\frac{1}{2jk})^3\frac{1}{2jk} & \int e^{2jkz_1}\mathcal{V}_1(z_1,\omega) \quad dz_1 \\ \times \frac{1}{-2jk} & \int e^{-2jkz_2}\mathcal{V}_1(z_2,\omega) \ \ dz_2 \\ \times \frac{1}{2jk} & \int e^{2jkz_3}\mathcal{V}_1(z_3,\omega) \quad dz_3 \end{aligned} \tag{2.39}$$

Following Ramírez (2007), we identify, due to the contribution of Razavy (1975):

$$\mathcal{V}_1(k,k,\omega) = \int e^{2jkz}\mathcal{V}_1(z,\omega) \quad dz \tag{2.40}$$

Furthermore, the definition of the effective data b_1, which represents the effective data generated by a monochromatic plane-wave incident field (Weglein et al., 2003), is (Ramírez, 2007):

$$b_1(k_z) = \frac{\mathcal{V}_1(k,k,\omega)}{2jk} = d(\omega)2jk \tag{2.41}$$

Furthermore, define the Fourier conjugate to pseudodepth: $k_z := 2k$. Equation (2.39) then reads:

$$d_3(\omega) = (\frac{1}{2jk})b_1(k_z)b_1(-k_z)b_1(k_z) \tag{2.42}$$

Consider an effective-data term on the lefthandside $b_3(k_z) = d_3(\omega)2jk$. Furthermore, the terms $b_1(k_z)$ are recognized as Fourier transforms, such that:

$$\begin{aligned} b_3(k_z) = \int e^{jk_zz_1}b_1(z_1) \quad dz_1 \int e^{-jk_zz_2}b_1(z_2) \quad dz_2 \\ \times \int e^{jk_zz_3}b_1(z_3) \quad dz_3 \end{aligned} \tag{2.43}$$

This expression can further be separated into four different contributions, regarding the relative locations between z_1, z_2, z_3. The only term that contributes to the internal multiple attenuator is the one that satisfies the LHL condition in pseudodepth (Weglein et al., 2003):

$$\begin{aligned} b_3^{IM}(k_z) = \int e^{jk_zz_1}b_1(z_1) \quad dz_1 \int_{-\infty}^{z_1-\epsilon} e^{-jk_zz_2}b_1(z_2) \quad dz_2 \\ \times \int_{z_2+\epsilon}^{\infty} e^{jk_zz_3}b_1(z_3) \quad dz_3 \end{aligned} \tag{2.44}$$

The parameter ϵ has been introduced to avoid non-linear self interaction in bandwidth-limited scenarios. In contrast to equations (2.25) or (2.36), the internal multiple attenuator given in equation (2.44) will actually yield an attenuation algorithm, due to its expression in terms of the effective data b_1. Equation (2.44) is the exact same as equation 83 in Weglein et al. (2003). Perhaps a curiosa, yet this result has been derived without the need for Cauchy principal value analysis as used in Weglein et al. (2003) and Ramírez (2007). Also note that the Green's function derivative approximation $\partial_i G n_i \approx \mp\frac{j\omega}{c}G$ is indeed exact for 1D wave propagation in a 1D medium, as can be verified by use of the analytic solution in equation (2.38).

2.8 The 2D internal multiple attenuator

The 2D generalization of equation (2.44) reads, as per Weglein et al. (2003):

$$
\begin{aligned}
b_3^{IM}(k_{x,r}, k_{x,s}, k_z) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} d\nu_1 e^{-j\nu_1(z_g - z_s)} \int_{-\infty}^{\infty} d\nu_2 e^{-j\nu_2(z_g - z_s)} \\
\int_{-\infty}^{\infty} dz_1 e^{j(\nu_r+\nu_1)z_1} b_1(k_{x,r}, k_{x,1}, z_1) \int_{-\infty}^{z_1-\epsilon} dz_2 e^{-j(\nu_1+\nu_2)z_2} b_1(k_{x,1}, k_{x,2}, z_2) \\
\times \int_{z_2+\epsilon}^{\infty} dz_3 e^{j(\nu_2+\nu_s)z_3} b_1(k_{x,2}, k_{x,s}, z_3) \quad (2.45)
\end{aligned}
$$

Where we have introduced:

- $k_{x,s}$ is the Fourier conjugate to x_s, i.e. a horizontal wavenumber.
- $k_{x,r}$ is the Fourier conjugate to x_r, i.e. a horizontal wavenumber.
- ν_s is the vertical wavenumber associated with source location:

$$\nu_s = -\operatorname{sign}(\omega)\sqrt{(\frac{\omega}{c_0})^2 - k_{x,s}^2} \quad (2.46)$$

- ν_r is the vertical wavenumber associated with receiver location:

$$\nu_r = -\operatorname{sign}(\omega)\sqrt{(\frac{\omega}{c_0})^2 - k_{x,r}^2} \quad (2.47)$$

- $k_{x,1}$ and $k_{x,2}$ are two auxilliary horizontal wavenumbers with corresponding vertical wavenumbers ν_1 and ν_2, respectively.
- k_z is defined as: $k_z = \nu_r + \nu_s$ and is the Fourier conjugate to the pseudodepth variable z.
- z represents pseudodepth.
- c_0 is the constant velocity of the reference medium, i.e. in the marine case it corresponds to the waterspeed velocity.
- b_1 is the effective data generated by a monochromatic plane-wave incident field (Weglein et al., 2003):

$$
\begin{aligned}
b_1(k_{x,r}, z_r, k_{x,s}, z_s, k_z) = \\
(-2j\nu_s)D(k_{x,r}, z_r, k_{x,s}, z_s, \omega) \quad (2.48)
\end{aligned}
$$

- ϵ is a parameter related to the wavelength of the wavelet present in the seismic data. Its purpose is to avoid self-interaction in band-width limited scenarios.

2.9 The 2D internal multiple attenuator in $\tau - p$ domain.

Equation (2.45) can be transformed to the coupled plane wave domain by using the contribution of Nita and Weglein (2009) , relating the intercept time τ to the pseudodepth z:

$$\omega\tau_i = k_z z_i \tag{2.49}$$

Furthermore, the slownesses and wavenumbers are related via:

$$k_{x,\alpha} = \omega p_\alpha \tag{2.50}$$

$$k_z = \omega q = \omega(q_s + q_r) \tag{2.51}$$

Hence, equation (2.49) can be re-written as:

$$\tau_i = q z_i \tag{2.52}$$

The contribution of Nita and Weglein (2009) made it clear that the mapping between pseudodepth and intercept-time is one-to-one, i.e.:

$$z_1 > z_2 \Leftrightarrow \tau_1 > \tau_2 \tag{2.53}$$

The coupled plane wave domain representation of (2.45) hence becomes:

$$\begin{aligned} b_3^{IM}(p_r, p_s, \omega) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} dp_1 e^{j\omega q_1(z_s - z_r)} \int_{-\infty}^{\infty} dp_2 e^{j\omega q_2(z_r - z_s)} \\ \times \int_{-\infty}^{\infty} d\tau_1 e^{j\omega\tau_1} b_1(p_r, p_1, \tau_1) \int_{-\infty}^{\tau_1 - \epsilon} d\tau_2 e^{-j\omega\tau_2} b_1(p_2, p_1, \tau_2) \\ \times \int_{\tau_2 + \epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_2, p_s, \tau_3) \end{aligned} \tag{2.54}$$

The input here is a scaled version of the input gathers after a transform to the coupled plane wave domain: $b_1(p_r, p_s, \tau_2) = -j2q_s D(p_r, p_s, \tau)$ (Sun and Innanen, 2016). It is the same type of effective data as seen in equation (2.48), yet in the coupled plane wave domain.

In order to provide an internal multiple prediction as a data term, d_3^{IM}, instead of an effective data term, b_3^{IM}, an inverse scaling is applied: $d_3^{IM}(p_r, p_s, \omega) = (-j2q_s)^{-1} b_3^{IM}(p_r, p_s, \omega)$.

Equation (2.54) is equivalent to those found in Coates and Weglein (1996) and Sun and Innanen (2016).

2.9.1 The 1.5D internal multiple attenuator in $\tau - p$ domain.

The 1.5D case describes the situation of 2D wave propagation in a laterally invariant medium. An implication of Snel's law, for media with variations only in the vertical direction, is that the horizontal slowness of any given ray is reserved. In essence, this leads to the property that for a given take-off slowness p_s, a ray propagating into the subsurface and returning to the surface, the recorded slowness component p_r is equivalent to the take-off slowness. Hence, for *all* waves scattered in a laterally invariant medium, the source and receiver slownesses are coupled one-to-one and the recorded reflection data 'lose' a degree of freedom:

$$D(p_r, p_s, \tau)\Big|_{\text{1D earth}} = D(p_r, \tau)\delta(p_r - p_s) = D(p_s, \tau)\delta(p_s - p_r) \tag{2.55}$$

By recognizing this property the internal multiple predictor in equation (2.54) can be simplified for horizontally layered media:

$$\begin{aligned} b_3^{IM}(p_r, \omega) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} d\tau_1 e^{j\omega\tau_1} b_1(p_r, \tau_1) \times \int_{-\infty}^{\tau_1 - \epsilon} d\tau_2 e^{-j\omega\tau_2} b_1(p_r, \tau_2) \\ \times \int_{\tau_2 + \epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_r, \tau_3) \end{aligned} \tag{2.56}$$

The input here is a scaled version of the reflection data transformed to the $\tau - p$ domain: $b_1(p_s, \tau) = b_1(p_r, \tau) = -j2q_r D(p_r, \tau)$.

2.9.2 The 1D internal multiple attenuator in $\tau - p$ domain.

For poststack applications, see e.g. Ramirez et al. (2017), the internal multiple predictor for one-dimensional wave propagation in a one-dimensional earth is often used. It is given by the zero-slowness component of the 1.5D predictor:

$$\begin{aligned} b_3^{IM}(\omega) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} d\tau_1 e^{j\omega\tau_1} b_1(\tau_1) \times \int_{-\infty}^{\tau_1 - \epsilon} d\tau_2 e^{-j\omega\tau_2} b_1(\tau_2) \\ \times \int_{\tau_2 + \epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(\tau_3) \end{aligned} \tag{2.57}$$

For one-dimensional wave propagation in a horizontally layered medium the vertical component of source slowness is simply $q_s = \frac{1}{c_0}$, s.t. the input becomes: $b_1(\tau) = -j2\frac{1}{c_0}D(\tau)$. Furthermore, in this situation the vertical intercept time, τ, is equal to the two-way traveltime, t.

2.10 Psuedodepth-wavenumber and coupled slowness domains for internal multiple prediction

The internal multiple prediction in psuedo-depth wavenumber given in equation (2.45), requires a mapping of seismic gathers from time to pseudodepth. This requires one to perform a constant-velocity Stolt migration of the data in the frequency domain. The velocity used for the marine case is the waterspeed velocity. Stolt migration involves a simple phaseshift mapping of angular frequency, ω, to vertical wavenumber, k_z. In spatial dimensions beyond one, the dispersion relation used for this mapping is non-linear in the relation $\omega \leftrightarrow k_z$ and vice versa. Creating the pseudodepth axis therefore requires extensive interpolation to create a regular grid in k_z, given a regular grid in ω. The interpolation operations themselves are not trivial to implement correctly. As shown by Harlan (1982), treating the interpolation operators incorrectly can cause incorrectly migrated events to entirely replace correctly migrated events.

Furthermore, the input to the psuedodepth-wavenumber prediction is not easily interpretable nor intuitive to display. Sun and Innanen (2015) have indicated that the parameter ϵ is not stationary in this domain and that this can lead to artifacts in the internal multiple prediction.

In contrast, Sun and Innanen (2015) demonstrated, for the 1.5D case, that the plane wave domain provides an environment where ϵ is quite stationary. Internal multiple predictions with relatively low levels of artifacts are achieved in this domain. Sun and Innanen (2016) showed that the input in the coupled plane wave domain is somewhat sparse with respect to (p_r, p_s), where the maximum contributing bandwidth for a constant $p_{r,s}$ section in terms of $p_{s,r}$ is related to the maximum dip angle of reflectors. Indeed, as the maximum dip of the medium approaches $0°$, the 2D predictor gradually deteriorates to the 1.5D predictor, except that the calculation is still performed in the coupled plane wave domain. For media with limited reflector dips, the sparseness of the input data in the coupled plane wave domain can potentially be used to minimimize the operations count required to calculate internal multiple predictions. For these reasons, the coupled plane wave domain is our preferred domain for internal multiple prediction. Its only downside is that the linear Radon transform(s) required to create the input data to this domain require some work in order to perform at the required fidelity. The properties of the linear Radon transform and a proposed algorithm to create the input in the coupled plane wave transform is discussed in chapter 3.

2.11 Internal multiple prediction using migrated data

The original input to the internal multiple prediction in equation (2.54), before coupled plane wave transforms, needs to be unmigrated. In exploration seismology, many seismic datasets are only available post migration, either collapsed or uncollapsed. In order to accomodate internal multiple prediction for uncollapsed migrated datasets, the author identified two options. Either, the internal multiple predictor in e.g. equation (2.54) must be modified to allow for migrated datasets. The other alternative is to construct the input data needed by undoing the migration performed. The process that accomplishes this is termed demigration.

The work performed in Aaker (2017) discussed these two alternatives. A key conclusion reached was that it would be computationally optimal to perform demigration prior to internal multiple prediction, rather than modifying the internal multiple prediction integrals. Demigration furthermore enables data reconstruction on survey geometries different from original, physical surveys. Hence, it has the possibility to provide data that satisfy the stringent and special sampling requirements of the multidimensional Inverse Scattering Series internal multiple predictors.

Chapter 3

The Linear Radon Transform and transformation to the Coupled Plane Wave domain

Disclaimer: The work presented in this chapter is partially derived from the work in Aaker (2017) performed in the course *TPG4570 - Petroleum Geosciences, Specialization Project*.

This chapter starts with the the definition of the linear Radon transform, and in particular provides the link to perform the transformation via frequency-domain operators. The formulation of the forward transform as an inverse problem is discussed in section 3.2. The underlying rationale of this formulation is also introduced. Based upon the work of Turner (1990), the sampling requirements of the linear RT are briefly reviewed. In section 3.4 transformations to the coupled plane wave domain are explicitly discussed. A proposed algorithm for performing the transformation is presented. The final section discusses some of the well-known artifacts that may arise due to application of (standard) linear Radon transforms, and attempts to point at which problems this might give for internal multiple prediction. In doing so, it also points to a source of improvement, so-called high-resolution Radon transforms-discussed thoroughly in chapter 4.

3.1 Definition of the Linear Radon Transform

For the first subsections the transform considered is not a coupled transform, as this is elaborated on in section 3.4. For now, consider the data $f(x,t)$ as a constant section of the same function with (possibly) more degrees of freedom. Id est:

$$f(x,t) := f(x', x'', \ldots, = \textbf{const}, x, t) \quad , \forall (x,t)$$

.

The general Radon transform with kernel ϕ is hereby written as:

$$\hat{f}(p,\tau) = \int_{\mathbb{R}} dx \int_{\mathbb{R}} dt \quad \phi(x,t,p,\tau) f(x,t) \tag{3.1}$$

For the linear Radon transform the kernel is defined to be: $\phi = \delta(t - px - \tau)$. As such, the transform reads, by the sifting properties of the delta function:

$$\hat{f}(p,\tau) = \int_{\mathbb{R}} dx f(x, t = \tau + px) \tag{3.2}$$

The corresponding inverse transform has the formal expression:

$$f(x,t) = \int_{\mathbb{R}} dp \hat{f}(p, \tau = t - px) \tag{3.3}$$

The linear Radon transform, in similarity to the parabolic RT, utilizes a time-invariant operator. For this very reason it is equivalently possible to calculate the transformations using frequency domain operators. In the latter case, the amount of required computations are bounded by $\mathcal{O}(N_\omega \times N_p \times N_x)$, whereas an implementation using time domain operators have an asymptotic bound of $\mathcal{O}(N_t \times N_\tau \times N_p \times N_x)$. For this very reason, the forward and inverse transforms are commonly implemented in frequency rather than in time domain. In this thesis we will only consider calculating the linear Radon transform using frequency domain operators.

Performing the Fourier transform of the forward transform (3.2) with respect to τ yields:

$$\hat{f}(p,\omega) = \int_{\mathbb{R}} d\tau \int_{\mathbb{R}} dx f(x, \tau + px) e^{-j\omega\tau} \tag{3.4}$$

A change of integration variables is introduced: $u = \tau + px$. This leads to the following frequency-domain forward transform.

$$\hat{f}(p,\omega) = \int_{\mathbb{R}} du \int_{\mathbb{R}} dx f(x,u) e^{-j\omega(u-px)} \tag{3.5}$$

$$\hat{f}(p,\omega) = \int_{\mathbb{R}} dx f(x,\omega) e^{j\omega px} \tag{3.6}$$

Idem, a frequency-domain inverse transformation exists, namely:

$$f(x,\omega) = \int_{\mathbb{R}} dp \hat{f}(p,\omega) e^{-j\omega px} \tag{3.7}$$

The discretized versions of equations (3.6) and (3.7) read:

$$\hat{f}(p_k,\omega) = \sum_{j=0}^{N-1} f(x_j,\omega) e^{j\omega p_k x_j},\ k = 0, 1, \ldots, M-1 \tag{3.8}$$

$$f(x_j,\omega) = \sum_{k=0}^{M-1} \hat{f}(p_k,\omega) e^{-j\omega p_k x_j},\ j = 0, 1, \ldots, N-1 \tag{3.9}$$

The summations are recognized as Matrix-Vector multiplications. The corresponding linear algebraic notation is chosen as follows:

$$\hat{\mathbf{f}}(\omega) = \mathbf{L}^{\dagger}(\omega)\mathbf{f}(\omega) \tag{3.10}$$

$$\mathbf{f}(\omega) = \mathbf{L}(\omega)\hat{\mathbf{f}}(\omega) \tag{3.11}$$

Using the linear operator and its adjoint constitute a way to perform the inverse and forward linear Radon transforms in the frequency domain. In practice however, such implementations yield smearing and poor resolution by the construction to the $\tau - p$ domain. The resulting reconstruction, via the inverse transform, is therefore poor (Sacchi and Ulrych, 1995). The forward transform is therefore cast as an inverse problem. This is the next point of study.

3.2 Forward Radon Transform as an inverse problem

In order to ease the notational treatment of the forward transform as an inverse problem, we will introduce the following notation:

$$\hat{\mathbf{f}}(\omega) = \mathbf{m}(\omega) \tag{3.12}$$

$$\mathbf{f}(\omega) = \mathbf{d}(\omega) \tag{3.13}$$

From hereon, the notation of frequency dependence is for the sake of convenience dropped.

We define the inverse transform to be the forward problem of study:

$$\mathbf{d} = \mathbf{L}\mathbf{m} \tag{3.14}$$

We introduce the (zeroth order) Tikhonov regularized objective function φ, the combination of two complex ℓ_2 norms:

$$\varphi(\mathbf{m}|\mathbf{d}, \lambda) = ||(\mathbf{d} - \mathbf{L}\mathbf{m})||_2^2 + \lambda||\mathbf{m}||_2^2 = (\mathbf{d} - \mathbf{L}\mathbf{m})^{\dagger}(\mathbf{d} - \mathbf{L}\mathbf{m}) + \lambda\mathbf{m}^{\dagger}\mathbf{m} \tag{3.15}$$

λ is a regularization parameter, controlling the relationship between least-squares fitting and regularization. The regularization imposed here is a damping towards a zero-valued prior model. It is often used to stabilize and filter the SVD components of the solution (Hansen, 2005). The Radon model optimal in the regularized least squares sense is estimated by minimizing the objective function with respect to $\mathbf{m}$.

$$\mathbf{m} = \arg\min_{\mathbf{m}} \Big\{\varphi(\mathbf{m}|\mathbf{d}, \lambda)\Big\} \tag{3.16}$$

By expanding the complex ℓ_2 norms, the objective function can be re-written as:

$$\varphi = \mathbf{m}^{\dagger}\mathbf{L}^{\dagger}\mathbf{L}\mathbf{m} + \mathbf{d}^{\dagger}\mathbf{d} - \mathbf{d}^{\dagger}\mathbf{L}\mathbf{m} - \mathbf{m}^{\dagger}\mathbf{L}^{\dagger}\mathbf{d} + \lambda\mathbf{m}^{\dagger}\mathbf{m} \tag{3.17}$$

The objective function is a scalar real-valued function consisting of complex-valued arguments: $\varphi : \mathbb{C} \rightarrow \mathbb{R}$. There are two equivalent conditions that describe the stationary points of such a function (Kreutz-Delgado, 2009):

$$\frac{\partial\varphi}{\partial\mathbf{m}}\Big|_{\mathbf{m}^*=const} = 0 \tag{3.18}$$

equivalently,

$$\frac{\partial\varphi}{\partial\mathbf{m}^*}\Big|_{\mathbf{m}=const} = 0 \tag{3.19}$$

We first consider the former condition, by applying the cogradient operator:

$$\frac{\partial\varphi}{\partial\mathbf{m}}\Big|_{\mathbf{m}^*=const} = \left(\mathbf{L}^\dagger\mathbf{L}\right)^T\mathbf{m}^* - \mathbf{L}^T\mathbf{d}^* + \lambda\mathbf{m}^* = \mathbf{L}^\dagger\mathbf{L}\mathbf{m}^* - \left(\mathbf{L}^\dagger\mathbf{d}\right)^* + \lambda\mathbf{m}^* = 0 \tag{3.20}$$

By applying the conjugate cogradient operator, the latter condition gives:

$$\frac{\partial\varphi}{\partial\mathbf{m}^*}\Big|_{\mathbf{m}=const} = \mathbf{L}^\dagger\mathbf{L}\mathbf{m} - \mathbf{L}^\dagger\mathbf{d} + \lambda\mathbf{m} = 0 \tag{3.21}$$

Evidently we have the symmetry that:

$$\frac{\partial\varphi}{\partial\mathbf{m}}\Big|_{\mathbf{m}^*=const} = \left\{\frac{\partial\varphi}{\partial\mathbf{m}^*}\Big|_{\mathbf{m}=const}\right\}^* \tag{3.22}$$

A closed form solution is reached. The regularized least squares optimal Radon model therefore reads:

$$\mathbf{m} = (\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})^{-1}\mathbf{L}^\dagger\mathbf{d} = (\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})^{-1}\mathbf{m}_{adj} \tag{3.23}$$

$\mathbf{m}_{adj}$ is defined to be the low resolution transform using the adjoint operator $\mathbf{L}^\dagger$.

On the structure of $(\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})$

Due to its computational implications, it will be demonstrated that the matrix of the normal equations, $(\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})$, possesses an Hermitian Toeplitz structure.

Consider the definition of the Hermitian transpose of a matrix $\mathbf{A}$:

$$(\mathbf{A}_{i,j})^\dagger = (\mathbf{A}_{j,i})^* \tag{3.24}$$

The definition of matrix-matrix multiplication is:

$$(\mathbf{C})_{i,j} = \sum_k \mathbf{A}_{i,k}\mathbf{B}_{k,j} \tag{3.25}$$

Thereby, we express the matrix-matrix product $\mathbf{L}^\dagger\mathbf{L}$ as:

$$[\mathbf{L}^\dagger\mathbf{L}]_{l,m} = \sum_{k=0}^{M-1}\mathbf{L}^*_{k,l}\mathbf{L}_{k,m} \qquad , \ \mathbf{L}_{k,m} = e^{-j2\pi f p_m x_k}$$

$$[\mathbf{L}^\dagger\mathbf{L}]_{l,m} = \sum_{k=0}^{M-1} e^{j2\pi f p_l x_k} e^{-j2\pi f p_m x_k} = \sum_{k=0}^{M-1} e^{j2\pi f(l-m)\Delta p x_k} \tag{3.26}$$

The uniform samplying property of horizontal slowness has been defined and used:

$$p_i := p_0 + i\Delta p,\ i = 0, 1, \ldots, M-1 \tag{3.27}$$

From equation (3.26) one can identify that the matrix $\mathbf{L}^\dagger\mathbf{L}$ has an Hermitian Toeplitz structure. Idem, the Tikhonov regularized variant, $(\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})$, is Hermitian Toeplitz with entries:

$$[(\mathbf{L}^\dagger\mathbf{L} + \lambda\mathbf{I})]_{l,m} = \sum_{k=0}^{M-1} e^{-j2\pi f(l-m)\Delta p x_k} + \lambda\delta_{l,m} \tag{3.28}$$

Inverting a Toeplitz matrix is highly computationally efficient by the Levinsion recursion scheme and is on the order of $\mathcal{O}(M^2)$ operations (Golub and Van Loan, 2012). Schemes valid for general, full-rank matrices, e.g. LU factorization or Gaussian Elimination are of the order $\mathcal{O}(M^3)$ operations. Inversion by the Levinson recursion schemes is therefore computationally very attractive. As an example, the program *sfradon* in Madagascar (Madagascar Development Team, 2012) utilizes this Toeplitz structure in order to perform the forward Radon transform.

However, as shown by e.g. Sacchi and Ulrych (1995), this particular form of regularized least-squares inversion does not retrieve a high-resolution Radon model. The zero-th order Tikhonov regularization, equivalent to assuming a Gaussian prior in model space, constitues a major source of amplitude smearing in the Radon domain. Improved schemes with less smearing and better reconstruction are available and are discussed in chapter 4.

3.3 Linear Radon Sampling Requirements

Turner (1990) derived sampling criterions for the plane wave transform by a geometrical argument of constructive summing along slants. These are briefly reviewed here.

- **Sampling in τ:**
  This is equivalent to the Shannon-Nyquist sampling criterion:

$$\Delta\tau \leq \frac{1}{2f_{\text{max}}} \tag{3.29}$$

- **Sampling in p:**

$$\Delta p \leq \frac{1}{x_{\text{range}} f_{\text{max}}} \tag{3.30}$$

$$x_{\text{range}} := x_{\text{max}} - x_{\text{min}}$$

- **Sampling in x:**

$$\Delta x \leq \frac{1}{p_{\text{range}} f_{\text{max}}} \tag{3.31}$$

$$p_{\text{range}} := p_{\text{max}} - p_{\text{min}}$$

The most critical sampling requirement for the input data to the linear Radon transform is that given by (3.31) as it is a data domain requirement. For seismic surveys, it is virtually always far more restrictive than that of (3.29). The sampling requirement in (3.30) is a model domain requirement and thus is selected by the end user of linear Radon transforms.

3.4 Radon Transform to the coupled plane wave domain

So far, the discussion on the linear Radon transform has been in a univariate setting with respect to spatial coordinates. Constructing the data in a coupled plane-wave domain can be done by a frequency domain implementation akin to (3.6) (Stoffa et al., 2006):

$$\hat{f}(\mathbf{p}_r, \mathbf{p}_s, \omega) = \int_{\mathbb{R}} d\mathbf{x}_s d\mathbf{x}_r f(\mathbf{x}_r, \mathbf{x}_s, \omega) e^{j\omega(\mathbf{p}_r\cdot\mathbf{x}_r + \mathbf{p}_s\cdot\mathbf{x}_s)} \tag{3.32}$$

Casting the transform in (3.32) as a least squares inverse problem with regularization is straightforward in the sense previously discussed. The most prominent downside to such an approach is that code from existing linear Radon transforms is not easily (re-)useable. This drawback can be avoided by performing the transform in two steps:

Algorithm 1 Perform coupled plane wave transform through univariate plane wave transform

- **I)** A linear Radon transform is performed over $\mathbf{x}_r$. The output data are in the domain $(\mathbf{p}_r, \mathbf{x}_s, \tau)$.
- **II)** Resort the data into constant $\mathbf{p}_r$ sections. These sections appear similar to common receiver sections, except that arrival times are replaced by vertical delay times and that the data for each source position have a common angle of incidence at the surface (instead of a common receiver).
- **III)** Perform linear Radon transform over $\mathbf{x}_s$. The output data are in the domain $(\mathbf{p}_r, \mathbf{p}_s, \tau)$, but sorted in common receiver slowness sections.
- **IV)** Resort data to regular sorting w.r.t $(\mathbf{p}_r, \mathbf{p}_s, \tau)$.

Hence, code for a normal common-source or common-offset section plane wave transform can easily be used for performing the forward transform to the coupled plane wave domain. Otherwise, the existing code can be modified in order to allow strided memory access across common source sections. Such an approach would avoid the two sorting operations in algorithm 1.

A particularity of transforming to the coupled plane wave domain via algorithm 1 Experimentally, I note that in order for Snel's law to read: $D(p_r, \tau) = D(p_r, p_s, \tau)\delta(p_r - p_s)$, one modification must be made. In step **III**, one needs to reverse either the p_s *or* the

x_s axis. If this is not done, the data represented in the coupled plane wave domain read $D(p_r, p_s, \tau)\delta(p_r + p_s)$.

Coupled plane wave domain sampling requirements
The only way to completely sample all of the source and receiver slowness contributions of the medium is to have an acquisition geometry where, within a given aperture, sources and receivers occupy all discrete positions along the acquisition surface. This translates to fixing the horizontal position of the receivers and allowing the sources to fire along the same positions. Real world examples of acquisition geometries that satisfy this requirement include marine OBC geometries and certain land geometries, e.g. the roll-along geometry. For marine streamer data this acquisition geometry cannot be met in the original survey. Data reconstruction techniques such as e.g. modelling by demigration are therefore needed in order to provide sufficient data in the presence of marine streamer surveys.

3.5 Truncation artifacts

The explicit linear Radon transform is only exact for infinite aperture data available in a continuum. Suffice to say, such data are never available (or even possible to work with). The absence of available information beyond the survey aperture causes an amplitude diminution and some signal distortion (Wang and Houseman, 1997). By applying a truncated summation to approximate the infinite integral of the linear Radon transform, certain linear dipping events are introduced in the transformed domain. They do not correspond to any physical events as e.g. mappings of traveltime curves, and are therefore considered unwanted.

Amelioration: Spatial tapering
A much used strategy to suppress linear truncation artifacts is to perform spatial tapering of the data near the ends of the survey apterture. This is similar to the effect of tapering a time series reduces spectral leakage in the Fourier domain. Smoothly varying tapering functions such as sine or cosine functions are often used (Wang and Houseman, 1997). E.g. for tapering a common shot gather along the ends of the receiver axis, the tapered data can be expressed as:

$$d^{\,\text{taper}}(x_r, x_s, t) = \left\{ \begin{array}{ll} d(x_r, x_s, t)\cos\left[\frac{\pi}{2}\left(\frac{x_r - x_{ref}}{x_{max} - x_{ref}}\right)\right], & \text{for } x_{ref} \leq x_r \leq x_{max} \\ d(x_r, x_s, t), & \text{for } x < x_{ref} \end{array} \right\} \tag{3.33}$$

x_{ref} is the reference position within the aperture beyond which spatial tapering should take place. The formulation ensures that at $x_r = x_{ref}$ the amplitude is preserved, while at $x_r = x_{max}$ it is 0.

Strategies such as tapering naturally suffer from the fact that they are not proper solutions to the non-orthogonality of the linear Radon transform. Their effectiveness is therefore only limited. Furthermore, any reconstruction of tapered parts of a given input dataset is highly limited after application of the inverse transform. For the reasons considered above, more sophisticated solutions to the aperture related artifacts are sought.

Amelioration: High-resolution linear Radon transforms
Not only are high-resolution Radon transforms efficient in reducing amplitude smearing, they are also able to minimize aperture effects. This is done by building a sparser Radon model than what is given by the adjoint or Tikhonov regularized least-squares optimal model. High-resolution Radon transforms are discussed in chapter 4.

Implications for Inverse Scattering Series based internal multiple prediction
1.5D: For internal multiple prediction using the 1.5D algorithm, equation (2.56), typical artifacts in the input are the lines shown on the right-hand panel of figure 3.1. When the linear artifacts coincide with the slowness range used to calculate the internal multiple prediction, they may combine with the data to create other linear artifacts, a type of 'pseudo-event'. This is naturally unwanted. However, as these pseudo-events are lines in the Radon space, they will map back to points in the physical domain via the inverse transforms. Single points per definition have sparse support. The effect of the pseudo-events should not be too large after inverse transforming. *In the 1.5D case the stationarity of the search-limiting parameter ϵ does not seem too threatened by the possible presence of a few linear artifacts in the input data.*

2D: When performing prediction using the 2D predictor in the coupled plane wave domain, equation (2.54), the situation is very different from the 1.5D case. Here, the input data have a representation which is a lot sparser than what they would have in the physical domain or after a univariate Radon transforms. Certain events originating from an approximately horizontal part of the subsurface, e.g. a waterbottom reflection, should map approximately to a point with respect to the support in (p_r, p_s), in the coupled plane wave domain. Only a linear Radon transform with a high resolving power will be able to do what theory describes is optimal and/or correct. Figure 3.2 shows typical artifacts in the coupled plane wave domain generated using a standard least squares transform. The input data were modelled using a 1.5D earth model. Knowing the particular properties of the model, every subevent in figure 3.2 should either collapse to a single point if the slowness component $(p_r,\ p_s)$ was recorded, or conversely completely vanish if it was not. The left-hand panel shows a common $p_s = 0\ s/km$ slice of the volume. Significant butterfly effects are evident. The right-hand panel demonstrates that at higher dips, $p_s = 0.2\ s/km$, certain events that should vanish (slowness component not recorded) are smeared out on lines.

The 2D internal multiple predictor searches across combinations of $p_r,\ p_s$ and auxiliary source and receiver slownesses to construct the internal multiple prediction. The physical subevents in the plane wave domain may be allowed to combine nonlinearly with transform artifacts. The result of an insuffcent Radon transform to the coupled plane wave domain will be that the search-limiting parameter ϵ, which attempts to prevent non-linear combinations, is challenged or even violated. *An implication of transform artifacts in the coupled plane wave domain is the presence of predicted 'pseudo-events' in the internal multiple prediction.* Compared to the 1.5D case, these artifacts may span a larger space both in the coupled plane wave domain and in the physical domain.

Figure 3.1 A shot gather (left) transformed to the $\tau - p_o$ domain via a standard Least Squares transform. Red arrows point at linear transform artifacts.

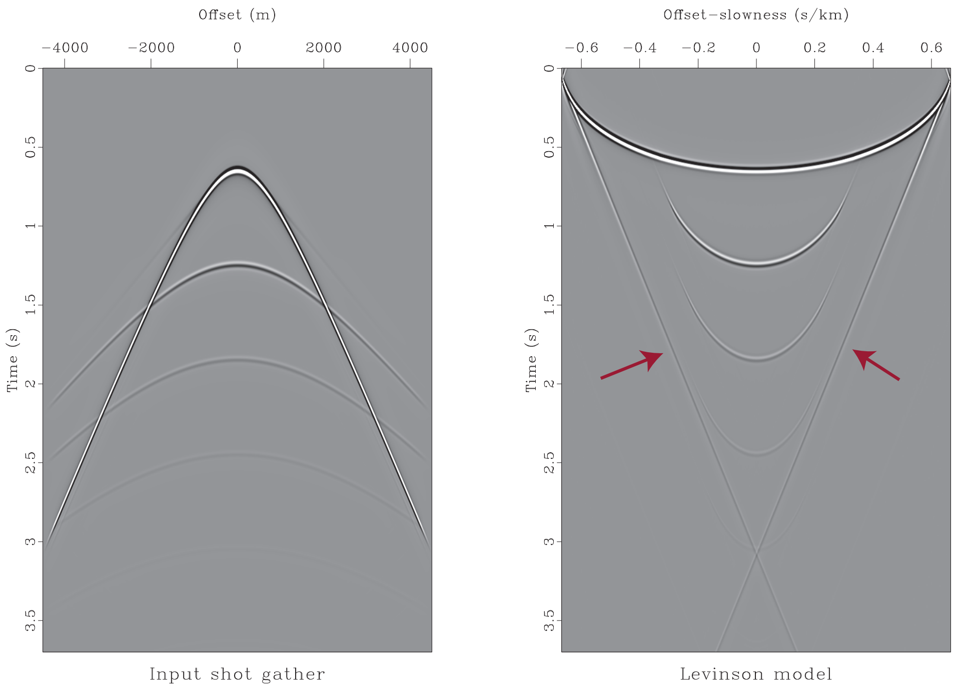
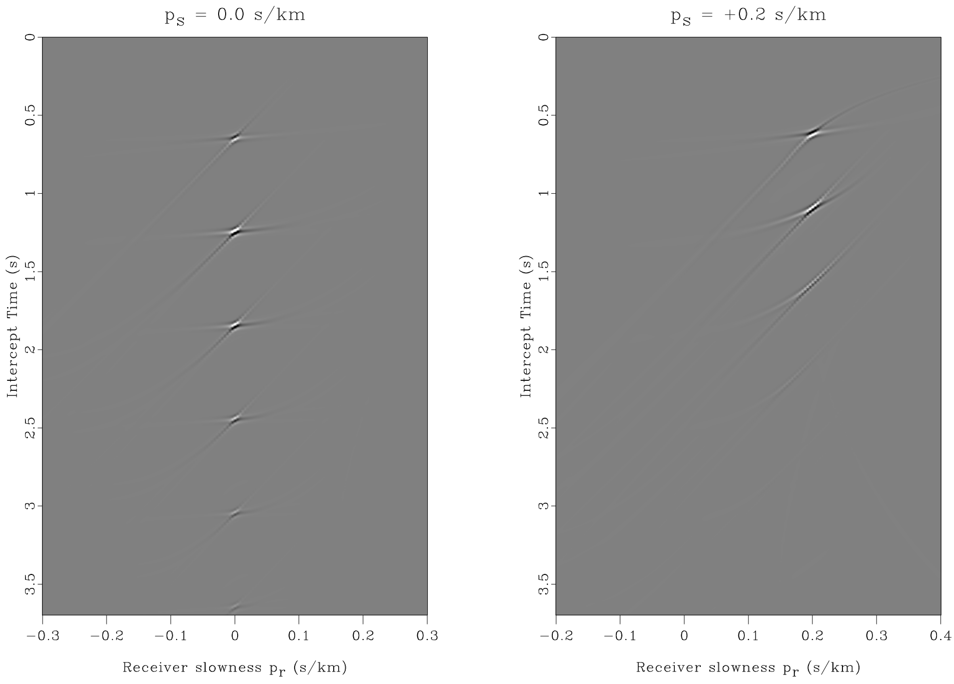


Figure 3.2 Typical 'butterfly' and linear artifacts in the coupled plane wave domain produced by using a standard, least squares linear RT.



Chapter 4

High resolution Linear Radon transforms

This chapter is motivated by the analysis in chapter 3 on some of the deficiences of the linear Radon transform, either as a direct implementation or in a least-squares, Tikhonov regularized formulation. Linear Radon transforms often suffer from truncation artifacts, as the explicit transformation is only exact for infinite aperture data. It is the conjecture from section 3.5 that truncation artifacts will negatively affect Inverse Scattering Series internal multiple prediction. This would especially be true for performing prediction in two or higher dimensions.

In order to provide optimal input for the internal multiple prediction we consider the use of high resolution, also termed sparse, linear Radon transforms. This chapter attempts to give a thorough analysis of sparse time-invariant RTs. Implementational results are demonstrated at the very end. Sacchi and Ulrych (1995) provided the first proper analysis on the underlying reasons why standard Radon transforms suffer from artifacts and low resolving power. They also provided the first analysis and derivations of high resolution time-invariant Radon transforms. This was done in a setting of probabilistic inverse problems. For historical reasons we will therefore start the analysis in a Bayesian setting in sections 4.1-4.4. By recognizing a special case of symmetry between Bayesian and deterministic inverse theory, we will from thereon move to the latter setting. From section 4.5 and onwards some modern advancements on sparse inversion will also be considered, e.g. results from compressive sensing.

4.1 Bayesian inversion theory

Consider the forward modelling theory given by the linear operator $\mathbf{L}$ acting on the model $\mathbf{m}$ yielding the data $\mathbf{d}$. Considering the effect of noise, due to either incomplete modelling theory and/or data with noise, this yields the relation:

$$\mathbf{Lm} + \mathbf{n} = \mathbf{d} \tag{4.1}$$

We will solve the linear inversion problem for $\mathbf{m}$ via a probabilistic formalism. For this, we introduce and consider Bayes' formula

$$p(\mathbf{m}|\mathbf{d}) = \frac{p(\mathbf{d}|\mathbf{m})p(\mathbf{m})}{\int_{\mathcal{M}} p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m}} \tag{4.2}$$

Bayes' formula gives the posterior probability density function (abbreviated pdf) of the model parameters $\mathbf{m}$ given the data $\mathbf{d}$, in terms of

- The prior in model space $p(\mathbf{m})$
- The data *likelihood function* $p(\mathbf{d}|\mathbf{m})$.
- The normalisation factor $\int_{\mathcal{M}} p(\mathbf{d}|\mathbf{m})p(\mathbf{m})d\mathbf{m} = p(\mathbf{d})$, commonly termed the *evidence*.

The model space posterior, $p(\mathbf{m}|\mathbf{d})$, contains all information obtainable by combining the prior knowledge of the forward modelling theory, the data and the parameters of the model. Prior knowledge of the forward modelling theory and the data are jointly incorporated into the likelihood function, $p(\mathbf{d}|\mathbf{m})$, whereas prior knowledge of the model parameters is represented by the prior in model space, $p(\mathbf{m})$. The posterior probability density function is *the* solution of the Bayesian inverse problem. In the probabilistic framework there are no constraints on the properties of the forward problem, e.g. linearity and differentiability[1]. The denominator, the evidence, in Bayes' formula is often ignored. As a normalisation factor it does not modify the relative probabilities in the model space. Furthermore, it requires an $\mathcal{M}$ dimensional integration routine, which can become prohibitively expensive for geophysical inverse problems. The unnormalised expression for the model space posterior therefore reads:

$$p(\mathbf{m}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{m})p(\mathbf{m}) \tag{4.3}$$

In order to derive a model from Bayesian inversion by equation (4.3) one must consider:

- I) How to determine *the* model $\mathbf{m}$ given $p(\mathbf{m}|\mathbf{d})$.
- II) How to construct the prior in model space $p(\mathbf{m})$

For the first problem one can determine a (suitable) model by applying a decision rule onto $p(\mathbf{m}|\mathbf{d})$. A frequently used rule is to consider the maximum a posteriori (MAP) model, $\mathbf{m}_{MAP}$, which is defined as the model that maximizes $p(\mathbf{m}|\mathbf{d})$:

$$\mathbf{m}_{MAP} := \arg\max_{\mathbf{m}}\{p(\mathbf{m}|\mathbf{d})\} \tag{4.4}$$

[1] Only in the special case of Radon transforms will we consider a linear forward problem.

The second problem concerns how to translate prior knowledge and preference of model space characteristics into a probability density function. In the context of linear Radon transforms we seek a solution that has sparse support, in order to minimize artifacts.

4.2 Constructing the prior from the principle of maximum entropy

For a continous variable $\mathbf{m}$ with probability density function $p(\mathbf{m})$ the entropy $\mathcal{H}$ is defined as:

$$\mathcal{H} := -\int_{\mathcal{M}} p(\mathbf{m}) \log_a[p(\mathbf{m})] d\mathbf{m} \tag{4.5}$$

By relation to Shannon's measure of information content:

$$\mathcal{H} = E\{-\log_a[p(\mathbf{m})]\} = E\{I(\mathbf{m})\} \tag{4.6}$$

The entropy is equivalent to the expected value of the information content. The base, a, of the logarithm determines the unit of the entropy.

The distribution that most honestly describes the model, given *only* what is known, is the maximum entropy distribution (Jaynes, 1968). When making inferences based upon incomplete information it is therefore sound to draw them from the pdf that has the maximum entropy allowed by the available information. We would also like to constrain the model towards our preferred characteristics.

Hence, we wish to describe the prior by maximizing:

$$\mathcal{H} = -\int_{\mathcal{M}} p(\mathbf{m}) \log_a[p(\mathbf{m})] d\mathbf{m} \tag{4.7}$$

Subject to the constraints:

$$\int_{\mathcal{M}} p(\mathbf{m}) d\mathbf{m} = 1 \tag{4.8}$$

$$\int_{\mathcal{M}} p(\mathbf{m}) f(\mathbf{m}) d\mathbf{m} = F := E\{f(\mathbf{m})\} \tag{4.9}$$

The constraint (4.8) ensures the second axiom of probability. In equation (4.9) we have introduced the constraint $f(\mathbf{m})$, acting on the prior $p(\mathbf{m})$, and its associated expected value F. In the continous case one can impose an arbitrary number of constraints on $p(\mathbf{m})$. In this consideration one constraint suffices, namely one that promotes sparseness.

For solving the maximization problem we consider the functional $J(p)$ constructed via the method of Lagrange multipliers:

$$J(p) := \int_{\mathcal{M}} p(\mathbf{m}) \log_a[p(\mathbf{m})] d\mathbf{m} \tag{4.10}$$

$$-\lambda_0 \Big[\int_{\mathcal{M}} p(\mathbf{m}) d\mathbf{m} - 1 \Big] - \lambda_1 \Big[\int_{\mathcal{M}} p(\mathbf{m}) f(\mathbf{m}) d\mathbf{m} - F \Big] \tag{4.11}$$

λ_0, λ_1 are the Lagrange multipliers associated with each of the constraints, respectively. In geophysical inverse problems we are not concerned with finding the Lagrange multipliers, they are selected by the end user, unless some automatic parameter selection procedure is considered. If the functional $J(p)$ has a local maxima at $p = h$, the test function $\eta(\mathbf{m})$ is an arbitrary, smooth function with compact support in $\mathcal{M}$ and for any number ϵ close to 0, the following inequality holds:

$$J(h) \geq J(h + \eta\epsilon) \tag{4.12}$$

$J(h+\eta\epsilon)$ is termed the first variation of the functional J. Given that $J(p)$ has a maximum at $p = h$ it follows that its first variation has a maximum around $\epsilon = 0$.

$$\frac{\partial J(h+\eta\epsilon)}{\partial\epsilon} = \int_{\mathcal{M}} \eta \log_a[h+\eta\epsilon] d\mathbf{m} \tag{4.13}$$

$$+ \int_{\mathcal{M}} h \frac{1}{h+\eta\epsilon} \eta d\mathbf{m} + \int_{\mathcal{M}} \eta\epsilon \frac{1}{h+\eta\epsilon} \eta d\mathbf{m} - \lambda_0 \int_{\mathcal{M}} \eta d\mathbf{m} - \lambda_1 \int_{\mathcal{M}} \eta f d\mathbf{m} \tag{4.14}$$

$$\frac{\partial J(h+\eta\epsilon)}{\partial\epsilon}\bigg|_{\epsilon=0} = \int_{\mathcal{M}} \eta(\mathbf{m}) \Big\{ \log_a[h(\mathbf{m})] + 1 - \lambda_0 - \lambda_1 f(\mathbf{m}) \Big\} d\mathbf{m} = 0 \tag{4.15}$$

Given the properties of $\eta(\mathbf{m})$ one can apply the fundamental lemma of calculus of variations and therefore retrieve:

$$\log_a[h(\mathbf{m})] + 1 - \lambda_0 - \lambda_1 f(\mathbf{m}) = 0 \tag{4.16}$$

By reversing the renaming previously implied, $h \rightarrow p$, and by re-writing:

$$p(\mathbf{m}) = e^{1-\lambda_0-\lambda_1 f(\mathbf{m})} \propto e^{\lambda_0-\lambda_1 f(\mathbf{m})} \tag{4.17}$$

The prior model $p(\mathbf{m})$ in (4.17) is the prior model that maximizes the entropy, (4.7) given the constraints in (4.8) and (4.9). In the last step we have for simplicity used logarithm base $a = e$, without any significant loss of generality.

4.3 Sparseness-promoting constraints

There still remains the task of determining a suitable constraint $f(\mathbf{m})$ given our preference for sparse models. Sacchi and Ulrych (1995) state that a constraint of the form

$$f(\mathbf{m}) = \sum_i \ln[m_i^* m_i + b^2] \tag{4.18}$$

can be used to quanitfy the amount of sparseness of a vector. It is a stabilized version of Burg's measure of entropy (Burg, 1975), where the parameter b^2 acts as a stabilizer. Alternatively, one can assume a Cauchy pdf for the model space prior, which will yield the exact same system of normal equations. The normal equations themselves are to be derived and will follow. In the case of a Cauchy prior however, the hyperparameter b^2

does not (explicitly) represent a background power or stabilization factor. Rather, it represents a probabilistic hyperparameter, namely the model space variance σ_m^2. Due to ease of reference we may refer to the minimization of Burg's measure of entropy constraint as assuming a Cauchy prior in the following sections.

When considering the maximum entropy prior, equation (4.17), the resulting prior pdf is given by:

$$p(\mathbf{m}) = \frac{e^{\lambda_0}}{\prod\limits_i [m_i^* m_i + b^2]^{\lambda_1}} \tag{4.19}$$

Ignoring the factor constant for all parts of model space, we get the per-element 1D prior:

$$p(m_i) \propto \frac{1}{[m_i^* m_i + b^2]^{\lambda_1}} \tag{4.20}$$

For large values of λ_1, the resulting prior distribution is sharp. Conversely, for very small values of λ_1 the distribution $p(m_i)$ approaches uniformity. A uniform prior is naturally uninformative. In order to have a preference for sparse models λ_1 can therefore not be set too small.

In the case that the stabilization factor b^2 is large compared to $m_i^* m_i$, one can approximate $\ln[m_i^* m_i + b^2] \approx \frac{m_i^* m_i}{b^2} + \ln(b^2)$ (Sacchi and Ulrych, 1995). Hence, for relatively large b^2 equation (4.20) can be replaced by:

$$p(m_i) \propto \frac{e^{-\lambda_1 \frac{m_i^* m_i}{b^2}}}{b^{2\lambda_1}} \tag{4.21}$$

This represents a Gaussian distribution, which in the deterministic setting is equivalent to using zero-th order Tikhonov regularization. Gaussian distributions promote smoothness and do not allow a sparse representation. With this in mind, it is very important not to select b^2 too large, otherwise no sparseness can be obtained.

4.4 The maximum a posteriori model

Under the assumption that the noise in equation (4.1) follows a Gaussian distribution, the calculated likelihood function is given by:

$$p(\mathbf{d}|\mathbf{m}) = e^{-\frac{1}{2}\{\mathbf{Lm}-\mathbf{d}\}^\dagger \mathbf{C}_n^{-1}\{\mathbf{Lm}-\mathbf{d}\}} \tag{4.22}$$

The unnormalized posterior distribution follows from Bayes' rule

$$p(\mathbf{m}|\mathbf{d}) \propto e^{-\lambda_1 f(\mathbf{m})} \cdot e^{-\frac{1}{2}\{\mathbf{Lm}-\mathbf{d}\}^\dagger \mathbf{C}_n^{-1}\{\mathbf{Lm}-\mathbf{d}\}} \tag{4.23}$$

$\mathbf{C}_n$ is the covariance matrix of the noise. Under the assumption that the noise is uncorrelated, $\mathbf{C}_n$ becomes a diagonal matrix, a property which would translate to its inverse,

$\mathbf{C}_n^{-1}$. Estimation of the covariance matrix and its inverse is in practice difficult and may not always be performed. As long as the noise variance is relatively stationary throughout the data $\mathbf{d}$, the weighting applied by a diagonal $\mathbf{C}_n^{-1}$ can be incorporated into the regularization parameter λ_1.

The MAP model can easily be calculated through minimization of the negative log-likelihood $-\ln\{p(\mathbf{m}|\mathbf{d})\}$, indeed:

$$\arg\max_{\mathbf{m}} \Big\{p(\mathbf{m}|\mathbf{d})\Big\} = \arg\min_{\mathbf{m}} \Big\{ -\ln\{p(\mathbf{m}|\mathbf{d})\}\Big\} \tag{4.24}$$

$$\mathbf{m}_{MAP} = \arg\min_{\mathbf{m}} \Big\{ -\ln\{p(\mathbf{m}|\mathbf{d})\}\Big\} = \arg_{\mathbf{m}} \Big\{ -\frac{\partial \ln\{p(\mathbf{m}|\mathbf{d})\}}{\partial \mathbf{m}} = 0\Big\} \tag{4.25}$$

Due to the symmetry between deterministic and probabilistic inversion for the special case of a linear forward operator and Gaussian distributed noise, the maximum a posteriori model of (4.23) is the one that minimizes the objective function consisting of an ℓ_2 data norm and the regularization function $f(\mathbf{m})$:

$$\varphi(\mathbf{m}|\mathbf{d}) = \lambda f(\mathbf{m}) + ||\mathbf{W}_d(\mathbf{d} - \mathbf{L}\mathbf{m})||_2^2 = \lambda f(\mathbf{m}) + (\mathbf{d} - \mathbf{L}\mathbf{m})^\dagger \mathbf{W}_d^\dagger \mathbf{W}_d(\mathbf{d} - \mathbf{L}\mathbf{m}) \tag{4.26}$$

$$\lambda := 2\lambda_1$$

Implicity we have introduced the data weighting matrix in order to express the inverse of the noise covariance matrix:

$$\mathbf{W}_d^\dagger \mathbf{W}_d = \mathbf{C}_n^{-1} \tag{4.27}$$

The unconstrained optimization problem therefore reads:

$$\mathbf{m} = \arg\min_{\mathbf{m}\in\mathcal{M}} \{\varphi(\mathbf{m}|\mathbf{d})\} \tag{4.28}$$

It should be noted that this is in fact equivalent to solving the constrained optimization problem:

$$\begin{gathered}\text{minimize } f(\mathbf{m}) \\ \text{subject to } ||\mathbf{W}_d(\mathbf{d} - \mathbf{L}\mathbf{m})||_2^2 \leq \gamma^2\end{gathered}$$

for some value of λ (Van Den Berg and Friedlander, 2008). Here, γ represents a suitable estimate of the 2-norm of the combined effect of data noise and errors in the forward modelling.

We continue the derivation from the unconstrained optimization in equation (4.28).
Application of the conjugate cogradient operator to the constraint function, $f(\mathbf{m})$, yields in elementwise notation:

$$\frac{\partial f(\mathbf{m})}{\partial m_\ell^*} = \sum_i \frac{1}{m_i^* m_i + b^2} m_i \delta_{i,\ell} = \frac{1}{m_\ell^* m_\ell + b^2} m_\ell \tag{4.29}$$

By introducing a diagonal matrix $\{\mathbf{D}(\mathbf{m})\}_{ii} := \frac{1}{m_i^* m_i + b^2}$, the full conjugate cogradient of $f(\mathbf{m})$ reads:

$$\frac{\partial f(\mathbf{m})}{\partial \mathbf{m}^*}\Big|_{\mathbf{m}=const} = \mathbf{D}(\mathbf{m})\mathbf{m} \tag{4.30}$$

The normal equations which retrieve the maximum a posteriori model with sparsity constraints therefore reads:

$$\Big[\lambda \mathbf{D}(\mathbf{m}) + \mathbf{L}^\dagger \mathbf{W}_d^\dagger \mathbf{W}_d \mathbf{L}\Big] \mathbf{m}_{MAP} = \mathbf{L}^\dagger \mathbf{W}_d^\dagger \mathbf{W}_d \mathbf{d} \tag{4.31}$$

Note that equation (4.31) is non-linear with respect to the model parameters, even though the forward model operator is linear and the data norm is an ℓ_2 norm. The nonlinearity is introduced solely by the regularization. Popular solvers for such nonlinear problems are e.g. those from optimization methods, which include Non-Linear Conjugate Gradient, Gauss-Newton or Full-Newton methods. However, because the forward operator is linear, one can avoid the full computational complexity of these methods. In the context of high-resolution Radon transforms, solving the normal equations given in (4.31) by Iteratively Reweighted Least Squares (IRLS) (Scales et al., 1988) is attractive. This approach is discussed in section 4.8.

4.5 Other sparsity norms and a generalization

In the years after Sacchi and Ulrych's article, other model constraints have gained popularity for promoting sparseness, mostly based upon using ℓ_p norms for model-space regularization, for $0 \le p \le 1$. A variant of the objective function (4.26) for an ℓ_q data norm and an ℓ_p model norm is given by:

$$\varphi(\mathbf{m}, p, q|\mathbf{d}) = ||\mathbf{W}_d(\mathbf{d} - \mathbf{L}\mathbf{m})||_q^q + \lambda ||\mathbf{m}||_p^p \tag{4.32}$$

The unconstrained optimization problem reads:

$$\mathbf{m} = \arg\min_{\mathbf{m}\in\mathcal{M}} \{\varphi(\mathbf{m}, p, q|\mathbf{d})\} \tag{4.33}$$

The sparsest model obtainable is the one that minimizes the ℓ_0 model norm. The ℓ_0 norm is defined as the number of nonzero elements in a vector:

$$||\mathbf{m}||_0 := \#\{i : \quad m_i \neq 0\} \tag{4.34}$$

Unfortunately, the ℓ_0 norm is non-convex, leading to a problem which is difficult to solve. The ℓ_1 norm can fortunately be seen as a convex relaxation of the ℓ_0 norm and can therefore be used as an approximation (Candès et al., 2006). Indeed, in Compressive Sensing, medical imaging and geophysics, choosing an ℓ_1 norm is much used for retrieving sparse models. References for the two latter fields of science may include Lustig et al. (2007) and Trad et al. (2003), respectively.

The ℓ_1 model norm reads $||\mathbf{m}||_1 = \sum_i |m_i|$. The relevant sensitivities are given by:

$$\left\{ \begin{array}{ll} \frac{\partial ||\mathbf{m}||_1}{\partial m_\ell} = \sum_i \frac{2}{2} \frac{1}{|m_i|} m_i \delta_{i\ell} = \frac{1}{|m_\ell|} m_\ell, & \mathbf{m} \in \mathbb{R}^M \\ \frac{\partial ||\mathbf{m}||_1}{\partial m_\ell^*} = \frac{1}{2} \frac{1}{|m_\ell|} m_\ell = \left\{ \frac{\partial ||\mathbf{m}||_1}{\partial m_\ell} \right\}^*, & \mathbf{m} \in \mathbb{C}^M \end{array} \right\} \tag{4.35}$$

In the following the constant factor $\frac{1}{2}$ present in the complex case will simply be ignored, and implicitly considered as a part of the regularization parameter.

By defining the model-weighting matrix as a diagonal weighting matrix with elements:

$$\{\mathbf{W}_m\}_{ii}^{\ell_1} = \frac{1}{\sqrt{|m_i|}} \tag{4.36}$$

one can re-write the ℓ_1 model norm as a weighted ℓ_2 norm:

$$||\mathbf{m}||_1^1 = ||\mathbf{W}_m \mathbf{m}||_2^2 = \mathbf{m}^\dagger \mathbf{W}_m^\dagger \mathbf{W}_m \mathbf{m} \tag{4.37}$$

In this formalism, the constraint resulting from assuming a Cauchy prior can also be expressed using a diagonal model space weighting matrix in a weighted ℓ_2 model norm. By this generalization, we express the general objective function for an $\ell_q - \ell_p$ inverse problem as the combination of weighted ℓ_2 norms:

$$\varphi = ||\mathbf{W}_d(\mathbf{d} - \mathbf{L}\mathbf{m})||_2^2 + \lambda ||\mathbf{W}_m \mathbf{m}||_2^2 \tag{4.38}$$

The effect of an ℓ_q data-norm is absorbed into the data weighting matrix $\mathbf{W}_d$ in a sense similar as for the model space constraints.

Discussed choices for the weighting matrices include, here shown for the model-space weighting matrix:

$$\{\mathbf{W}_m\}_{ii} = \left\{ \begin{array}{lr} 1, & \ell_2 \text{ norm. (Gaussian prior)} \\ \frac{1}{\sqrt{|m_i|}}, & \ell_1 \text{ norm. (Laplace prior)} \\ \frac{1}{\sqrt{m_i^* m_i + b^2}}, & \text{Burg's measure of entropy (Cauchy prior)} \end{array} \right\} \tag{4.39}$$

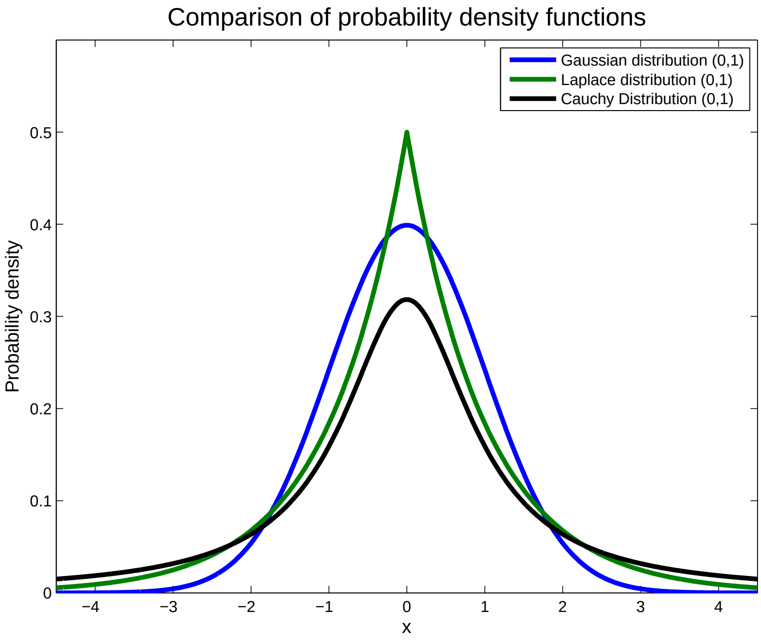
The text in the parenthesis labels the probability distribution assumed on the prior $p(\mathbf{m})$ for the corresponding norm/measure. Only the two latter choices of model space norms can promote sparsity. In order to provide some intuition as to why this is the case, we will consider the three corresponding probability density functions.

Figure 4.1 shows a comparison of a standard Gaussian distribution $\mathcal{N}(0, 1)$ alongside a Laplace distribution and a Cauchy distribution. The two latter have median and mode at $\mathbf{x} = 0$ and scale parameters set equal to unity. One must note that the scale parameters have different interpretations for the distinct pdfs. Therefore, *the plot is only indicative.*

A very important observation available from figure 4.1 is that the Gaussian distribution is only gently peaked around its mean and rapidly approaches zero away from it. By Bayes' rule, the posterior distribution of the model is proportional to $p(\mathbf{m}|\mathbf{d}) \propto p(\mathbf{d}|\mathbf{m})p(\mathbf{m})$. Assumption of a Gaussian model space prior can therefore not retrieve models composed of a few, large coefficents.

The Laplace pdf is notably peaked around $p(\mathbf{x} = 0)$, while still allowing large coefficients. It is therefore much more effective at obtaining models composed of many zero-valued and a few, large elements, i.e. sparse models. Idem, this can be argued for the Cauchy distribution, yet it has a characteristic pdf somewhat different to the Laplace distribution.

Figure 4.1 Comparison of univariate Gaussian, Laplace and Cauchy distributions for the pdf $p(\mathbf{x})$.



4.6 Stabilization of the ℓ_1 norm and its relation to the Huber norm.

The weighting operator for the ℓ_1 norm defined in equation (4.36) is in practice modified slightly in order to avoid the singularity that occurs when $|m_i| \to 0$. A common way to stabilize it is to modify it such that:

$$\{\mathbf{W}_m\}_{ii}^{\ell_1} = \left\{ \begin{array}{ll} \frac{1}{\sqrt{|m_i|}}, & |m_i| > \varepsilon \\ \frac{1}{\sqrt{\varepsilon}} & |m_i| \leq \varepsilon \end{array} \right\} \tag{4.40}$$

ε is here a user chosen parameter introduced in order to avoid division by zero.

The Huber norm, proposed by Huber (1973), is given by the following equation:

$$||\mathbf{m}_i||_{\text{Huber}} := \left\{ \begin{array}{ll} |m_i| - \frac{\varepsilon}{2} & |m_i| > \varepsilon \\ \frac{m_i^2}{2\varepsilon} & |m_i| \leq \varepsilon \end{array} \right\} \tag{4.41}$$

The Huber norm can be interpreted as a hybrid ℓ_1/ℓ_2 norm. For small values of $|m_i|$ it behaves like an ℓ_2 norm, while large entries $|m_i|$ are effectively treated by an ℓ_1 norm. In this setting the threshold value, ε, is not a small value in order to avoid division by zero, but rather the turning point where the norm changes from ℓ_1 to ℓ_2. The gradient of the Huber norm reads:

$$\frac{\partial ||m_i||_{\text{Huber}}}{\partial m_\ell} = \left\{ \begin{array}{ll} \frac{m_i}{|m_i|}\,\delta_{i\ell} & |m_i| > \varepsilon \\ \frac{m_i}{\varepsilon}\,\delta_{i\ell} & |m_i| \leq \varepsilon \end{array} \right\} \tag{4.42}$$

Therefore, the weighting matrix associated with the Huber norm is given by:

$$\{\mathbf{W}_m\}_{ii}^{\text{Huber}} = \left\{ \begin{array}{ll} \frac{1}{\sqrt{|m_i|}}, & |m_i| > \varepsilon \\ \frac{1}{\sqrt{\varepsilon}} & |m_i| < \varepsilon \end{array} \right\} \tag{4.43}$$

The stabilized ℓ_1 weighting matrix entries, eq. (4.41), and the Huber norm weighting matrix entries, eq. (4.43), are evidenly equal. Again, I stress that interpretations of the threshold values ε are to be different. However, this result does imply that choosing a stabilization parameter ε too large will effectively yield an ℓ_2 type regularization term. ε should be chosen in order to maintain a balance between stabilization and sparseness.

4.7 Transformation to the standard form

The regularization through weighting matrices can be imposed in a slightly modified formulation, by transforming the inverse problem to standard form, which is a form in which a general regularization term $||\mathbf{W}_m\mathbf{m}||_2^2$ is replaced by $||\tilde{\mathbf{m}}||_2^2$ (Hansen, 2005). The standard form is useful for situations where one wants to decrease the dependence of the solution upon the hyperparameter λ. This can be achieved by regularization through iteration. In order to achieve the standard form normal equation, right-hand preconditioning is applied to the forward modelling equation:

$$\mathbf{d} = \mathbf{L}\mathbf{W}_m^{-1}\mathbf{W}_m\mathbf{m} = \tilde{\mathbf{L}}\tilde{\mathbf{m}} \tag{4.44}$$

We have implicitly introduced the quantities $\tilde{\mathbf{L}} = \mathbf{L}\mathbf{W}_m^{-1}$ and $\tilde{\mathbf{m}} = \mathbf{W}_m\mathbf{m}$. The effect of the right-hand preconditioning is to set the regularization as part of the modelling, rather than a penalty factor in the objective function. Note that the right-hand preconditioning is only valid for a full-rank matrix $\mathbf{W}_m$. Due to the diagonality and the stabilization of $\mathbf{W}_m$, it suffices to say that the matrix is full rank[2].

By right-hand preconditioning, the null space of the forward model operator is modified. Hence, information that lives in the null space of the operator is added (Nichols, 1997).

The standard-form objective function simply reads:

$$\varphi(\mathbf{d}, \tilde{\mathbf{m}}) = ||\mathbf{W}_d(\mathbf{d} - \tilde{\mathbf{L}}\tilde{\mathbf{m}})||_2^2 + \lambda||\tilde{\mathbf{m}}||_2^2 \tag{4.45}$$

Minimization of (4.45) minimizes the norm of the solution in the transformed model space, $\tilde{\mathcal{M}}$, and not the norm in the original space:

$$\tilde{\mathbf{m}} = \arg\min_{\tilde{\mathbf{m}} \in \tilde{\mathcal{M}}} \varphi(\mathbf{d}, \tilde{\mathbf{m}}) \tag{4.46}$$

The resulting normal equations read:

$$\left[\mathbf{W}_m^{-\dagger}\mathbf{L}^\dagger\mathbf{W}_d^\dagger\mathbf{W}_d\mathbf{L}\mathbf{W}_m^{-1} + \lambda\mathbf{I}\right]\tilde{\mathbf{m}} = \mathbf{W}_m^{-\dagger}\mathbf{L}^\dagger\mathbf{W}_d^\dagger\mathbf{W}_d\mathbf{d} \tag{4.47}$$

The preconditioned operator $\tilde{\mathbf{L}}$ and its adjoint have been written out only for clarity. In the context of using an iterative solver to solve the normal equation (4.47), one can set the hyperparameter $\lambda = 0$ and let the number of iterations play the role of regularization. The approach of regularization through iteration is discussed in section 4.10 and some of the underlying mathematics are reviewed in a Conjugate Gradient setting in Appendix B.

4.8 Iteratively Reweighted Least Squares

The approach used by IRLS to solve equation (4.31) or (4.47) is to solve a sequence of the equation with recursively computed weighting matrices. At each step one therefore approximates the nonlinear normal equations with its corresponding linear version. This

[2] The singular values of a diagonal matrix are the diagonal entries themselves. As long as no diagonal entry approaches zero or infinity, a square diagonal matrix will be full rank.

approach will be demonstrated for the normal equations given in (4.31). The Iteratively Reweighted Least Squares approach to solving the latter equation is:

$$\mathbf{m}^{(k)} = \left[\lambda \mathbf{W}_m^{(k-1)\dagger}\mathbf{W}_m^{(k-1)} + \mathbf{L}^\dagger \mathbf{W}_d^{(k-1)\dagger}\mathbf{W}_d^{(k-1)}\mathbf{L}\right]^{-1}\mathbf{L}^\dagger \mathbf{W}_d^{(k-1)\dagger}\mathbf{W}_d^{(k-1)}\mathbf{d} \qquad k \in \{1, ..., N_{\text{external}}\} \tag{4.48}$$

From hereon, the iterations involved in updating the model and/or data weights are referred to as external iterations. N_{external} is the number of external iterations involved in the IRLS scheme.

At each external iteration of (4.48) the system of normal equations is linear, and can therefore be solved with classical linear systems solvers. In the case that the linear solver involves iterations, we refer to its iterations as internal iterations.

The functions that update $\mathbf{W}_m^{(k)}$ and $\mathbf{W}_d^{(k)}$ are denoted $\chi(\mathbf{m}^{(k)}, \varepsilon_m)$ and $\psi(\mathbf{d}, \mathbf{L}\mathbf{m}^{(k)}, \varepsilon_d)$, respectively.

Algorithm 2 Iteratively reweighted least-squares algorithm

```
Solves the unconstrained $\ell_q - \ell_p$ optimization problem given in equation (4.33).
Input: $\mathbf{d}, \lambda, \varepsilon_m, \varepsilon_d, N_{\text{external}}$
Output: $\mathbf{m}$
$k = 0$ , $\mathbf{m}^{(0)} = 0$, $\mathbf{W}_d^{(0)} = \mathbf{I}$, $\mathbf{W}_m^{(0)} = \mathbf{I}$
while $k < N_{external}$ do
    $\mathbf{m}^{(k+1)} = \left[\lambda \mathbf{W}_m^{(k)\dagger}\mathbf{W}_m^{(k)} + \mathbf{L}^\dagger \mathbf{W}_d^{(k)\dagger}\mathbf{W}_d^{(k)}\mathbf{L}\right]^{-1}\mathbf{L}^\dagger \mathbf{W}_d^{(k)\dagger}\mathbf{W}_d^{(k)}\mathbf{d}$
    $\mathbf{W}_m^{(k)} = \text{diag}[\chi(\mathbf{m}^{(k)}, \varepsilon_m)]$
    $\mathbf{W}_d^{(k)} = \text{diag}[\psi(\mathbf{d}, \mathbf{L}\mathbf{m}^{(k)}, \varepsilon_d)]$
    $k = k + 1$
end
return $\mathbf{m}^{(k)}$
```

Algorithm 2 can handle mixed $\ell_p - \ell_q$ problems through the action of the data and model weighting matrices. The IRLS algorithm for solving a standard-form inverse problem follows directly from this algorithm.

A clear distinction between the types of iterations involved in IRLS, external and internal, is now evident. External iterations update the model weights. Internal iterations attempt to solve the resulting linear normal equations and update the model. In the context of this thesis, the iterative solver Conjugate Gradient Least Squares (CGLS) has been used for internal iterations (Hestenes and Stiefel, 1952).

4.9 Time-Frequency domain Linear Radon transform

Frequency domain forward and adjoint operators have constituted the standard way of performing linear Radon transforms because of their computational advantage over time-domain operators. However, in the context of linear Radon transforms with sparsity constraints, the frequency domain provides some disadvantages. The primary disadvantage is that model weights set in the frequency domain are coupled for all times, which they should not be. Sparse Radon transforms in the frequency domain can often generate artifacts in the situation of strong amplitude inbalance (i.e. in presence of high-amplitude events) (Trad et al., 2003). Furthermore, the hyperparameters in the frequency domain are not easily or intuitively set, and are generally harder to decide by data inspection. This holds in particular for the background power b^2 or the threshold parameter ε in the cases of regularization by minimization of Burg's entropy (Cauchy prior) or ℓ_1 norm, respectively.

The weaknesses of the frequency domain for computing sparse Radon transforms was first recognized by Cary et al. (1998), which proposed to compute time-invariant Radon transforms in the time domain through the application of frequency domain operators. This would allow one to set the model weighting matrices in the $\tau - p$ domain directly, rather than in the $\omega - p$ domain. Furthermore, the hybrid approach still allows one to take advantage of the computational flexibility of frequency domain operators. If chosen, the hybrid approach still allows exploiting the Hermitian Toeplitz symmetry of the operator $\mathbf{L}^\dagger\mathbf{L}$ to perform MVMs via circular convolutions in the Fourier domain.

In order to accomodate the time-frequency domain linear Radon transform in the sense indicated, we consider the modified forward modelling operation:

$$\mathbf{d} = \boldsymbol{\mathcal{F}}^\dagger \mathbf{L} \boldsymbol{\mathcal{F}} \mathbf{V} \mathbf{m} \tag{4.49}$$

$\mathbf{L}$ is still the linear operator performing the inverse linear Radon transform in the frequency domain, yet it is naturally modified from its monofrequency representation. It is simply expanded such that all frequencies are calculated simultaneously. $\boldsymbol{\mathcal{F}}$ is the unitary normalized Fourier transform operator from time to frequency, such that: $\boldsymbol{\mathcal{F}}^\dagger\boldsymbol{\mathcal{F}} = \boldsymbol{\mathcal{F}}\boldsymbol{\mathcal{F}}^\dagger = \mathbf{I}$.

$\mathbf{V}$ is a circulant matrix performing convolution with the source wavelet. Its adjoint is the cross-correlation matrix. By including it into the forward modelling operation we attempt to ensure that:

- **I)** The predicted data have an approximately correct waveform.
- **II)** The action of the operator does not modify the linear Radon transform sampling requirements, and thus does not introduce any aliasing.

Furthermore, the introduction of the source convolution operator enables the representation of a single, constant amplitude plane in the $t - x$ space to map to a single point in the $\tau - p$ domain. In fact, when considering a time-invariant RT with sparsity constraints, the time-frequency domain formulation will promote sparsity not just in the slowness direction, p, but also in intercept time, τ. This is a feat that frequency domain formulations are not able to enforce.

The time-frequency domain unconstrained optimization problem reads:

$$\mathbf{m} = \underset{\mathbf{m}\in\mathcal{M}}{\arg\min}\{\varphi(\mathbf{m}, p, q|\mathbf{d})\} = \underset{\mathbf{m}\in\mathcal{M}}{\arg\min}\{||\mathbf{W}_d(\mathbf{d} - \boldsymbol{\mathcal{F}}^\dagger\mathbf{L}\boldsymbol{\mathcal{F}}\mathbf{V}\mathbf{m})||_q^q + \lambda||\mathbf{m}||_p^p\} \tag{4.50}$$

Due to the fact that $\mathbf{V}$ is a circulant matrix we will make use of its eigendecomposition, with the discrete Fourier basis acting as the matrix of eigenvectors (Golub and Van Loan, 2012):

$$\mathbf{V} = \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V \boldsymbol{\mathcal{F}} \tag{4.51}$$

The resulting normal equations are:

$$\Big[\lambda \mathbf{W}_m^\dagger \mathbf{W}_m + \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \boldsymbol{\mathcal{F}}\boldsymbol{\mathcal{F}}^\dagger \mathbf{L}^\dagger \boldsymbol{\mathcal{F}} \mathbf{W}_d^\dagger \mathbf{W}_d \boldsymbol{\mathcal{F}}^\dagger \mathbf{L} \boldsymbol{\mathcal{F}}\boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V \boldsymbol{\mathcal{F}}\Big]\mathbf{m} = \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \boldsymbol{\mathcal{F}}\boldsymbol{\mathcal{F}}^\dagger \mathbf{L}^\dagger \boldsymbol{\mathcal{F}} \mathbf{W}_d^\dagger \mathbf{W}_d^\dagger \mathbf{d} \tag{4.52}$$

In order to exploit any possible Hermitian Toeplitz symmetry we assume that $\mathbf{W}_d^\dagger \mathbf{W}_d = \alpha^2 \mathbf{I} \quad \forall \alpha \in \mathbb{R}$. The consequence of which is that the following holds only for an ℓ_2 data norm in the presence of uncorrelated, constant variance Gaussian noise. Furthermore, we make use of the property that the Fourier operator is a unitary matrix. The simplified normal equations reduce to:

$$\Big[\lambda \mathbf{W}_m^\dagger \mathbf{W}_m + \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \mathbf{L}^\dagger \mathbf{L} \boldsymbol{\Lambda}_V \boldsymbol{\mathcal{F}}\Big]\mathbf{m} = \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \mathbf{L}^\dagger \boldsymbol{\mathcal{F}} \mathbf{d} \tag{4.53}$$

For completeness, the solution of the standard-form unconstrained optimization problem in the time-frequency formulation for $q = 2$ is then simply:

$$\Big[\lambda \mathbf{I} + \mathbf{W}_m^{-\dagger} \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \mathbf{L}^\dagger \mathbf{L} \boldsymbol{\Lambda}_V \boldsymbol{\mathcal{F}} \mathbf{W}_m^{-1}\Big]\tilde{\mathbf{m}} = \mathbf{W}_m^{-\dagger} \boldsymbol{\mathcal{F}}^\dagger \boldsymbol{\Lambda}_V^\dagger \mathbf{L}^\dagger \boldsymbol{\mathcal{F}} \mathbf{d} \tag{4.54}$$

Because the model-weights and the Radon-model itself are both defined in the $\tau - p$ domain the transformation to standard form is also here given by $\tilde{\mathbf{m}} = \mathbf{W}_m \mathbf{m}$.

4.10 Hyperparameter dependence

The quality of the output of the sparse Radon transforms considered in this thesis all depend heavily on the choice of the hyperparameters. In an optimal setting one should therefore have some robust, preferably automatic way to estimate good choices of the hyperparameters. Other ways to handle this issue is to select solution strategies that do not depend on these parameters. Because the types of hyperparameters introduced by the sparsity promoting regularization are of different nature, a Lagrange multiplier λ and a stabilization parameter b^2/ε, we will treat them in individual subsections.

4.10.1 Stabilization parameters b^2 and ε

For estimating the stabilization parameter some authors use the process of trial and error, such as Trad (2001) used for σ_m^2/b^2 for frequency domain Radon transforms with a Cauchy prior. Others, e.g. Ibrahim and Sacchi (2013) use some percentage of the maximum of the model or data. Trad et al. (2003) use a percentile sorting algorithm where the threshold is determined by the value located at percentile P. In such an approach the hyperparameter dependence is shifted to the given percentile chosen, which must to some degree be chosen via trial and error.

For the frequency-domain implementation of a parabolic RT with a Cauchy model space prior, Hokstad and Sollie (2006) gave another type of data-driven approach to this problem. Their approach was to take some percentile of the maximum frequency-averaged spectral amplitude of the model. This was done at each step of the external iterations in the IRLS sequence. The averaging involves a smoothing effect, which may aid in stabilizing the inversion scheme. In any case, as with all other approaches for choosing the stabilization parameter, it only partially solves the problem of hyperparameter estimation. As much of the focus on sparse Radon Transforms in this thesis is towards hybrid time-frequency methods, robust frequency domain parameter estimation schemes have not been implemented.

Preferences for stabilization parameter
Madagascar already has pre-implemented a percentile sorting function using Hoare's algorithm (Claerbout, 2014) in the function *sf_quantile*. Ioan Vlad's program *sfquantile* demonstrates usage of this functionality. For hybrid time-frequency Radon transforms both percentile sorting and simple trial and error have been used to estimate good values for ε. To some extent both methods are quite equivalent, they assume to know something on the model-space sparseness and they often require some trial and error in assuming either a decent percentile value or a decent value for ε directly. However, the percentage clip method can offer certain diagnostics when echoing the determined thresholds to command line during IRLS (test) runs. Furthermore, it allows a varying threshold between IRLS iterations. One of its biggest pitfalls come to show in situations where the input data, and therefore often the model as well, is quite empty. In such situations proper and improper percentage clip values may be distinguished based on fractions of percentages. Estimating ε directly through trial and error can therefore be an easier approach in particular situations.

4.10.2 The regularization parameter λ

The penalty-parameter λ is perhaps even more difficult to estimate, as it is the parameter that controls the relationship between data-fitting and model regularization. It can be estimated via e.g. the L-curve criterion (Hansen, 2005), but that requires many solutions of the inverse problem for a given range of λ and is therefore impractical and too expensive in terms of computations.

Hansen (2005) recognized that the number iterations of the conjugate gradient algorithm can be exploited to generate a regularizing effect. In order to enable such regularization the inverse problem must be formulated in standard form. Given the unconstrained $\ell_2 - \ell_2$ minimization problem:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \{ ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2 + \lambda ||\mathbf{W}_x \mathbf{x}||_2^2 \} \tag{4.55}$$

This is transported to standard form as per section 4.7 by applying right-hand preconditioning of $\mathbf{A}$ by the matrix $\mathbf{W}_x^{-1}$.

$$\tilde{\mathbf{x}} = \arg\min_{\tilde{\mathbf{x}}}\{||\mathbf{b} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}||_2^2 + \lambda||\tilde{\mathbf{x}}||_2^2\} \tag{4.56}$$

Again, the normal equations that result from equation (4.56) read:

$$[\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} + \lambda\mathbf{I}]\tilde{\mathbf{x}} = \tilde{\mathbf{A}}^T\mathbf{b} \tag{4.57}$$

For the normal equations of a standard problem, the effect of the regularization parameter λ can be approximated by setting $\lambda = 0$ and rather use an appropriate termination condition for the Conjugate Gradient iterations (Hansen, 2005). The results presented by Hansen (2005) for geophysical inverse problems indicate that a termination condition based upon reaching the minimum of the Generalized Cross Validation (GCV) can come close to the effect of L-curve estimation of the most optimal value for λ. One of the key strengths of the GCV function is that it needs no estimate of noise variance. It is based upon the consideration that proper regularization parameter values should not lead to a solution that is sensitive to the elimination of one data point (Haber and Oldenburg, 2000). Furthermore, the observation that is left out should be predicted fairly well. Let $\tilde{\mathbf{x}}_n(\lambda)$ be the minimizer of:

$$\varphi_n = ||\mathbf{r}||_2^2 - r_n^2 + \lambda||\tilde{\mathbf{x}}||_2^2 \tag{4.58}$$

φ_n is a modified objective function where the $n - th$ element is missing. For each regularization parameter λ, the function φ_n can be minimized to yield a solution $\tilde{\mathbf{x}}_n(\lambda)$. The standard Cross-Validation function is defined to be the sum of square differences between predicted data with and without the n-th element (Haber and Oldenburg, 2000):

$$CV(\lambda) := \sum_{n=1}^{N}(r_n(\lambda))^2 \tag{4.59}$$

The Cross-Validation, or rather its minimizer, is not very practical to compute. For the $k - th$ Conjugate Gradient solution $\tilde{\mathbf{x}}_k$, the matrix $\tilde{\mathbf{C}}_k$ is termed the influence matrix and describes how well $\tilde{\mathbf{x}}_k$ predictes the right-hand-side vector $\mathbf{b}$:

$$\tilde{\mathbf{A}}\tilde{\mathbf{x}}_k = \tilde{\mathbf{C}}_k\mathbf{b} \tag{4.60}$$

For this particular choice of the influence matrix, the GCV function is defined as (Favati et al., 2014) (Hansen, 2005):

$$GCV(k) = \frac{||\mathbf{r}_k||_2^2}{[\text{Trace}(\mathbf{I} - \tilde{\mathbf{C}}_k)]^2} \tag{4.61}$$

In this formulation, the GCV function is linearly proportional to an estimate of the mean predictive error. Its minimizer can be used as an approximation of the optimal number of iterations, k_{opt}. From a probabilistic point of view, the denominator contains the degrees of freedom of the Generalized Cross Validation function.

The diagonal of $\tilde{\mathbf{C}}_k$ contains the Conjugate Gradient filter factors (Favati et al., 2014). The filter factors are however impractical to calculate for large matrices $\tilde{\mathbf{A}}$, as it requires the knowledge of the singular values of $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}$ projected onto the k-th step Krylov subspace $\mathcal{K}_k(\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}, \tilde{\mathbf{A}}^T\mathbf{b})$. Approximate expressions to the trace of the influence matrix are therefore often sought.

Trad et al. (2003) uses a simplified denominator for the GCV function, which is termed the *KA* approach by Favati et al. (2014). It reads:

$$GCV_{KA}(k) \approx \frac{\sum\limits_{j=1}^{N_d} [r_j^{(k)}]^2}{(N_d - k)^2} \tag{4.62}$$

For 2D problems Favati et al. (2014) found that the KA approach to computing the trace of the influence matrix lead to a GCV function which in most cases had no minimum. In the short note below follows a discussion on the GCV function and the choice of CGLS algorithm, as well as some findings related to the implementation presented in this thesis.

Other stopping conditions for regularizing CG iterations also appear in the literature. Ibrahim and Sacchi (2013) used primarily a stopping criterion based on misfit change between Conjugate Gradient iterations, while Liu and Sacchi (2004) used a condition based on the value of the residual norm. The latter examples go on to show that any stopping condition is not entirely exact and that there may be room and need for empirical testing.

As a heuristic, one can also simply use a predetermined number of iterations as stopping criterion and simply adjust it as needed. The amount of regularization imposed is then directly transported from λ to N_{iter}, which at first might seem hardly an improvement. However, N_{iter} is a natural number and imposing regularization this way will also remove any dependence on choice of stopping conditions that must be applied for solving a system of linear normal equations. Furthermore, by first testing small values of N_{iter} one can determine a suitable value within small amounts of (computational) time.

Other approaches to resolving hyperparameter dependence

Other approaches to solving the dependence on the regularization parameter λ have also appeared in the literature in recent years. As an example, Gholami and Aghamiry (2017) propose the *Iteratively re-weighted and refined least squares* (IRRLS) algorithm. In this formulation, λ is allowed to vary between external iterations. Its value is estimated by using the secant method for root finding, in order to concentrate around a solution that satisfies either a data or model space constraint, i.e. a target misfit or model norm size. Implementation and testing of such automatic parameter selection algorithms is beyond the scope of this thesis, but perhaps future works should address them.

A note on the GCV function and the CGLS algorithm chosen

Trad et al. (2003) used the standard formulation, set $\lambda = 0$ and let the termination condition based upon the KA-approximate GCV function play the role of the regularization.

However, there is discrepancy/drawback with this approach. Only variants of CGLS that do not explicitly form matrix-vector products $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{v}$ have data-space residuals, $\mathbf{r}$, available. The most known example of this is what is referred to by Björck et al. (1998) as CGLS1. The variants that do explicitly form $\tilde{\mathbf{A}}^T\tilde{\mathbf{A}}\mathbf{v}$ instead, do not have data-space residuals available, but rather the residuals of the normal equations: $\mathbf{s} = \tilde{\mathbf{A}}^T\mathbf{b} - [\tilde{\mathbf{A}}^T\tilde{\mathbf{A}} + \lambda\mathbf{I}]\tilde{\mathbf{x}}$. This variant is referred to by Björck et al. (1998) as CGLS2. Though the latter suffers more from finite precision errors, in the context of linear Radon transforms we would like to be able to exploit the symmetry of the matrix $\tilde{\mathbf{L}}^\dagger\tilde{\mathbf{L}}$ to perform matrix-vector multiplications using circular convolutions. Minimizing the amount of operations needed is especially relevant for hybrid time-frequency domain implementations.

In theory, only the squared 2-norm of the residual, $||\mathbf{r}^{(k)}||_2^2$, is needed for calculating the KA approximation to the GCV. The conjugate gradient data residuals can be written as:

$$\mathbf{r}^{(k+1)} = \mathbf{r}^{(k)} - \alpha^{(k+1)}\mathbf{A}\mathbf{p}^{(k+1)} = \mathbf{r}^{(0)} - \sum_{j=1}^{k+1}\alpha^{(j)}\mathbf{A}\mathbf{p}^{(j)} \tag{4.63}$$

The squared 2-norm of the residuals at each step therefore becomes:

$$||\mathbf{r}^{(k)}||_2^2 = (\mathbf{r}^{(k)})^T\mathbf{r}^{(k)} = \left[\mathbf{r}^{(0)} - \sum_{i=1}^{k}\alpha^{(i)}\mathbf{A}\mathbf{p}^{(i)}\right]^T\left[\mathbf{r}^{(0)} - \sum_{j=1}^{k}\alpha^{(j)}\mathbf{A}\mathbf{p}^{(j)}\right] \tag{4.64}$$

Due to the $\mathbf{A}^T\mathbf{A}$ orthogonality of the $\mathbf{p}^{(k)}$ vectors: $\mathbf{p}^{(i)}\mathbf{A}^T\mathbf{A}\mathbf{p}^{(j)} = 0 \quad , i \neq j$ (Shewchuk et al., 1994), this simplifies to:

$$||\mathbf{r}^{(k)}||_2^2 = ||\mathbf{r}^{(0)}||_2^2 - 2\sum_{j=1}^{k}\alpha^{(j)}(\mathbf{s}^{(0)})^T\mathbf{p}^{(j)} + \sum_{j=1}^{k}(\alpha^{(j)})^2(\mathbf{p}^{(j)})^T\mathbf{A}^T\mathbf{A}\mathbf{p}^{(j)} \tag{4.65}$$

Where we have also used the fact that $\mathbf{r}^T\mathbf{A}\mathbf{p} = \left(\mathbf{A}^T\mathbf{r}\right)^T\mathbf{p} = \mathbf{s}^T\mathbf{p}$.

Therefore, in theory the KA approach could be used within the CGLS2 framework using only circular convolutions to perform matrix-vector multiplications. **The proposed approach has been implemented and tested. Tests demonstrated that it does not work.**. In these tests, using the $\ell_2 - \ell_1$ hybrid time-frequency transform, the approximate GCV function did not show any minimum. There are two underlying reasons for this. Interestingly, they both point at the same issue, and are listed below.

- **I)** The $\mathbf{A}^T\mathbf{A}$ orthogonality of the $\mathbf{p}^{(j)}$ vectors holds only in infinite precision. Therefore, in finite-precision implementations this type of orthogonality is lost after a few iterations and the expression for the residual norm is not strictly valid.
- **II)** Secondly, and perhaps even more important, the KA approximation to the GCV function has been found by Favati et al. (2014) to be an arguable approximation. Experimentally they found that for 1D problems this approximation could be defendable, even though it was outperformed by more sophisticated ways to approximate

> the trace of the influence matrix. For 2D problems the KA approach was found to be very unreliable, as in most cases it led to a GCV function which showed no minimum. The KA approach hinges on the analytic property of the CG iterations that the residual of normal equations $\mathbf{s}^{(k)}$ are mutually orthogonal. In finite precision however, the $\mathbf{s}^{(k)}$ lose their orthogonality after a few steps and the KA approach cannot give the correct degrees of freedom for the GCV function.

Evidently, finite precision limits the practical validity of not only equation (4.65) but also of the KA approximation, equation (4.62). Based upon my experimental results and the theoretical analysis and results by Favati et al. (2014), using the KA-approximate GCV with reccured $||\mathbf{r}^{(k)}||_2^2$ can not be recommended. Instead, if one wishes to use the GCV function for stopping criterion for regularizing CG iterations, one should use a more sophisticated GCV approximation. A good discussion on this is found in Favati et al. (2014). In particular, the Incomplete Derivative (ID) approach, equivalent to a Monte-Carlo based trace estimation method, appears as an attractive algorithm.

Preferences for regularization parameter
Given that explicitly calculating optimal values for λ via e.g. the L-curve criterion is impractical for large data and model vectors, this approach was never realistically considered. Therefore, the main priority has been to find a good way to perform regularizing CG iterations. Quite some time was spent on deriving expressions from CGLS2 in order to use the KA-approach to calculate the GCV function. It was therefore disappointing that the calculated GCV function showed no minimum, however the discussion and research on this topic is quite interesting. Transformation of the inverse problem to standard-form showed somewhat improved convergence properties for nonzero values of the regularization parameter. In the setting of regularization by iteration it was discovered that simply setting a pre-determined number of iterations gave an intuitive way to perform regularization. With the current experiences in mind, the author would have liked to explore the usage of a more robust approach to calculating the GCV function. In that case, however, it would be necessary to dismiss matrix-vector calculations via circular convolutions, and therefore the whole modelling code would have to be rebuilt. In a future work it would be interesting to do exactly this. For this thesis the current approach will have to suffice.

4.11 Implementational notes: Fast Matrix-Vector Multiplication

This small section intends to shortly describe how to perform matrix-vector multiplications with the matrix $\mathbf{L}^\dagger\mathbf{L}$ with an $\mathcal{O}(N\log(N))$ computational complexity instead of the $\mathcal{O}(N^2)$ complexity valid for general matrices. This is possible by exploiting the Hermitian Toeplitz symmetry of the matrix and the intimate relations between Circulant and Toeplitz matrices.

4.11.1 Toeplitz and Circulant Matrices

DEFINITION (Complex Toeplitz matrix) *Suppose* $\mathbf{T} \in \mathbb{C}^{N\times N}$. *The matrix is Toeplitz iff. there exist scalars* $t_{-N+1}, ..., t_0, ..., t_{N-1}$ *such that* $\mathbf{T}_{i,j} = t_{j-i} \quad \forall(i,j)$ *and the Hermitian structure* $t_{-j} = t_j^*$ *holds. Hence, with* $N = 4$*:*

$$\mathbf{T} = \begin{pmatrix} t_0 & t_1 & t_2 & t_3 \\ t_1^* & t_0 & t_1 & t_2 \\ t_2^* & t_1^* & t_0 & t_1 \\ t_3^* & t_2^* & t_1^* & t_0 \end{pmatrix} \tag{4.66}$$

Toeplitz matrices are completely determined from $2N - 1$ values, eliminating the need for explicit matrix storage.

DEFINITION (Circulant Matrix) *Let* $\mathbf{C} \in \mathbb{C}^{N\times N} \vee \mathbf{C} \in \mathbb{R}^{N\times N}$ *be a matrix with the form*

$$\mathbf{C} = \begin{pmatrix} c_0 & c_{N-1} & c_{N-2} & \cdots & c_1 \\ c_1 & c_0 & c_{N-1} & \cdots & c_1 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_{N-1} & c_{N-2} & c_{N-3} & \cdots & c_0 \end{pmatrix} \tag{4.67}$$

It is known as a circulant matrix. As with a Toeplitz matrix, a Circulant matrix is completely determined from its first column. In fact, Circulant matrices are considered special classes of Toeplitz matrices (Golub and Van Loan, 2012)

Fourier transform based Circulant matrix vector multiplication.
Any Circulant matrix can be written on the eigenvalue decomposition given by (Golub and Van Loan, 2012):

$$\mathbf{C} = \mathcal{F}^\dagger \mathbf{\Lambda} \mathcal{F} \tag{4.68}$$

$$\mathbf{\Lambda_c} \coloneqq \mathrm{diag}(\mathcal{F}\mathbf{c}) \tag{4.69}$$

$\mathcal{F}$ is the unitary discrete Fourier basis for a vector of length N and $\mathbf{c}$ is the first column of the circulant matrix $\mathbf{C}$. Hence, a matrix-vector product of the form $\mathbf{b} = \mathbf{Cz}$ can be computed as:

$$\begin{aligned} \tilde{\mathbf{z}} &= \mathcal{F}\mathbf{z} \\ \tilde{\mathbf{c}} &= \mathcal{F}\mathbf{c} \\ \tilde{\mathbf{b}} &= \tilde{\mathbf{c}} \odot \tilde{\mathbf{z}} \\ \mathbf{b} &= \mathcal{F}^\dagger \, \tilde{\mathbf{b}} \end{aligned} \tag{4.70}$$

The operation $\odot$ denotes element-wise multiplication. Hence, matrix-vector multiplication involving circulant matrices can be performed with the asymptotic cost function

$\mathcal{O}(N \log N)$ via Fast Fourier Transforms.

Fourier transform based Toeplitz matrix vector multiplication.
The Fast Fourier Transform based matrix vector multiplication can be translated to Toeplitz matrices by recognizing the intimate relationship between the two matrix classes.
In general, if $\mathbf{T}_{i,j} = t_{j-i}$ is an $N \times N$ complex Toeplitz matrix, then $\mathbf{T}$ can be considered a subset of a larger, circulant matrix: $\mathbf{T} = \mathbf{C}(0 : N-1, 0 : N-1)$. $\mathbf{C} \in \mathbb{C}^{(2N-1)\times(2N-1)}$ is a circulant with:

$$c_i = \left\{ \begin{array}{ll} t_i, & \text{for } 0 \leq i \leq N-1 \\ t^*_{i-N} & \text{for } N \leq i < 2N-1 \end{array} \right\} \tag{4.71}$$

Then, note that for the matrix vector product $\mathbf{b} = \mathbf{T}\mathbf{z}$, if $\mathbf{z}(N : 2N-2) = 0$, then $\mathbf{b}(0 : N-1) = \{\mathbf{C}\ \mathbf{z}\}(0 : N-1)$. This shows that Toeplitz matrix vector products can also be computed at the computational complexity of Circulant matrices. This is done by embedding the Toeplitz matrix into a Circulant matrix and zero-padding the vector $\mathbf{z}$. The output, $\mathbf{b}$, is truncated to original size after inverse transformation. Explicitly for $N = 4$, this yields the system:

$$\mathbf{b} = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ b_3 \\ \times \\ \times \\ \times \end{pmatrix} = \left(\begin{array}{cccc|ccc} t_0 & t_1 & t_2 & t_3 & t^*_3 & t^*_2 & t^*_1 \\ t^*_1 & t_0 & t_1 & t_2 & t_3 & t^*_3 & t^*_2 \\ t^*_2 & t^*_1 & t_0 & t_1 & t_2 & t_3 & t_2 \\ t^*_3 & t^*_2 & t^*_1 & t_0 & t_1 & t_2 & t_3 \\ \hline t_3 & t^*_3 & t^*_2 & t^*_1 & t_0 & t_1 & t_2 \\ t_2 & t_3 & t^*_3 & t^*_2 & t^*_1 & t_0 & t_1 \\ t_1 & t_2 & t_3 & t^*_3 & t^*_2 & t^*_1 & t_0 \end{array}\right) \begin{pmatrix} z_0 \\ z_1 \\ z_2 \\ z_3 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

The vertical and horizontal lines in equation (4.11.1) are added purely for demonstration. They delimit the left-and uppermost part, consisting of the original Toeplitz matrix, from the parts that are augmented in order to form the resulting circulant matrix.

Fourier transform based Fast Matrix Vector multiplication for linear Radon transforms
The complex hermitian matrix $\mathbf{L}^\dagger\mathbf{L}$ is a Toeplitz matrix, c.f (3.26) in chapter 3. By choosing the CGLS2 algorithm (Björck et al., 1998), all matrix-vector multiplications in the Conjugate Gradient algorithm are performed with this Toeplitz matrix. Via equation (4.71) the resulting Circulant matrix is formed and matrix-vector multiplications are performed via FFTs. For time-frequency domain implementations it is advisable to modify the modelling operation (4.49) such that it includes a unitary transposition matrix and its inverse. By doing so, the resulting matrix $\mathbf{L}^\dagger\mathbf{L}$ is a block diagonal matrix consisting of Toeplitz-submatrices (one per frequency). The transposition matrix is never explicitly formed, it is

handled by the FFT operator.

A short note on numerical precision
For time-frequency domain implementations the author notes that using single-precision floating point numbers dramatically slowed convergence. Comparing the FFT based matrix-vector multiplication with a double-precision MATLAB implementation, discrepancies of up to a few percent were observed. In a Conjugate Gradient setting the stated level of round-off errors is unacceptable. Such problems were never seen in frequency-domain implementations, indicating that the time-frequency domain complex Hermitian matrix has a considerably higher condition number than its frequency-domain counterpart. After re-implementation, using double precision floating point numbers, the numerical round-off errors returned to acceptable levels and the CG algorithm converged.

4.12 Examples of implementational results

The author of this thesis has implemented an $\ell_2 - \ell_1$ hybrid time-frequency domain sparse linear Radon transform. It is written in the **C** programming language (Kernighan and Ritchie, 2017), utilizing the Madagascar API (Madagascar Development Team, 2012) for input and output routines. Fast Fourier Transforms are handled by the FFTW3 library (Frigo and Johnson, 2005).

This section attempts to showcase some of the potential which sparse linear Radon transforms can unlock. In consideration are two examples of transforms calculated using the implemented $\ell_2 - \ell_1$ hybrid time-frequency domain sparse RT. The inputs to the two examples are shown together in figure 4.2. We will compare the Radon gathers to those calculated with an $\ell_2 - \ell_2$ frequency domain linear RT, which is referred to as a "standard" linear Radon transform.

Example A: Decomposition of planes
Decomposition of planes is an especially relevant measure of the resolution of a linear RT. Subject to an optimal transformation, a constant amplitude plane defined by $t = \hat{\tau} + \hat{p}x$ should decompose to the point $(\hat{\tau}, \hat{p})$. The two planes in the input, shown in the lefthand panel of figure 4.2, should decompose to two points $(\tau_1, p_1) = (0.4\ s, 0.0\ s/km)$ and $(\tau_2, p_2) = (0.35\ s, 0.1\ s/km)$, respectively. The frequency domain $\ell_2 - \ell_2$ transform, calulculated via *sfradon*, and the Radon gather calculated via the sparse RT are shown in figure 4.3. As expected, the standard transform suffers from aperture artifacts, while it also has a few other artifacts present. The sparse Radon transform focuses much better to two points in Radon space. It should be noted that the transform parameters could possibly be tweaked even more in order to focus exactly to two points.

Example B: Offset transform of symmetric shot gather
In order to show how the high resolution RT performs on actual seismic data, a transform of a simple shot gather is demonstrated in this section. The input consists of two primary reflections and is shown in the righthand panel of figure 4.2. The resulting Radon gathers are shown in figure 4.4. From left to right, they are calculated via a frequency domain

$\ell_2 - \ell_2$ transform, using *sfradon*, and the $\ell_2 - \ell_1$ hybrid time-frequency domain transform, respectively. As expected, the standard transform naturally shows artifacts related to the limited aperture, which here manifest in the shape of lines. The sparse Radon transform is completely free from this type of artifact. A slice through $p_0 = 0\ s/km$ is shown in figure 4.5. It confirms that indeed the sparse hybrid time-frequency transform promotes sparsity in time as well as slowness.

Figure 4.2 Inputs for example A (left) and example B (right) tests of the sparse, hybrid time-frequency domain Radon transform.

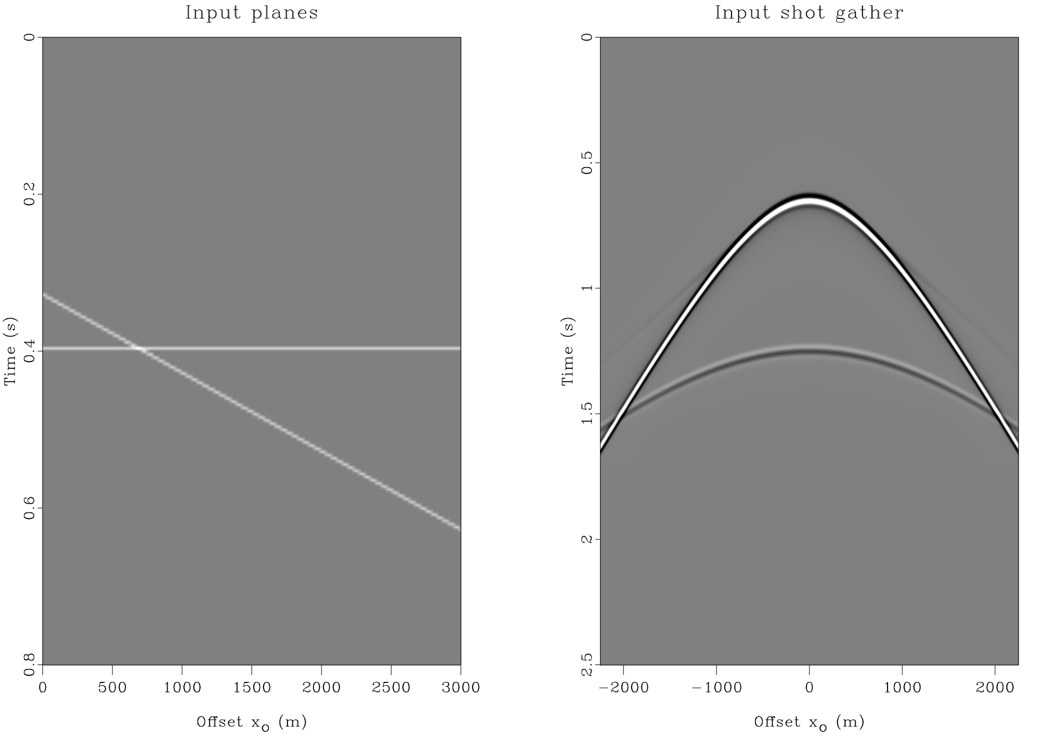


Figure 4.3 Resolution comparison of decomposition of two planes using a standard $\ell_2 - \ell_2$ linear RT (left) and an $\ell_2 - \ell_1$ hybrid time-frequency domain linear RT (right)

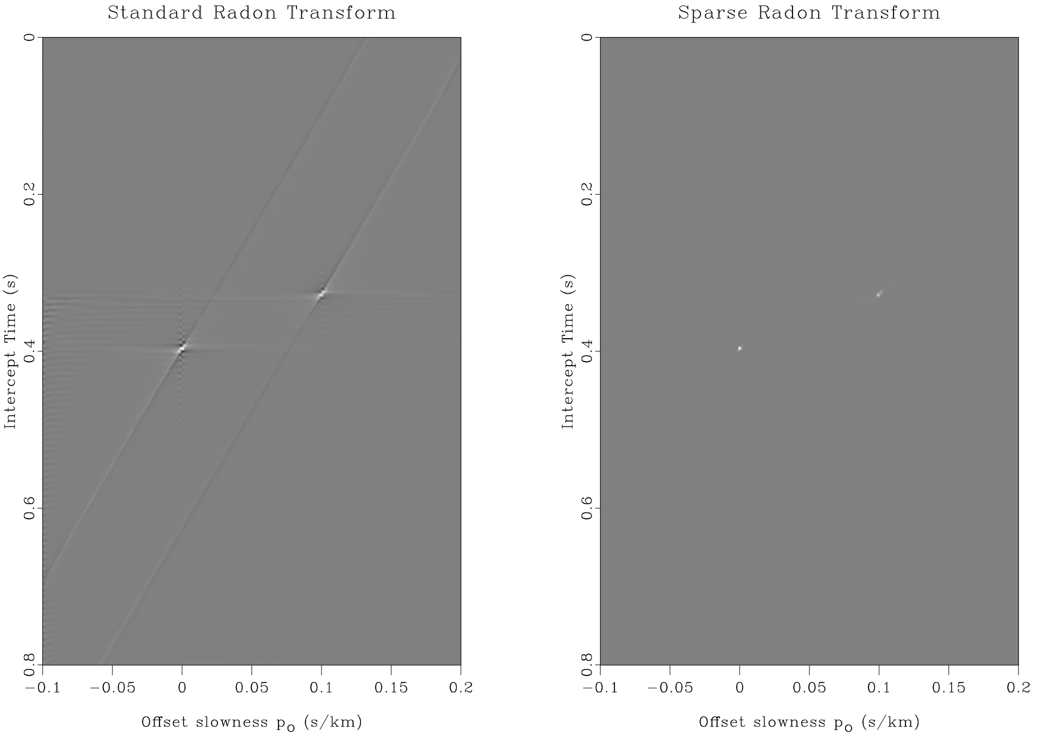


Figure 4.4 Resolution comparison of transformation of a simple shot gather using a standard $\ell_2 - \ell_2$ linear RT (left) and an $\ell_2 - \ell_1$ hybrid time-frequency domain linear RT (right).

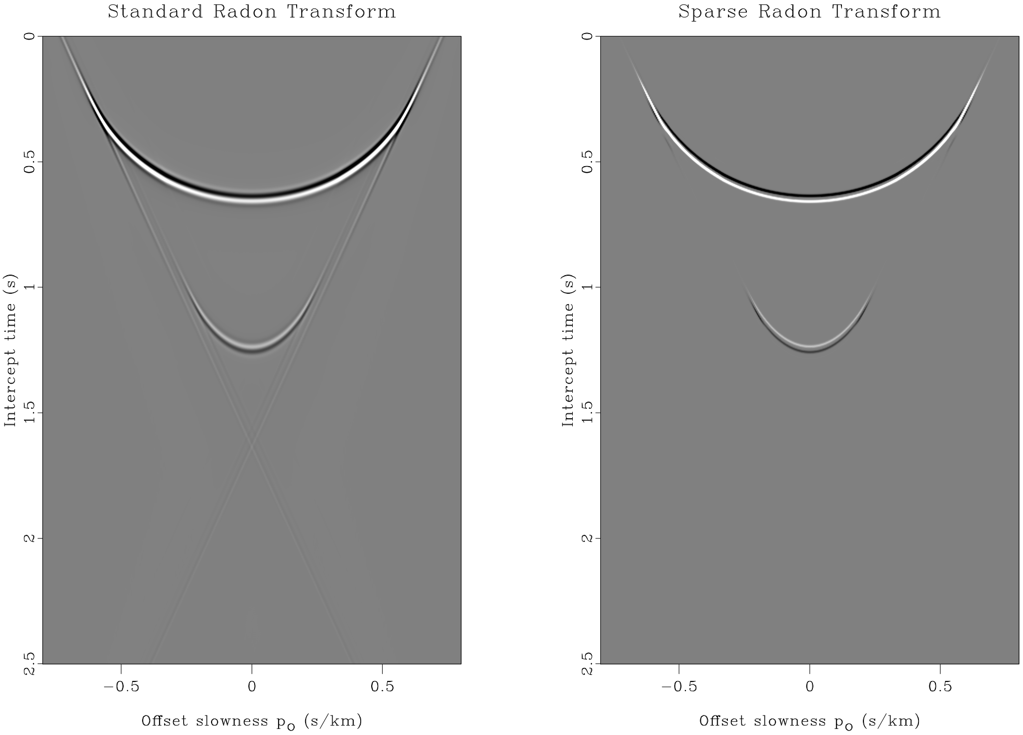
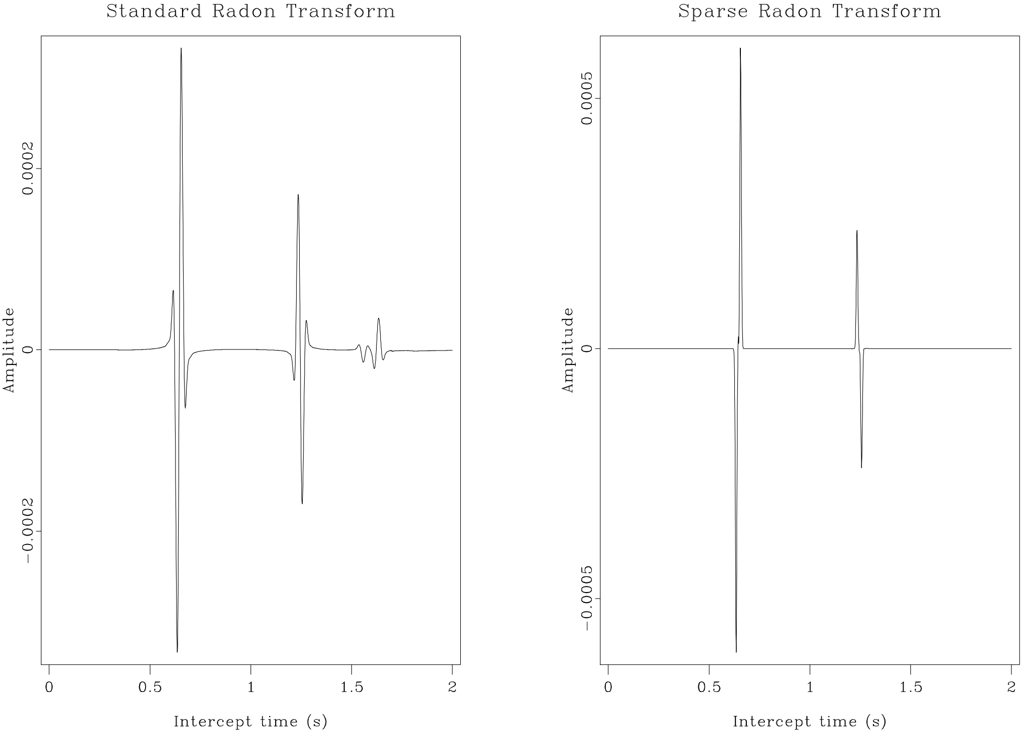


Figure 4.5 A $p_o = 0.0\ s/km$ slice through figure 4.4.



4.13 Limitations and possible improvements

This section will mainly discuss the limitations in the proposed implementation of the hybrid time-frequency $\ell_2 - \ell_1$ high-resolution RT. One of its weaknesses may lie in the properties that the use of an ℓ_2 data norm will make the transformation more sensitive to outliers and erratic (non-Gaussian noise) than more robust data norms, e.g. the ℓ_1 or Huber norm. On this note it is however important to note that the article by Ibrahim and Sacchi (2013) demonstrated that it is often difficult to enforce robustness and sparseness simultaneously. In terms of model space regularization, the approximation of the ℓ_0 norm with the ℓ_1 norm will in many cases retrieve the same model (Candès et al., 2006). In cases where this is not applicable, the ℓ_0 norm should yield improved results. However, as a non-convex optimization problem it gives a problem of a much more complex nature.

As a further improvement one can use pure time-domain operators instead of the hybrid-time frequency formulation. As stated by Trad et al. (2003), time domain operators are expected to produce the most sparse results. They can also provide flexibility with regards to the dimensions of the input and output space. The most obvious drawback of using time domain operators is the increased asymptotic computational cost function. In order to build (relatively) efficient time domain operators, one needs to build them in a sparse manner. This involves storing and computing only the parts of input and output space needed, leading to sparse Matrix storage and multiplications with vectors.

Regardless of which sparsity norms and operators that are chosen, the choice of regularization and stabilization parameter will have great impact on the computed solutions. The current implementation is somewhat hamstrung by this aspect. For inverse problems with linear forward model operators, the use of regularizing Conjugate Gradient iterations as described by Hansen (2005) is an elegant proposition to circumvent the regularization parameter alltogether. In order for regularization by iteration to give results similar to what is obtained using highly optimal regularization parameters, a proper termination condition is required. The Generalized Cross Validation function can yield what is sought (Hansen, 2005). The most simple approximation to the GCV function is the KA-approximation (Favati et al., 2014). Thorough attempts were made to incorporate the latter in a setting using the CGLS2 algorithm (Björck et al., 1998) while performing all matrix-vector multiplications using circular convolutions, i.e. by exploiting the Toeplitz structure of $\mathbf{L}^{\dagger}\mathbf{L}$. This did not lead to success as several of the identities used in the derivations do not hold in a finite precision setting. In order to make regularizing CG iterations work optimally for multidimensional inverse problems using the GCV function to determine termination, two modifications need to be made. The Toeplitz structure of $\mathbf{L}^{\dagger}\mathbf{L}$ can not be exploited and the CGLS1 algorithm (Björck et al., 1998) should be used. Secondly, a more robust approximation to the GCV function needs to be considered. This is properly reviewed in Favati et al. (2014).

Chapter 5

Implementational aspects of the Internal multiple predictor

The inverse scattering series based internal multiple predictors are algorithms that are very demanding in terms of computations. In order to be able to realize an implementation that has practical value it is important to take all possible steps to minimize program run-time. The first section attempts to elaborate on the high computational cost function of the 2D ISS internal multiple predictor, and relates this to another costly algorithm, namely 3D SRME. Sections 5.2 through 5.5 show that it is possible to reduce the complexity of the algorithm substantially, mostly through mathematical analysis and, in some cases, through a priori knowledge of the angular dip of reflectors in the medium. Section 5.7 discusses relevant programming choices in order to yield a high performance implementation, given the algorithm from the preceding sections. Doing so, it introduces some concepts from computer science along the way. Section 5.9 is the last section of this chapter. It gives some benchmarks of the code optimizations presented in the section preceding it.

5.1 The computational complexity of the Internal multiple predictor.

For completeness we re-iterate the leading contribution to the first order internal multiple predictor, b_3 in the coupled plane wave domain for an arbitrary 2D earth:

$$b_3(p_r, p_s, \omega) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} dp_1 e^{j\omega q_1(z_s - z_r)} \int_{-\infty}^{\infty} dp_2 e^{j\omega q_2(z_r - z_s)} \int_{-\infty}^{\infty} d\tau_1 e^{j\omega\tau_1} b_1(p_r, p_1, \tau_1)$$
$$\times \int_{-\infty}^{\tau_1 - \epsilon} d\tau_2 e^{-j\omega\tau_2} b_1(p_2, p_1, \tau_2) \times \int_{\tau_2+\epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_2, p_s, \tau_3) \tag{5.1}$$

As it reads, the asymptotic cost function for all possible p_r, p_s, ω is:

$$\hat{C} = \mathcal{O}(N_\omega \times N_{p_s} \times N_{p_r} \times N_{p_s} \times N_{p_r} \times N_\tau \times N_\tau \times N_\tau) \propto N^8 \tag{5.2}$$

$\hat{C}$ does not differentiate between the different types of floating point operations involved[1]. Efficient evaluation of the multidimensional integral that computes b_3 is of serious importance due to the high computational cost of the algorithm.

Comparison to 3D SRME

A surface-related multiple model in the full 3D[2] sense through the theory of SRME can read Dragoset et al. (2010):

$$\mathcal{M}(x_r, y_r, x_s, y_s, \omega) = r_0 \sum_{x_k, y_k} \mathcal{R}(x_r, y_r, x_k, y_k, \omega)\mathcal{P}^-(x_k, y_k, x_s, y_s, \omega) \tag{5.3}$$

where $\mathcal{R}(x_r, y_r, x_k, y_k, \omega)$ denotes the response of the earth in absence of a free surface, and $\mathcal{P}^-$ is the upgoing part of the recorded wavefield. Both are recorded at vertical position $z = 0$, coinciding with the location of the free surface for the latter term. For simplicity assume that source and receiver coordinates and angular frequency dimensions contain roughly the same amounts of samples N. Then, the associated asymptotic cost function is: $\hat{C}^{SRME,3D} = \mathcal{O}(N^7)$. We have ignored the steps necessary to compute the reflection response $\mathcal{R}$. We can however state that the baseline Inverse Scattering Series internal multiple predictor in 2D has a computational complexity comparable to that of surface related multiple prediction in 3D. The latter is known as a highly expensive algorithm in terms of computations.

With this in mind, in order to implement a tool that has any practical value, one must

- **a)** make good efforts in order to reduce the algorithmic complexity

and

- **b)** write a code that is high-performance and utilizes, to the best of its ability, the hardware it will run on.

5.2 Algorithmic optimizations through mathematical analysis.

Most of the optimizations presented in this section were originally discovered by Kaplan et al. (2004), who gave a good discussion on this subject. Only a few new optimizations are present in this discussion.

Lemma 1, presented below, will be demonstrated particularly useful for reducing the computational complexity of computing the innermost three integrals in the internal multiple predictor.

[1] That is, there is no distinction between multiplication, addition, division, exponential evaluation etc.

[2] That is, for a three-dimensional earth. The dataset is five-dimensional.

Lemma 1

Let $f(t)$ and $g(t)$ be any integrable functions. Then:

$$\int_{-\infty}^{\infty} dt\ f(t) \int_{-\infty}^{t-\epsilon} dt'\ g(t') = \int_{-\infty}^{\infty} dt\ g(t) \int_{t+\epsilon}^{\infty} dt'\ f(t') \tag{5.4}$$

The proof of Lemma 1 is demonstrated in appendix C.

We define the variable R to consist of the innermost three integrals:

$$R(p_r, p_s, p_1, p_2, \omega) := \int_{-\infty}^{\infty} d\tau_1 e^{j\omega\tau_1} b_1(p_r, p_1, \tau_1)$$
$$\times \int_{-\infty}^{\tau_1-\epsilon} d\tau_2 e^{-j\omega\tau_2} b_1(p_2, p_1, \tau_2) \times \int_{\tau_2+\epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_2, p_s, \tau_3) \tag{5.5}$$

As such, when re-arranging the sums over the auxilliary source and receiver slownesses, equation 5.1 can be arranged as:

$$b_3(p_r, p_s, \omega) = \frac{1}{(2\pi)^2} \int_{-\infty}^{\infty} dp_1 e^{j\omega q_1(z_s - z_r)} \int_{-\infty}^{\infty} dp_2 e^{j\omega q_2(z_r - z_s)} R(p_r, p_s, p_1, p_2, \omega) \tag{5.6}$$

The quantity R is decomposed so that Lemma 1 may be used:

$$f(\tau) := e^{j\omega\tau} b_1(p_r, p_1, \tau) \tag{5.7}$$

$$g(\tau) := e^{-j\omega\tau} b_1(p_2, p_1, \tau) \times \int_{\tau+\epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_2, p_s, \tau_3) \tag{5.8}$$

By Lemma 1 we can re-express R as:

$$R(p_r, p_s, p_1, p_2, \omega) = \int_{-\infty}^{\infty} d\tau_1 g(\tau_1) \int_{\tau_1+\epsilon}^{\infty} d\tau_2 f(\tau_2)$$
$$R(p_r, p_s, p_1, p_2, \omega) = \int_{-\infty}^{\infty} d\tau_1 e^{-j\omega\tau_1} b_1(p_2, p_1, \tau_1)$$
$$\times \int_{\tau_1+\epsilon}^{\infty} d\tau_3 e^{j\omega\tau_3} b_1(p_2, p_s, \tau_3) \times \int_{\tau_1+\epsilon}^{\infty} d\tau_2 e^{j\omega\tau_2} b_1(p_r, p_1, \tau_2) \tag{5.9}$$

The two innermost integrals are now decoupled. By simply re-naming τ_3 to τ_2, this can be seen even better:

$$R(p_r, p_s, p_1, p_2, \omega) = \int_{-\infty}^{\infty} d\tau_1 e^{-j\omega\tau_1} b_1(p_2, p_1, \tau_1)$$
$$\times \left[\int_{\tau_1+\epsilon}^{\infty} d\tau_2 e^{j\omega\tau_2} b_1(p_2, p_s, \tau_2) \right] \times \left[\int_{\tau_1+\epsilon}^{\infty} d\tau_2 e^{j\omega\tau_2} b_1(p_r, p_1, \tau_2) \right] \tag{5.10}$$

Decoupling of the two integrals makes the new asymptotic cost function $\mathcal{O}(N^7)$, constituting a saving of one order of magnitude N.

5.2.1 Recognizing recursion patterns

In the discrete sense, the integrations involved in constructing b_3 correspond to summations. For certain methods of discrete integral evaluation, there is a recursive structure in the terms involved in the integration over intercept time. Exploitation of the recursive structure can further reduce the asymptotic complexity. For studying this, we define the functions[3] f, g and h such that

$$\begin{aligned} R(p_1, p_2, p_r, p_s, \omega) &:= \int_{-\infty}^{\infty} d\tau_1 f(p_1, p_2, \omega, \tau_1) \\ \times \int_{\tau_1+\epsilon}^{\infty} d\tau_2 h(p_2, p_s, \omega, \tau_2) &\times \int_{\tau_1+\epsilon}^{\infty} d\tau_2 g(p_r, p_1, \omega, \tau_2) \qquad (5.11) \\ f(p_1, p_2, \omega, \tau_1) &:= e^{-j\omega\tau_1} b_1(p_2, p_1, \tau_1) \qquad (5.12) \\ g(p_r, p_1, \omega, \tau_2) &:= e^{j\omega\tau_2} b_1(p_r, p_1, \tau_2) \qquad (5.13) \\ h(p_2, p_s, \omega, \tau_2) &:= e^{j\omega\tau_2} b_1(p_2, p_s, \tau_2) \qquad (5.14) \end{aligned}$$

From these definitions, the left handed Riemann expansion of R yields:

$$\begin{aligned} R(p_1, p_2, p_r, p_s, \omega) \approx \Delta\tau^3 \Bigg(& f_0[h_{0+\epsilon} + h_{1+\epsilon} + ...+][g_{0+\epsilon} + ... + g_{1+\epsilon} + ...+] \\ & +f_1[h_{1+\epsilon} + h_{2+\epsilon} + ...+][g_{1+\epsilon} + ... + g_{2+\epsilon} + ...+] + ... + \Bigg) \qquad (5.15) \end{aligned}$$

The Riemann expansion can be re-written as:

$$R(p_1, p_2, p_r, p_s, \omega) = \sum_{j=0}^{N-1} f_j H_j G_j \qquad (5.16)$$

Note that two of the involved terms, H and G, in equation (5.15) have recursive forumations:

$$\begin{aligned} H_0 &= h_{0+\epsilon} + h_{1+\epsilon} + ...+ \\ H_1 &= h_{1+\epsilon} + h_{2+\epsilon} + ...+ = H_0 - h_{0+\epsilon} \\ H_2 &= H_1 - h_{1+\epsilon} \\ &\vdots \end{aligned}$$

[3] Note that the definitions of g and f here are not related to those of the previous subsection.

$$
\begin{aligned}
G_0 &= g_{0+\epsilon} + g_{1+\epsilon} + ...+ \\
G_1 &= g_{1+\epsilon} + g_{2+\epsilon} + ...+ = G_0 - g_{0+\epsilon} \\
G_2 &= G_1 - g_{1+\epsilon} \\
&\vdots
\end{aligned}
\tag{5.17}
$$

The calculation of each G_i and H_i is $\mathcal{O}(N)$ for $i = 0$, but $\mathcal{O}(1)$ for each subsequent term. The total complexity of the construction of R in equation 5.16 is of order $\mathcal{O}(N)$ when considering the recursive formulations for G_i, H_i.

It is possible to do slightly better in terms of the actual implementation, by reversing the order of evaluation involved in equation (5.15):

$$
\begin{aligned}
R(p_1, p_2, p_r, p_s, \omega) \approx \Delta\tau^3 \Bigg(& f_{N-1-\epsilon}[h_{N-1}][g_{N-1}] \\
& + f_{N-2-\epsilon}[h_{N-1} + h_{N-2}][g_{N-1} + ... + g_{N-2}] + \cdots + \Bigg)
\end{aligned}
\tag{5.18}
$$

$$
R(p_1, p_2, p_r, p_s, \omega) = \sum_{j=(N-1)}^{0} f_j H_j G_j \tag{5.19}
$$

Where the suitable recursive formulations read:

$$
H_{N-j} = H_{N-(j-1)} + h_{N-j+\epsilon}, \quad \text{for } 1 \leq j \leq N \tag{5.20}
$$

$$
G_{N-j} = G_{N-(j-1)} + g_{N-j+\epsilon}, \quad \text{for } 1 \leq j \leq N \tag{5.21}
$$

This formulation avoids the need to re-calculate previously calculated terms. It yields a truly optimal $\mathcal{O}(N)$ formulation for calculating $R(p_1, p_2, p_r, p_s, \omega)$. By decoupling of the innermost integrals and recognizing the recursion involved in the numerical integration, the asymptotic cost function has become $\hat{C} \propto \mathcal{O}(N^6)$. This constitutes two orders of N in savings.

5.3 Algorithmic simplifications for the 1.5D predictor

In a 1D medium the representation of b_3 in terms of the defined quantity R simplifies to:

$$
b_3(p_r, \omega) = R(p_r, \omega) \tag{5.22}
$$

The expression for R itself has simplified in terms of its constituent terms:

$$
R^{1.5D}(p_r, \omega) := \int_{-\infty}^{\infty} f^{1D}(p_r, \omega, \tau_1) d\tau_1 \times \int_{\tau_1+\epsilon}^{\infty} d\tau_2 h^{1D}(p_r, \omega, \tau_2) \times \int_{\tau_1+\epsilon}^{\infty} d\tau_2 g^{1D}(p_r, \omega, \tau_2) \tag{5.23}
$$

Where the integral kernels are represented by:

$$f^{1.5D}(p_r, \omega, \tau_1) = b_1(p_r, \tau_1)e^{-j\omega\tau_1} \tag{5.24}$$

$$h^{1.5D}(p_r, \omega, \tau_2) = g^{1.5D}(p_r, \omega, \tau_2) = b_1(p_r, \tau_2)e^{j\omega\tau_2} \tag{5.25}$$

Due to the equality between the two innermost integration kernels, we can re-write for R:

$$R^{1.5D}(p_r, \omega) := \int_{-\infty}^{\infty} f^{1.5D}(p_r, \omega, \tau_1)d\tau_1 \times \left[\int_{\tau_1+\epsilon}^{\infty} d\tau_2 g^{1.5D}(p_r, \omega, \tau_2) \right]^2 \tag{5.26}$$

The recursive formulation for G still holds, yet now G and H are equal. The optimal asymptotic cost function for the 1.5D predictor is $\hat{C}^{1.5D} \propto \mathcal{O}(N^3)$.

5.4 Algorithmic optimization: A priori dip angle knowledge

Liu et al. (2000) recognized that for a wavefield of plane waves the possible coupling between slownesses on source and receiver sides is limited by the maximum dip and the minimum velocity of the medium. The expression reads:

$$|p_s - p_r| \leq \frac{2\sin(\alpha)}{v_{min}} \tag{5.27}$$

Equation (5.27) mirrors the slowness bandwidth of the input dataset b_1, and, by the interpretation of Ma et al. (2009), also the bandwidth of the output dataset b_3.

We repeat that the computational complexity of the 2D internal multiple predictor in terms of slownesses is proportional to $(N_p)^4$. Searching only for contributions within the range defined by equation (5.27) can yield a highly significant speedup in cases of limited dip of interfaces related to internal multiple generation. In fact, for a laterally invariant earth, $\alpha = 0$, the 2D internal multiple predictor deteriorates to the 1.5D predictor, except that the former computes the prediction the $\underline{\text{coupled}}$ plane wave domain.

In order to demonstrate the possible computational savings involved, we will demonstrate a simple case where the medium and sampling parameters are given by:

v_{min}	1500 m/s
p_{min}	-0.5 s/km
p_{max}	0.5 s/km
N_p	251
α	$\{5°, 10°, 20°\}$
$\lvert p_s - p_r \rvert$	$\{0.1162, 0.2315, 0.456\}$

Table 5.1: Parameters used for demonstrating the effect of dip restriction upon number of calculations required.

p_{min} and p_{max} define the range of slownesses in which the internal multiple prediction is calculated. For this range, the number of slownesses N_p are determined using an incremental slowness Δp at the aliasing limit set by a maximum frequency of $f_{max} = 50\ Hz$ and a horizontal range of the domain $x_{s,max} - x_{s,min} = x_{r,max} - x_{r,min} = 5000\ m$. The example attempts to represent a realistic scenario.

We define the number of calculations as the number of computations of the quantity $R(p_1, p_2, p_r, p_s, \omega)$ normalized by the number of frequencies.

	Calculations		
Dip (α)	No dip restriction	Dip restricted by equation (5.27)	Relative saving
5°	3.9691e+09	2.9210e+07	135.89
10°	3.9691e+09	1.9765e+08	20.08
20°	3.9691e+09	1.1630e+09	3.4130

Table 5.2: Calculations of $R(p_1, p_2, p_r, p_s, \omega)$ needed for certain maximum dips, given the parameters in table 5.1. Calculations are normalized by the number of frequencies.

The results in table 5.2 indicate that it is possible to realize large savings in the presence of only a slightly dipping subsurface. In the limit of $\alpha \rightarrow 45°$ the relative computational saving approaches 1, i.e. no saving.

5.5 Algorithmic optimization: Symmetry of the frequency spectrum

The internal multiple prediction $d_3(p_r, p_s, \tau)$ is calculated via an inverse Fourier transform of its Fourier-domain representation:

$$d_3(p_r, p_s, \tau) = \mathcal{F}^{-1}\big\{d_3(p_r, p_s, \omega)\big\} \tag{5.28}$$

In the discrete sense, the inverse Fourier transform of $F[\omega_n],\ n \in 0, \dots, N-1$ yields the time series $f[\tau_k],\ \tau_k = k\Delta\tau,\ k \in 0, \dots, N-1$. When $f[\tau_k]$ is a real-valued time-series, $F[\omega_n]$ is completely determined from the first $N/2+1$ frequencies:

$$F[\omega_{N-n}] = F^*[\omega_n] \tag{5.29}$$

I.e. the amplitude spectrum of the Fourier representation of the time series is symmetric around the Nyquist frequency.

In the context of internal multiples, we know a-priori that the internal multiple model must be real-valued, also in the coupled plane wave domain. Therefore, we need only to calculate approximately half of the frequencies involved in constructing $d_3(p_r, p_s, \tau)$. Information at frequencies above the Nyquist frequency can either be calculated by equation (5.29) or be implicitly known by utilizing a complex-to-real discrete inverse Fourier transform.

5.6 Algorithm for calculating asymptotically optimal 2D internal multiple predictor

In order to elaborate a bit more to the reader on how the internal multiple predictor is implemented, a pseudo-code listing is found in algorithm 3. It contains the algorithimic simplifications introduced, mostly through the quantity R, defined in equation (5.5). The only quantity not explicitly introduced is the table E containing the tabulation of the complex exponential terms, which is introduced in section 5.7.2. For compactness of the algorithm listing we have assumed $z_r = z_s$.

Algorithm 3 2D Internal multiple predictor with algorithmic optimizations

```
Calculates the internal multiple prediction $d_3^{IM}(p_r, p_s, \tau)$
Input: $d(p_r, p_s, \tau)$, $\epsilon$, $|p_r - p_s|$, $c_0$
Output: $d_3^{IM}(p_r, p_s, \tau)$
/* Initialization                                                    */
$\beta \leftarrow |p_r - p_s|$ /* Bandwidth in terms of slownesses.    */
for $p_s = p_{min}$ to $p_s = p_{max}$ do
    for $p_r = p_{min}$ to $p_r = p_{max}$ do
        $b_1(p_r, p_s, \tau) \leftarrow -2jq_s d(p_r, p_s, \tau)$ /* Create effective data   */
    end
end
/* Compute loop                                                      */
for $\omega = 0.0$ to $\omega = \omega_{Nyq}$ do
    $E \leftarrow$ tabulateExponentials$(\omega, d\tau, \tau_0, N_\tau)$
    for $p_s = p_{min}$ to $p_s = p_{max}$ do
        forall contributing $p_r$ do
            accumulator $\leftarrow 0.0 + 0.0i$
            forall contributing $p_1$ do
                forall contributing $p_2$ do
                    accumulator += calculate_R$(b_1, p_s, p_r, p_1, p_2, E)$

                end
            end
            $d_3(p_r, p_s, \omega) \leftarrow (-2jq_s)^{-1} \times$ accumulator
        end
    end
end
/* Inverse Fourier transform, complex to real                        */
$d_3(p_r, p_s, \tau) \leftarrow \mathcal{F}^{-1}\Big\{d_3(p_r, p_s, \omega)\Big\}$
return $d_3^{IM}(p_r, p_s, \tau)$
```

5.7 High-performance code development

5.7.1 Operational Intensity

We define operational intensity, I_o, as the number of floating point operations per byte transferred from DRAM (Williams et al., 2009).

$$I_O(N) = \frac{C(N)}{B(N)} \tag{5.30}$$

$C(N)$ is the floating point operations complexity function for a given size N, and $B(N)$ is the number of bytes transferred from dynamic RAM. Operational intensity gives an indicator for whether the performance of a computational kernel (i.e. an algorithm) is most likely to be limited by the instructions throughput of the hardware, *compute bound*, or by the transfer bandwidth between caches and DRAM, *memory bound*. The two extrema correspond to high and low computational intensity, respectively. An exact cutoff for what defines high or low operational intensity is not particularly meaningful. However, in terms of asymptotics, a kernel with an asymptotic bound greater than $\mathcal{O}(1)$ is often not limited by memory bandwidth and is therefore compute bound. For example, general Matrix-Matrix multiplication is a kernel with $I_O(N) \propto \mathcal{O}(N)$ and well-designed implementations are able to reach full hardware performance for large input sizes N.

In the context of the internal multiple predictor we will attempt to state an approximate operational intensity in order to highlight which optimizations are most important in order to improve program performance. This forms the theory for the optimizations implemented into the code.

The floating point cost function of the internal multiple predictor after algorithmic optimizations is asymptotically $C(N) \propto \mathcal{O}(N^6)$. For the amounts of bytes transferred from DRAM we will only consider compulsory loads and implicitly ignore capacity and conflict misses on the level of caches. The measure of amounts of bytes transferred will therefore not be strictly correct, yet for the demonstrational purpose considered it suffices. Empirical insight into the exact number of memory transfers can be given from performance monitors, which can be called from inside an implemented program. An example of such a monitor is the Intel Performance Counter Monitor (PCM) (Intel, 2017 (accessed February 2018). As this was not available for installation in the development system the author used, the simple asymptotic bounds have to suffice. The amount of compulsory loads on b_1 is bounded by $B(N) \propto \mathcal{O}(N^3)$. The asymptotic bound on the operational intensity is therefore:

$$I_O(N) \propto \frac{\mathcal{O}(N^6)}{\mathcal{O}(N^3)} = \mathcal{O}(N^3) \tag{5.31}$$

This demonstrates that the inverse scattering series internal multiple predictor can be considered a compute-bound algorithm. The majority of code optimizations implemented and described in the subsequent sections are therefore optimizations that aim to improve the instruction throughput and/or reducing the number of floating point operations performed.

5.7.2 Computational strategies: Tabulation

Tabulation is the process of pre-calculating and/or storing previously calculated values in a table. Certain computational algorithms, when implemented in a straight-forward fashion, may involve large amounts of redundant computations. In such cases, values that have already been calculated are re-calculated on a further iteration. By simply storing the values in question in an array, one can prevent an excessive waste of clock cycles. Pre-calculation may also be beneficial in cases where instruction switches yield significant amounts of latency, e.g. frequent switching from multiplication to addition. Tabulation may then yield a performance closer to the throughput bound, instead of the latency bound. The success of tabulation routines will depend on the size of the resulting table. If it is small enough to be 'guaranteed' to be in caches, tabulation most certainly can be considered. If its values always need to be loaded from DRAM, tabulation may even cause lower program performance.

Tabulation of exponentials

Tabulation is especially beneficial if the values considered are generated by an expensive instruction. All instructions that are not directly pipelined, such as trigonometric expressions, logarithms etc., and depend on series expansion need several clock cycles to complete. For simple, pipelined instructions such as floating point multiplication and addition the instruction throughput may be more than one instruction per clock cycle[4]. Evaluation of sine and cosine expressions typically need to wait at least 20 cycles per issue, making this an extremely expensive operation[5].

Herein lies one of the most important potentials for optimizing implementations of the ISS internal multiple predictors. The factors $e^{-j\omega\tau_1}$ and $e^{j\omega\tau_2}$ are per-frequency constant. In an ordinary implementation these would be calculated via Euler's identity $e^{jax} = \cos(ax) + j\sin(ax)$. Instead, by implementing the loop over ω as the outermost one, and at this level tabulating the exponential factors one can gain serious speedups. The tables are also small, needing only $8N_\tau$ bytes each, and will be kept in cache as they are always re-used at the innermost computational level.

Furthermore, we recognize that for a regularly sampled τ:

$$e^{\pm j\omega\tau_k} = e^{\pm j\omega\tau_0} \prod^{k} e^{\pm j\omega d\tau} \tag{5.32}$$

Expression (5.32) can be evaluated using only two complex exponential evaluations and $k-1$ complex-complex multiplications. The total amount of complex exponential evaluations can therefore be reduced from $\mathcal{O}(N^6)$ to $\mathcal{O}(N)$, and the time to calculate the tables therefore becomes negligible.

[4] Depending on the number of floating-point unit (FPU) ports on the given hardware.

[5] It is however not possible give an exact latency for trigonometric functions. Inherently, this is because the number of clock cycles depend not only on the hardware but also on the implementation, and even the input value. Indeed, for small values x the best approximation to $\sin(x)$ is simply $\sin(x) \approx x$, involving almost no clock cycles used.

Tabulation of the G term
The sequence $G_0, \ldots, G_{N-1}$, see e.g. equations (5.17) and (5.13), depend on the slownesses p_r and p_1. By demanding that the loop over p_2 is the innermost, the terms G_i constitute a constant sequence there. By simply storing the values of G calculated on each new iteration over p_1, one can avoid some redundant computations. This table is also small and is expected to reside in cache.

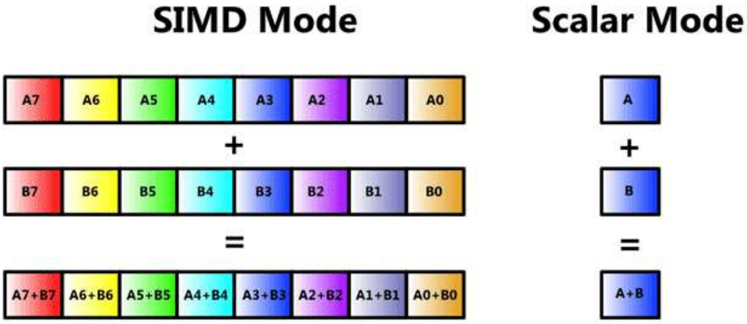
5.7.3 Vectorization: The principle of SIMD

If utilized, the vectorization features implemented in modern central processing units (CPUs) can bring highly significant improvements to programs performance. Perhaps the best way to introduce the concept is to quote Intel, the currently dominant producer of CPUs.

> *Computing architecture can be described at the highest level using Flynn's architecture classification scheme as single instruction, single data (SISD); multiple instruction, single data (MISD); single instruction, multiple data (SIMD); or multiple instruction, multiple data (MIMD) systems. General-purpose computing most often employs simple SISD processing units, even if they are multi-core, super-scalar, or hyper-threaded; but the addition of vector instructions that can operate on multiple data words in parallel with a single instruction (SIMD) can provide powerful acceleration for data-intensive algorithms* (Intel (2016 (accessed February 2018))

In the SIMD model there are simultaneous (parallel) computations that arise due to only a single instruction. This is often employed by making use of so-called vector registers which, in distinction from scalar registers, can hold multiple entries of the given datatype. The operations are then employed in parallel, independently for each entry in the vector register. The parallelism employed is at one of the finest grain scales possible, e.g. it is much more fine-grained than multi-core or multi-thread parallelism. The difference between a SIMD instruction for addition of numbers in arrays **A** and **B** is graphically shown in figure 5.1.

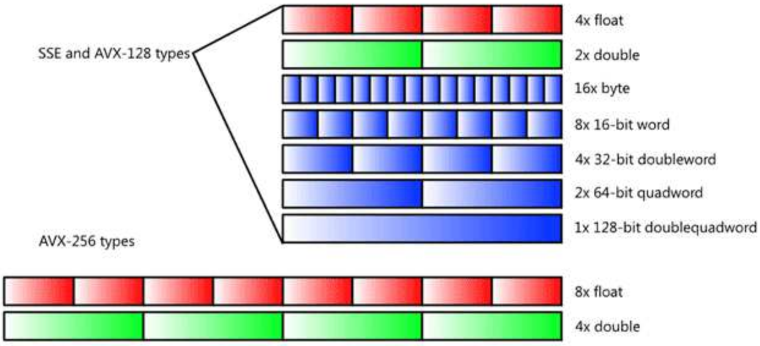
Figure 5.1 Comparison between a SIMD and a scalar instruction for vector addition. The width of the SIMD register can hold 8 values of the given datatype. Retrieved from Intel (2011 (accessed February 2018).



In this particular situation, in the SIMD case eight additions can be independently performed per instruction, while in the scalar case only one addition can be performed per instruction. In both cases the number of instructions performed is directly determined by the width of the registers. On modern hardware, scalar registers are physically and logically only subparts of vector registers[6]. Supporting legacy instructions this way ensures that both the vector and scalar instruction have the same latency and throughput. In essence, this means that for an n-value wide SIMD register, the maximum theoretical speedup the SIMD instruction can give relative to the scalar instruction is n times.

The instruction set used for vectorization is the Advanced Vector Extensions with 256 bit registers (AVX-256), originally introduced in 2011 (Lomont, 2011 (accessed February 2018). The available vector datatypes are shown in figure 5.2.

Figure 5.2 Datatypes of the AVX-256 extension to the x86-64 ISA. Retrieved from Intel (2011 (accessed February 2018)



[6] On x86 this started with the SIMD extension SSE2, introduced with the Pentium 4 in year 2000. Floating point instructions in x86-64 are based on the vector extensions SSE or AVX. All processors capable of executing x86-64 code must support at least SSE2 (Bryant et al., 2003).

The single-precision `float` constitues the standard datatype in virtually all seismic data processing. The AVX-256 type `__m256` can hold eight single-precision floating point values, corresponding to eight real values or four complex values. In figure 5.2, this corresponds to the second lowermost datatype.

The following notes outline some of the subroutines in the ISS internal multiple predictors most relevant for vectorization.

Vectorized complex-complex math multiplication.

Multiplication of two complex numbers $z = a + jb$ and $w = c + jd$ is defined mathematically as:

$$zw = (ac - bd) + j(bc + ad) \tag{5.33}$$

The operation therefore requires four multiplications and two additions. Moreover, several terms interact across the boundaries of register entries. In order to enable that interaction, shuffle instructions are needed in order to create the layout required. A minimum-work implementation uses three shuffles, one multiplication and one fused multiplication and addition (FMA) operation.

INSTRUCTION	AMOUNT	LATENCY	RECIPROCAL THROUGHPUT
_mm256_shuffle_ps	3	1	1
_mm256_mul_ps	1	3	0.5
_mm256_fmaddsub_ps	1	3	0.5

Table 5.3: Instruction table for complex-complex multiplication

By table 5.3 the lower bound on cycles needed for multiplication of four complex numbers by four complex numbers is given by 9 cycles. If FMA is not supported, one would have to add three cycles, as the `_mm256_fmaddsub_ps` operation would be replaced by one `_mm256_mul_ps` and one `_mm256_addsub_ps`. A lower bound for scalar calculation for a single complex multiplication would be $9.5 \rightarrow 10$ or $10.5 \rightarrow 11$ cycles, with and without FMA operations, respectively. In terms of four complex numbers this would respectively yield 40 and 44 cycles.

OPERATION:COMPLEX-COMPLEX MULTIPLICATION

INSTRUCTION SET	RECIPROCAL THROUGHPUT PER 4 COMPLEX
AVX-256 & FMA	9
AVX-256	12
x86-64 & FMA	40
x86-64	44

Table 5.4: Lower bounds for complex-complex multiplication

From table 5.4 there are potentially good gains to be had from implementing vectorized code for this operation.

Vectorized imaginary-complex math multiplication.

The motivation for imaginary-complex math multiplication stems from the fact that the input data $b_1(p_r, p_s, \tau) = (-2jq_s)d(p_r, p_s, \tau)$ is purely imaginary, and can therefore be stored as a normal `float`. This can enable a reduced amount of computations as well as freeing up memory bandwidth.

The multiplication of the imaginary number $z = jb$ and the complex number $w = c + jd$ is defined mathematically as:

$$zw = -bd + jbc \tag{5.34}$$

This operation requires two multiplications and a sign-flip. However, some extra care must be taken when implementing this kernel. An __m256 register can theoretically hold eight imaginary numbers, yet it can only hold four complex numbers. For performing loads on z one should only load four values. This is implemented by using SSE-128 loads on z and AVX-256 loads on w.

INSTRUCTION	AMOUNT	LATENCY	RECIPROCAL THROUGHPUT
_mm256_castps128_ps256	1	0⋆	0⋆
_mm256_permute2f128_ps256	1	3	1
_mm256_permutevar_ps	1	1	1
_mm256_permute_ps	1	1	1
_mm256_mul_ps	1	3	0.5
_mm256_xor_ps	1	1	1
_mm256_blend_ps	1	1	1/3

*Does not generate an instruction. Only for compilation.

Table 5.5: Instruction table for imagininary-complex multiplication

The lower bound for four imaginary-complex multiplications is 10 cycles. The lower bound of a scalar version for one imaginary-complex multiplication would be $4.5 \rightarrow 5$ cycles, as it requires only two multiplications and a bitwise XOR operation[7]. However, the scalar operation would need an intrinsic of sort, as bitwise XOR is not defined for floating point numbers in x86-64. Therefore, the lower bound for the scalar version might be even higher.

OPERATION:IMAGINARY-COMPLEX MULTIPLICATION

INSTRUCTION SET	RECIPROCAL THROUGHPUT PER 4 COMPLEX
AVX-256	10
x86-64	20

Table 5.6: Lower bounds for imaginary-complex multiplication

We note however, that in the case that the FMA instruction set is available, it could be slightly cheaper to perform this operation via complex-complex multiplication with the real part of one of the inputs set to zero.

5.7.4 Parallelization

Parallelization is unavoidable for the more demanding algorithms and/or bigger datasets. This is especially true for the 2D ISS internal multiple predictor, while the 1.5D predictor can avoid the need for parallelization, depending on the size of the input.

Modern CPUs typically feature several independent workers, or cores, typically in the range 2 – 32 per socket. Several processors are often linked together across nodes in order to form computing clusters. One can utilize the sheer number of processors and program them to perform independent calculations. In this way, more work is performed in the same amount of time.

I will briefly discuss the parallelization implemented in the 2D predictor. It is implemented using the distributed memory library standard named Message Passing Interface (MPI) (Walker and Dongarra, 1996). Domain decomposition is used by splitting the dimensions of the output along the slowest varying dimension, in this case the source slowness p_s. The input, b_1, is distributed to all processes through collective communicators. Idem, the output is collected from all processes through a (variable) *gather* collective communicator. This scheme minimizes the amount of communication required and works well in 2D. This particular parallelization model scales up to a few hundred processes. A more sophisticated approach would make use of shared-memory parallelization within each node using e.g. OpenMP or POSIX Threads. For research grade code the implemented parallelization scheme is more than sufficient.

[7] This would be the cheapest way to perform a sign change of floating point number.

5.8 Implemented internal multiple predictors

The internal multiple predictors implemented in this thesis are:

- The 1.5D internal multiple predictor in the plane wave domain, $\tau - p$, equation (2.56).
- The 2D internal multiple predictor in the coupled plane wave domain, $\tau - p_r - p_s$, equation (2.54).

The algorithmic optimizations present in the code are:

- Optimal asymptotic cost functions, $\mathcal{O}(N^3)$ and $\mathcal{O}(N^6)$ for the 1.5D and 2D predictors, respectively.
- Only contributing slownesses are calculated for a given dip range.
- Utilization of symmetry of frequency spectrum.

The code-specific optimizations include:

- General, scalar optimizations such as function inlining, scalar replacement etc.
- Tabulation of the innermost exponential terms in the computational kernel.
- AVX-256 vectorization of imaginary-complex and complex-complex math routines.
- MPI-based parallelization based on domain decomposition and collective communicators.

The author of this thesis has written all of the predictor codes in the **C++** programming language. The Madagascar API (Madagascar Development Team, 2012) has been utilized for input and output routines whereas Fast Fourier Transforms are handled by the FFTW3 library (Frigo and Johnson, 2005).

5.9 Benchmarks

For benchmarking the effect of some of the proposed implementational strategies we will make use of the 1.5D internal multiple predictor. It can contain virtually all the optimizations possible to use for the full 2D predictor, and still provide runtimes acceptable for benchmarking even the slowest implementations. Especially for the baseline code, implementing this into the 2D code would lead to unacceptable runtimes. Furthermore, for the 1.5D predictor no parallellization is needed, allowing the study of purely serial optimizations. The results from this section will demonstrate that serial optimizations might be just as important for program performance as parallelization yet with the advantage that no extra resources (hardware) is required. The parallelization considered for the 2D predictor scales linearly and is not particularly interesting for benchmarking.

For the purpose of benchmarking we will consider three implementations. All of which include all of the possible algorithmic simplifications described in sections 5.2 through 5.5. Except for the differences denoted in the listing below, all implementations have the same general optimizations present, e.g. function inlining, scalar replacement etc[8].

- **Baseline**: An implementation in which all exponentials are explicitly evaluated. No SIMD is present
- **Tabulation**: An implementation in which all exponentials are tabulated according to the procedure in section 5.7.2. No SIMD is present.
- **Tabulation+AVX-256**: An implementation in which all exponentials are tabulated according to the procedure in section 5.7.2. AVX-256 based SIMD routines for all of the innermost computational routines are present.

INFO ON BENCHMARKING SYSTEM

CPU	Intel(R) Xeon(R) CPU E5-2670
Operating System	RedHatEnterpriseServer 6.6
Compiler	G++ version 4.4.7
Compiler Flags	-O3 -fno-tree-vectorize -std=c++0x (-mavx)[a]
Timing infrastructure	std::chrono
Timing sensitivity	1E+06 TICKS/SECOND
General math library	<math.h>
Complex math library[b]	std::complex

[a] Only used for vectorized code

[b] Only used in scalar implementations.

Table 5.7: Benchmarking system and software.

Basic information on the input dataset is as follows:

Benchmarking dataset

Radon gathers	3
N_p	1089
N_t	949
ϵ	$0.25s$

Table 5.8: Properties of dataset used for benchmarking

Using the dataset properties in table 5.8 and the benchmarking system properties in table 5.7, the benchmarking of the performance of the different implementations is performed.

[8] An excellent reference on scalar optimizations can be found in e.g. Bryant et al. (2003).

The measurements are all cold-cache but do not show any significant variability across runs. The results were averaged over two independent runs and are shown in table 5.9. The graphical bar plots of the information contained in the table are shown in figures 5.3 and 5.4.

Implementing tabulation of complex exponential functions lead to a speedup of a factor `18.4`. This demonstrates that indeed trigonometric functions are expensive to compute. In this particular case, highly significant speedups are available from performing tabulation.

The gain that the implementation of AVX-256 routines for complex and imaginary mathematics brought over the scalar, tabulated implementation reads a factor of `16.9`. Writing such vectorized code is naturally more challenging from the view of the implementer, however it often pays off; the speedup presented is very good. The speedup in fact exceeds a-priori expectations of maximum speedup, namely a factor `8x`. SIMD code requires the developer to explicitly handle memory references including loads and stores to/from registers. Well written vectorized code will therefore typically enable the compiler to perform memory optimizations beyond what it is allowed to perform in the scalar case. It is also probable that the complex math functions in `std::complex` are not optimal in terms of (scalar) performance.

The total speedup from worst to best is a factor of more than 300 times the program performance of the former. From these benchmarks we conclude that attempts to understand and exploit the underlying hardware limitations and capabilities are very important in order to realize implementations of demanding geophysical algorithms, with practical value.

Benchmark results

IMPLEMENTATION	RUNTIME [s]	SPEEDUP VS BASELINE
BASELINE	1277.86	1.0
TABULATE	69.5104	18.4
TABULATE+AVX-256	4.1116	310.8

Table 5.9: Benchmark results

Figure 5.3 Benchmark of three implementations of the 1.5D predictor

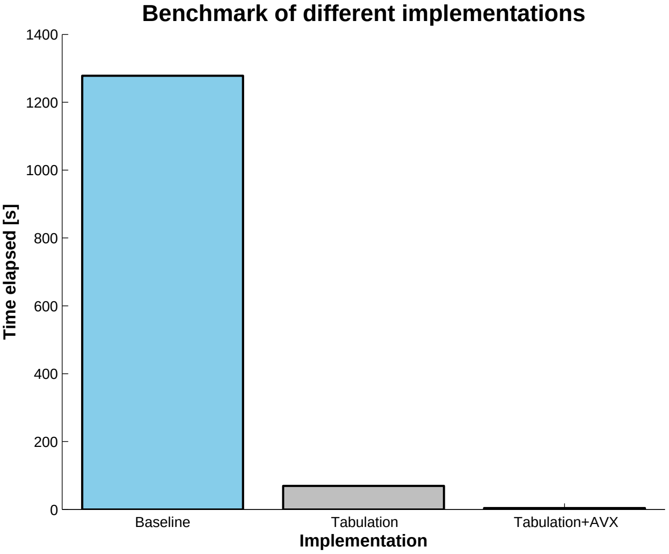
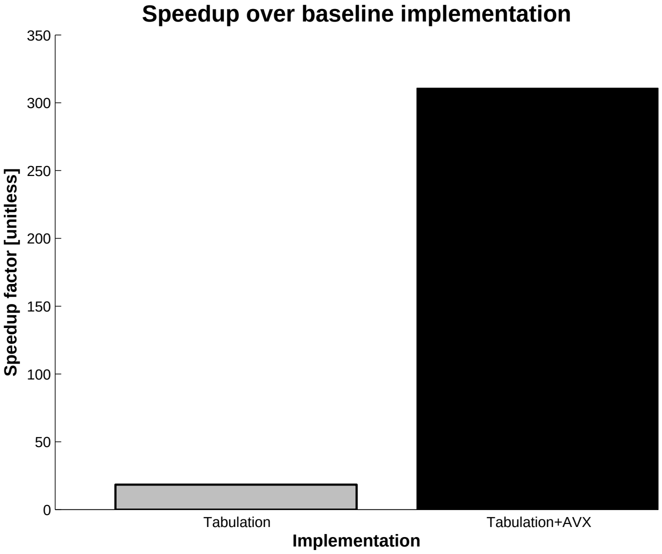


Figure 5.4 Speedup of two implementations of the 1.5D predictor with respect to the baseline implementation.



Chapter 6

Predictions of internal multiples

6.1 2D predictor using 1.5D data

The usage of one dimensional earth models, can often be very helpful. Although they are simple, they can be used to test code, algorithms and routines as analytical solutions are often available or intuitively deduced. For this very purpose, the aim of this subsection is to test the 2D internal multiple predictor in the coupled plane wave domain using a 1.5D dataset. The one dimensional earth model is shown in figure 6.1 and contains only two interfaces.

The forward modelled seismic data are calculated with Jan Thorbecke's & Deyan Draganov's open source Finite Difference code (Thorbecke and Draganov, 2011). Perfectly Matching Layers (Berenger, 1996) were used for all four boundaries in order to suppress simulation artifacts as well as to not model free surface multiples. Only one shot gather is modelled and a simple utility was built for creating the remaining shots by exploiting the lateral invariance of the earth model. A cube of the resulting shots with the geometry of shots and receivers occupying all horizontal positions is shown in figure 6.2.

The input data were transformed to the coupled plane wave domain via algorithm 1, using the standard-resolution transform in *sfradon*. Cosine tapering of the edges of source and receiver coordinates have been performed to suppress artifacts. The coupled plane wave domain representation is shown in figure 6.3 for three source slownesses $p_s \in \{-0.2, 0.0, 0.2\}$ s/km, from left to right respectively. For reference, this is the same dataset used for discussing artifacts in (p_r, p_s, τ) in section 3.5 where it is represented in figure 3.2.

The internal multiple prediction is calculated using an ϵ value of $\epsilon = 0.25s$ and a contributing slowness bandwidth corresponding to $|p_r - p_s| = 0.05$ s/km. The resulting internal multiple predictions are compared to the input data for source slownesses $p_s \in \{-0.16, 0, 0.16\}$ s/km in figures 6.4 through 6.6, respectively. The coupled plane wave domain representation of the internal multiples seem to be well recovered. After inverse transforming to the physical domain that description holds very much still. The internal multiple predictions for shot gathers at source positions $x_s \in \{500, 2500, 4000\}\ m$ are shown in figures 6.7 through 6.9. One can observe that the internal multiple predictions are indeed kinematically correct. The temporal support of the predicted internal multiples is larger than that what found in the input. This is a known effect of not performing source-signature deconvolution before prediction (Weglein et al., 2003).

From the discussion on the effect of transform artifacts on internal multiple predictions from section 3.5 these results, using a dataset with a considerable amount of artifacts, renders the question:

Why are the predicted internal multiples relatively free from artifacts when computed from an input dataset with artifacts?

This question fortunately has a straight-forward answer. Because the contributing slowness bandwidth $|p_r - p_s|$ is very much restricted, the data are only barely able to combine with their artifacts. Had the prediction been run with a higher value of $|p_r - p_s|$, then artifacts would contribute more to the prediction. We will see examples of this in section 6.2 when proper 2D data are used.

Figure 6.1 A simple horizontally layered medium consisting of two interfaces.

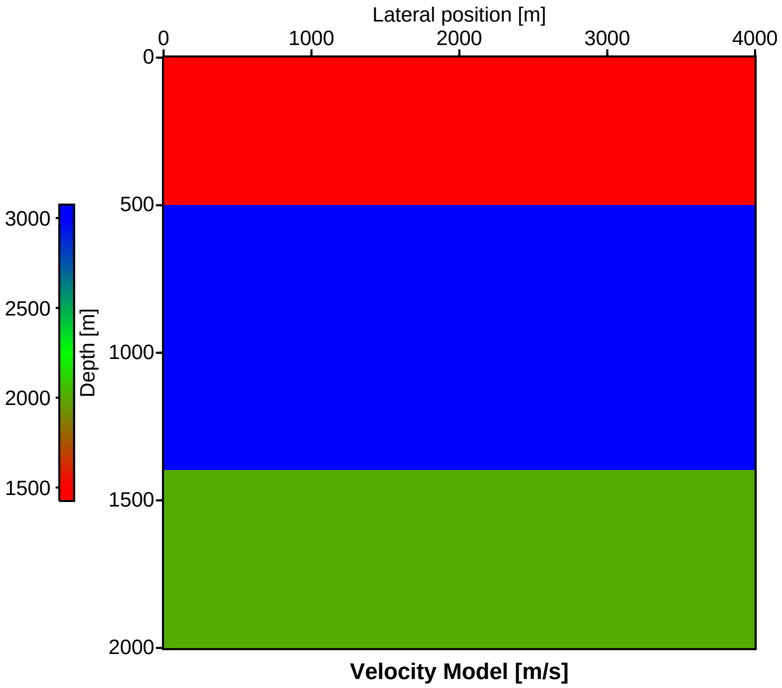


Figure 6.2 Shot gathers modelled via the Finite Difference method

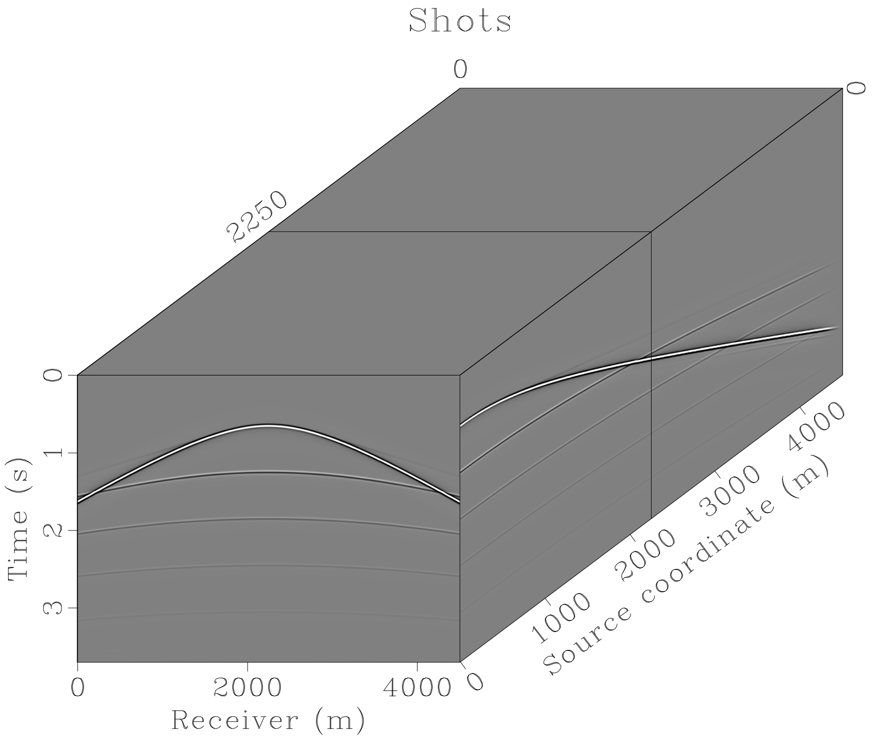


Figure 6.3 1.5D Input data shown for three distinct source slownesses.

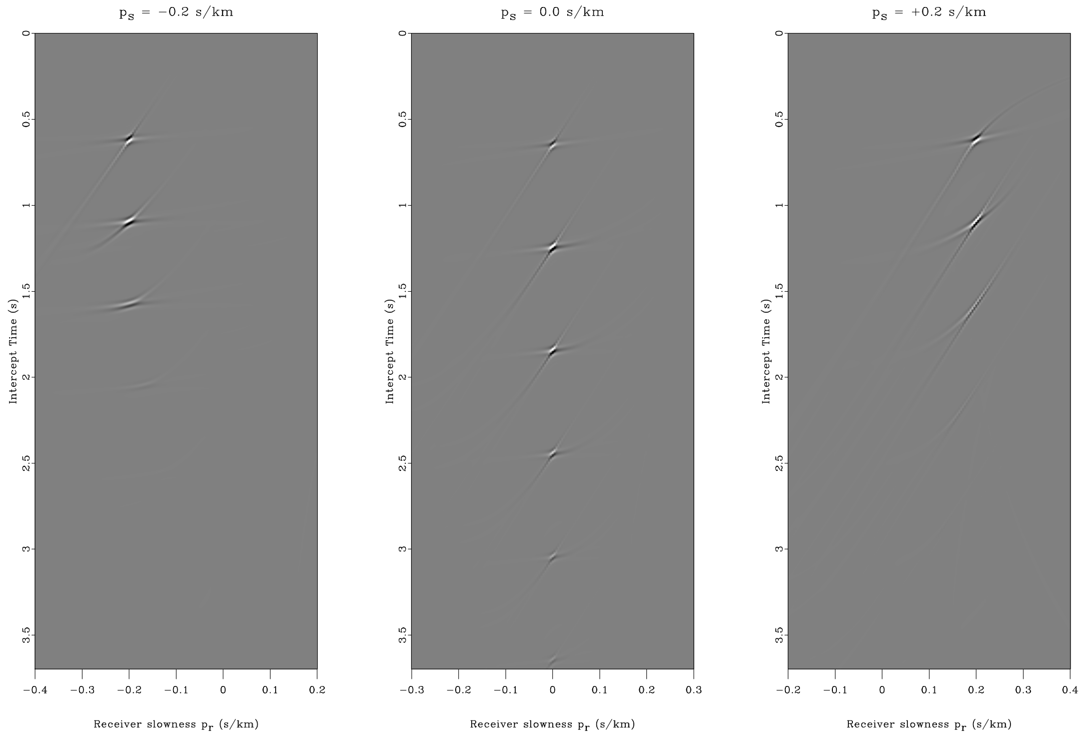


Figure 6.4 Comparison of internal multiple prediction in coupled plane wave domain: $p_s = -0.2$ s/km

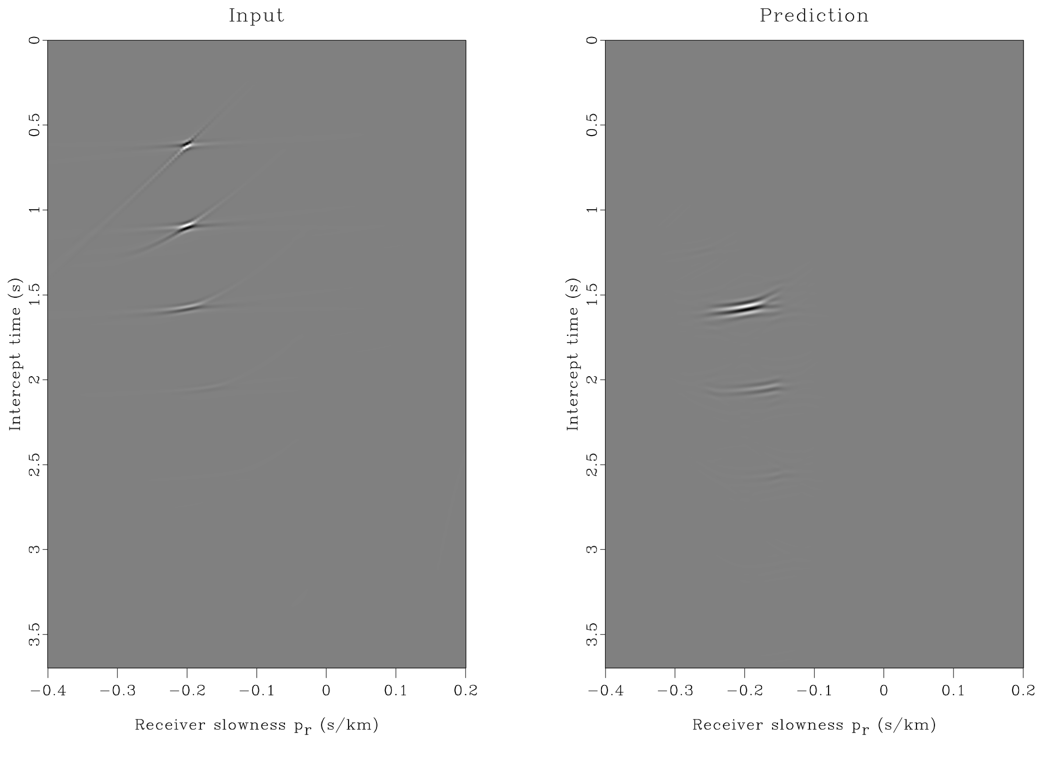


Figure 6.5 Comparison of internal multiple prediction in coupled plane wave domain: $p_s = 0.0$ s/km

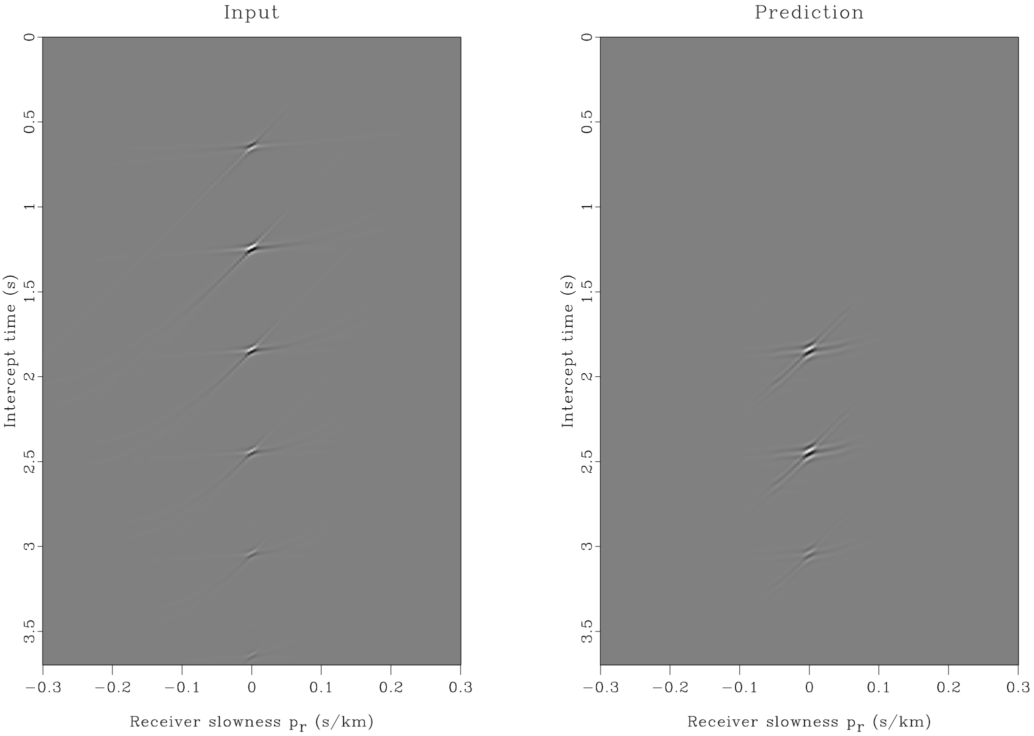


Figure 6.6 Comparison of internal multiple prediction in coupled plane wave domain: $p_s = 0.2$ s/km

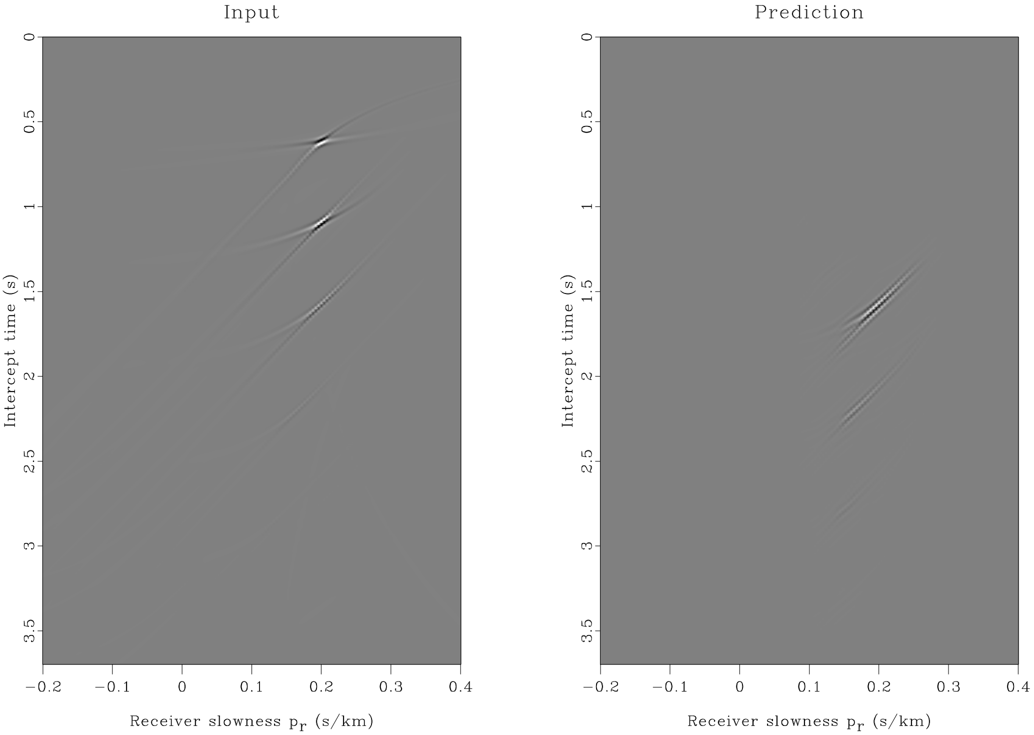


Figure 6.7 Comparison of internal multiple prediction in time-space domain: $x_s = 500$ m

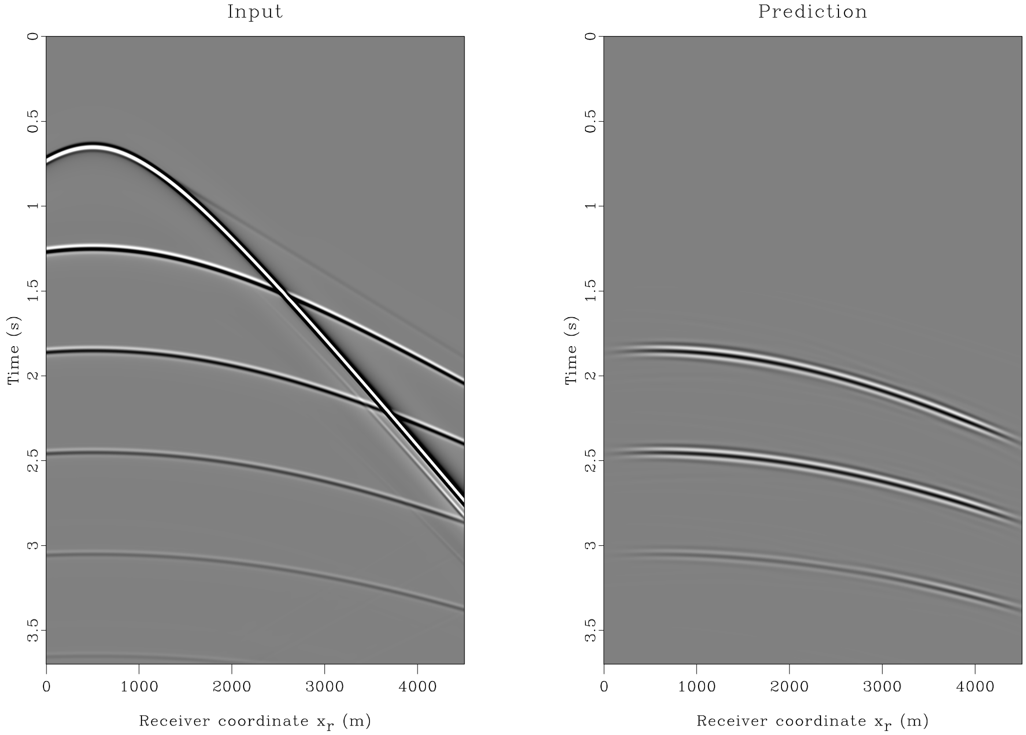


Figure 6.8 Comparison of internal multiple prediction in time-space domain: $x_s = 2250$ m

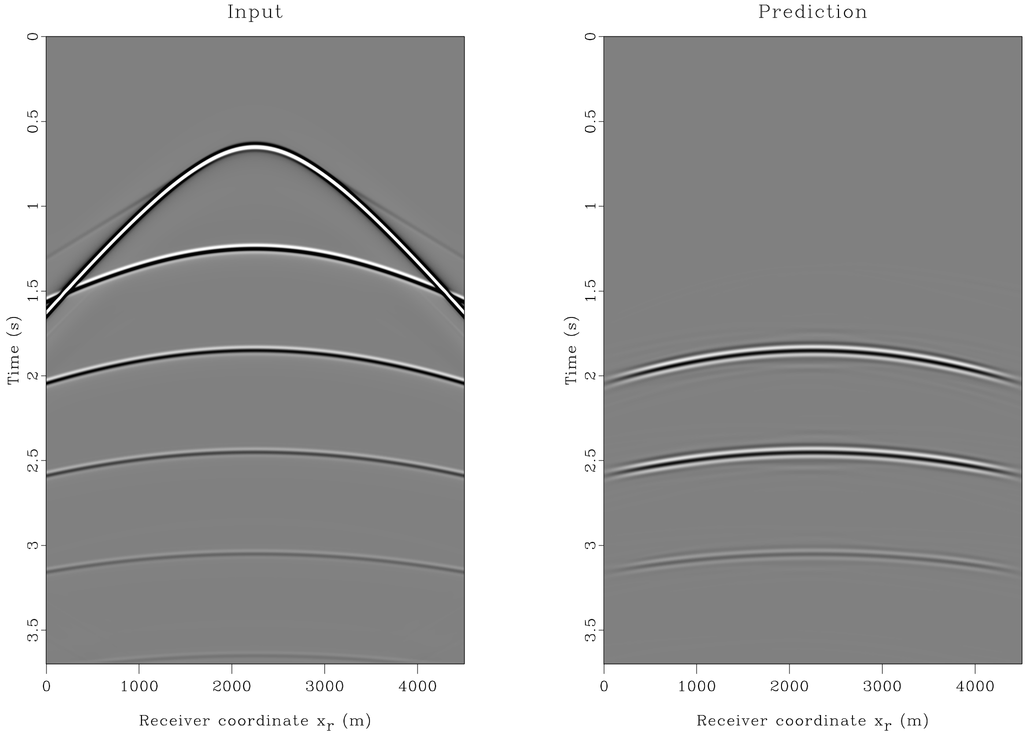
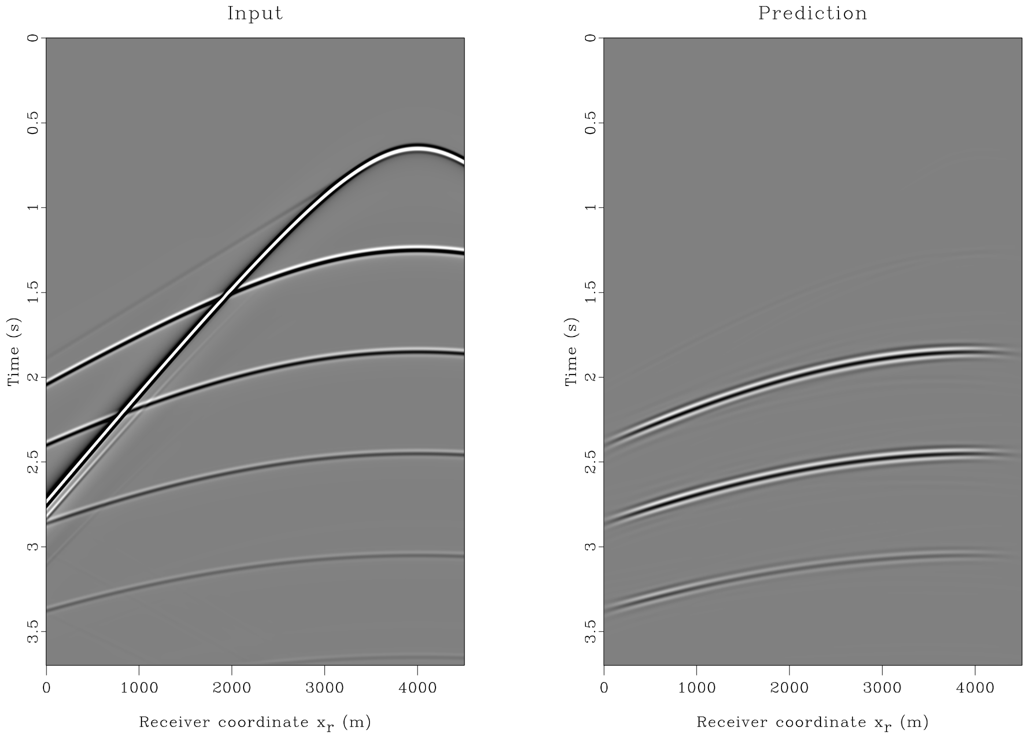


Figure 6.9 Comparison of internal multiple prediction in time-space domain: $x_s = 4000$ m



6.2 2D predictor using 2D data

For properly testing the 2D predictor it is required to use an earth model containing two dimensional features. We will consider the velocity and density models based upon a model of the Johan Sverdrup field. The full models are shown in figure 6.10. From the discussion in chapter 5 it would be useful, for research purposes, to consider an earth model with a restricted dip range, so that internal multiple predictions can be calculated in a relatively short time. For this, we consider the velocity and density models displayed in figure 6.11, a subpart of the full Johan Sverdrup model. The maximum dips in the overburden are somewhat less than 10°. The steeply dipping underburden is not considered.

6.2.1 Using standard resolution Radon Transforms.

The author wishes to thank professor Børge Arntsen for providing the Finite Difference modelling code written by Espen B. Raknes and Wiktor Weibull used to create the synthetic data. A cube of the modelled shots are shown in figure 6.12. As a general characteristic, the Finite Difference method suffers somewhat from the discretization of the interfaces in the velocity model. This is termed the 'staircasing' problem, yielding numerical diffractions in the modelled data. Figure 6.13 shows an example shot gathers plotted with a colorscale in order to emphasize the numerical diffractions present.

For the first experiment, the modelled data were transformed to the coupled plane wave domain via algorithm 1, using the standard-resolution transform in *sfradon*. Cosine tapering of the edges of source and receiver coordinates have been performed to suppress artifacts. Figure 6.14 shows the coupled plane wave representation for the slownesses $p_s \in \{-0.2, 0.0, 0.2\}\ s/km$, respectively. The two convex-up curves are mappings of the reflection events related to the two most curved interfaces in the model. As seen in the previous section for the 1.5D case, the waterbottom reflection here suffers from the butterfly-type artifact. Furthermore, the straight and curved lines not related to aperture artifacts are mappings of numerical diffractions to the coupled plane wave domain. The numerical diffractions are smeared out over a large spatial extent.

In order to show the predicted internal multiples in a concise manner, they are compared to the input in the shot domain directly. The comparisons are extracted at three distinct shot locations $x_s \in \{500, 2500, 4500\}\ m$. The corresponding figures are shown figures number 6.15 through 6.17. The input gathers are consistently shown in the lefthand panel, although do note that either dataset might be reversed in order to ease any interpretation. Also do note that colorscale produced in the input gathers is heavily clipped in order to emphasize the internal multiples present.

Although the internal multiple predictions seem quite accurate kinematically, the internal multiple hyperboloids seem somewhat more 'wavy' than what is seen in the input data. Furthermore, som of the smaller scale features are not exactly reproduced due to the inferior frequency bandwidth of the internal multiple prediction. This effect is largely due to not performing source-signature deconvolution before prediction. The linear artifacts seen above the first internal multiple are most likely due to some non-linear interaction of artifacts at the far ends of the coupled plane wave domain.

More worryingly, however, is the situation seen in figure 6.17 where the two first primaries are also a part of the prediction. This arises due to combination of the butterfly artifact of the first primary and the other two primary events themselves. Prediction of primaries is naturally highly unwanted as this can lead to damaging results after adaptive subtraction.

Figure 6.10 Johan Sverdrup synthetic velocity and density models.

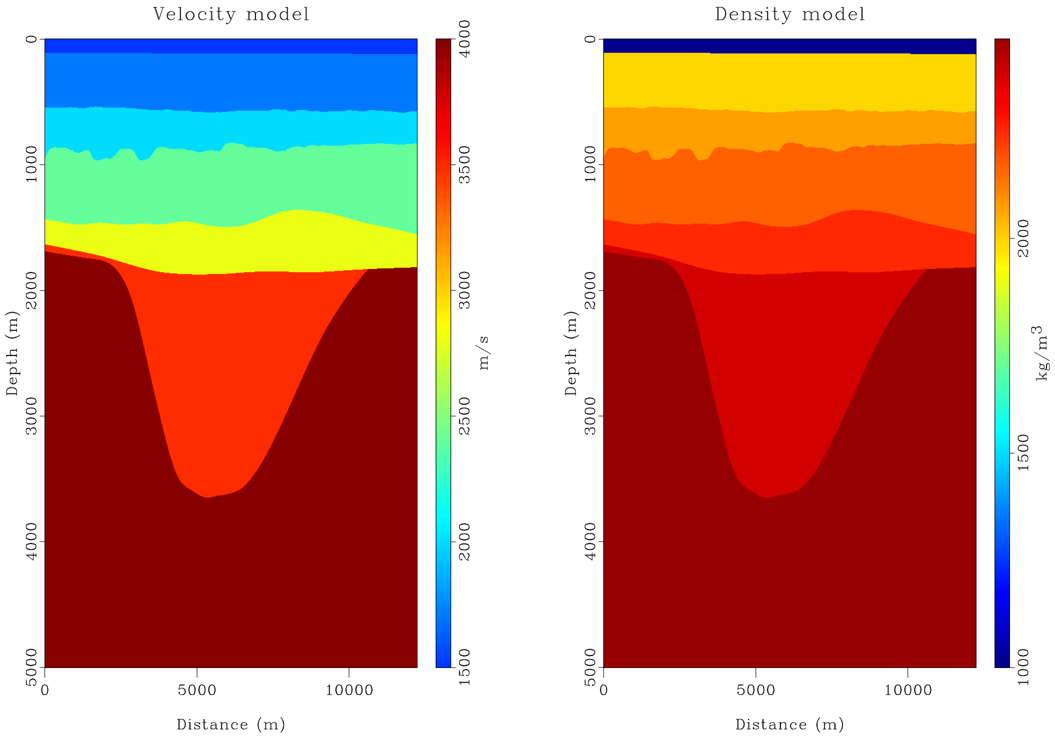


Figure 6.11 Velocity and density models in area of figure 6.10 used for internal multiple prediction study

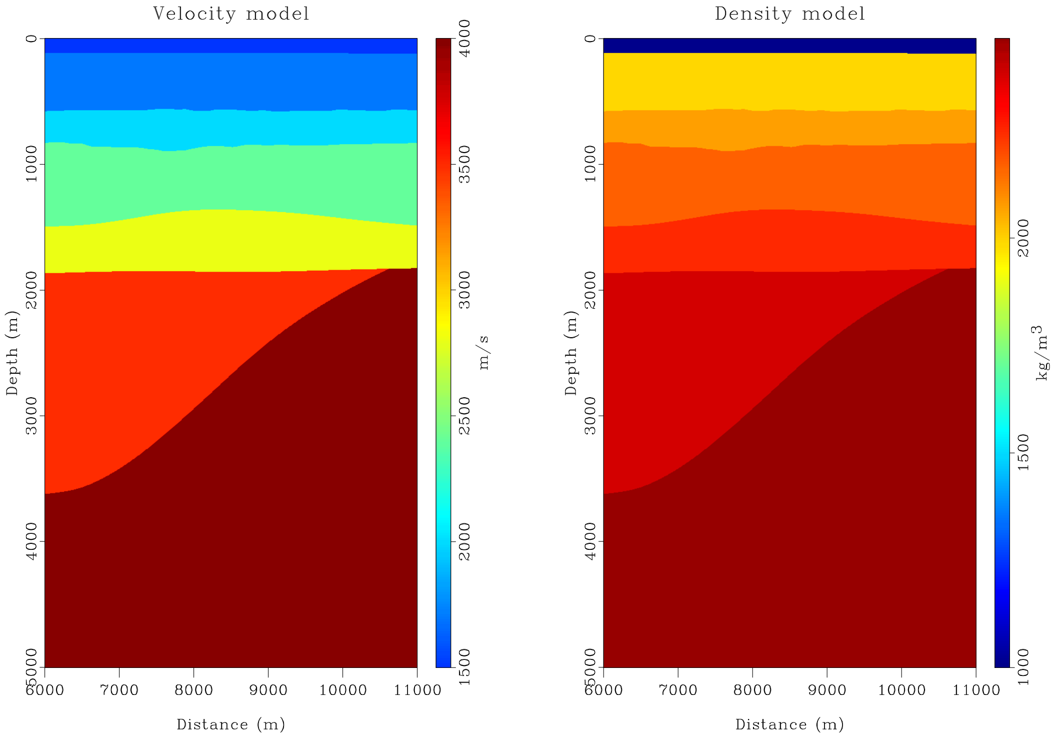


Figure 6.12 Cube of shots modelled using the velocity and density models in figure 6.11

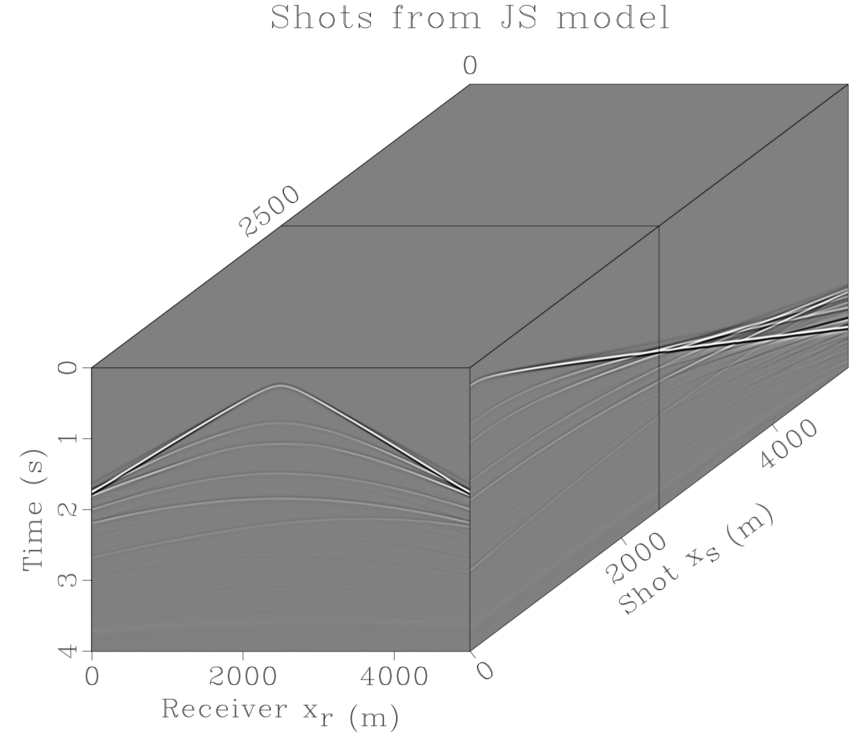


Figure 6.13 Example shot gather extracted from figure 6.12 at $x_s = 2500\ m$ in order to show numerical diffractions

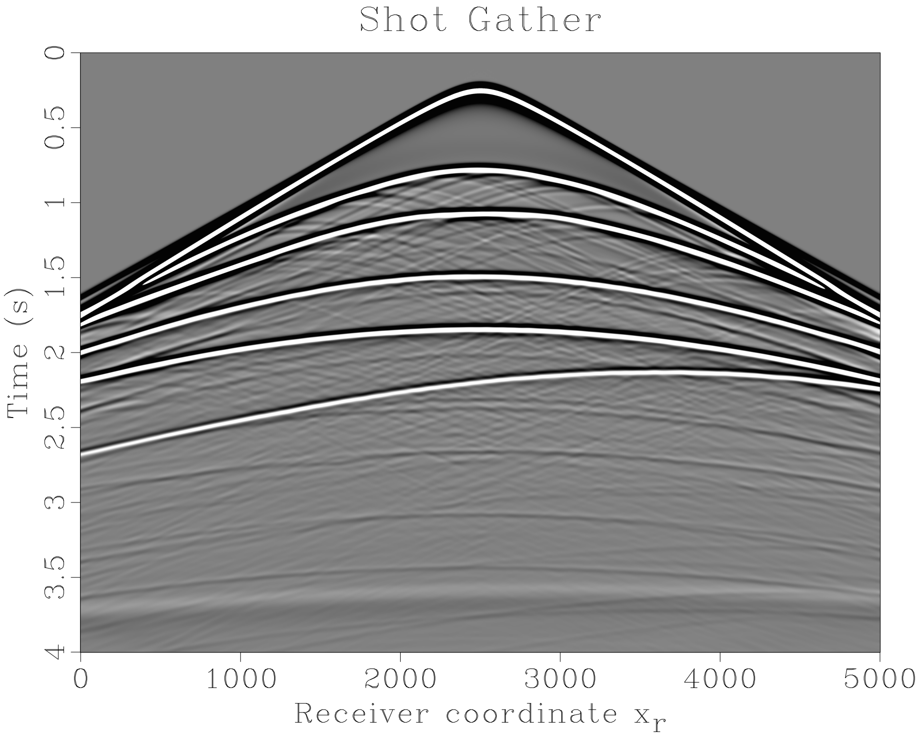


Figure 6.14 Coupled plane wave domain representation of 6.12 using *sfradon* extracted at slownesses $p_s \in \{-0.2, 0.0, 0.2\}\ s/km$

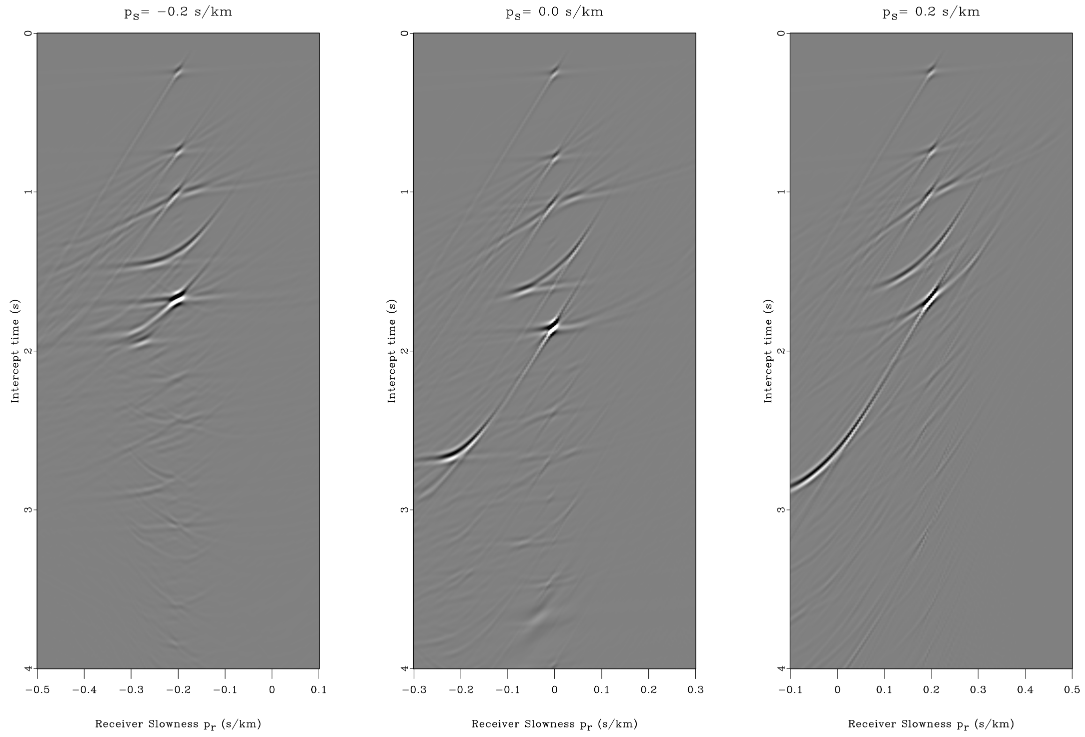


Figure 6.15 Comparison of input (left) and predicted internal multiples (right) using a standard Radon transform. Shot gathers extracted at $x_s = 500\ m$.

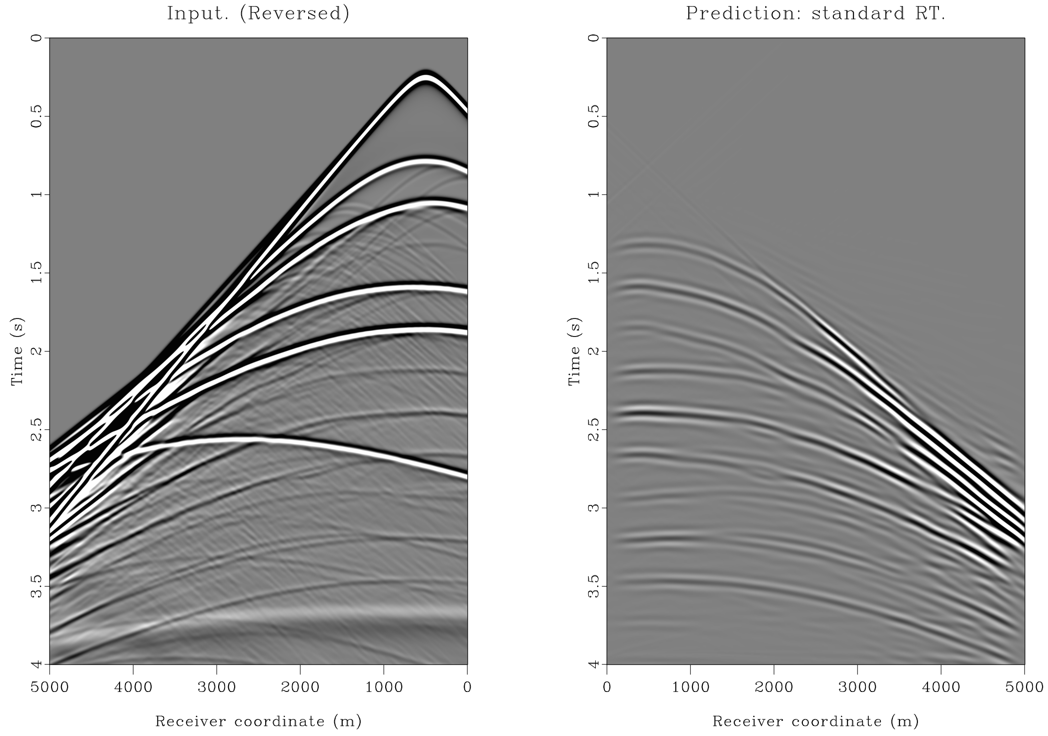


Figure 6.16 Comparison of input (left) and predicted internal multiples (right) using a standard Radon transform. Shot gathers extracted at $x_s = 2500\ m$.

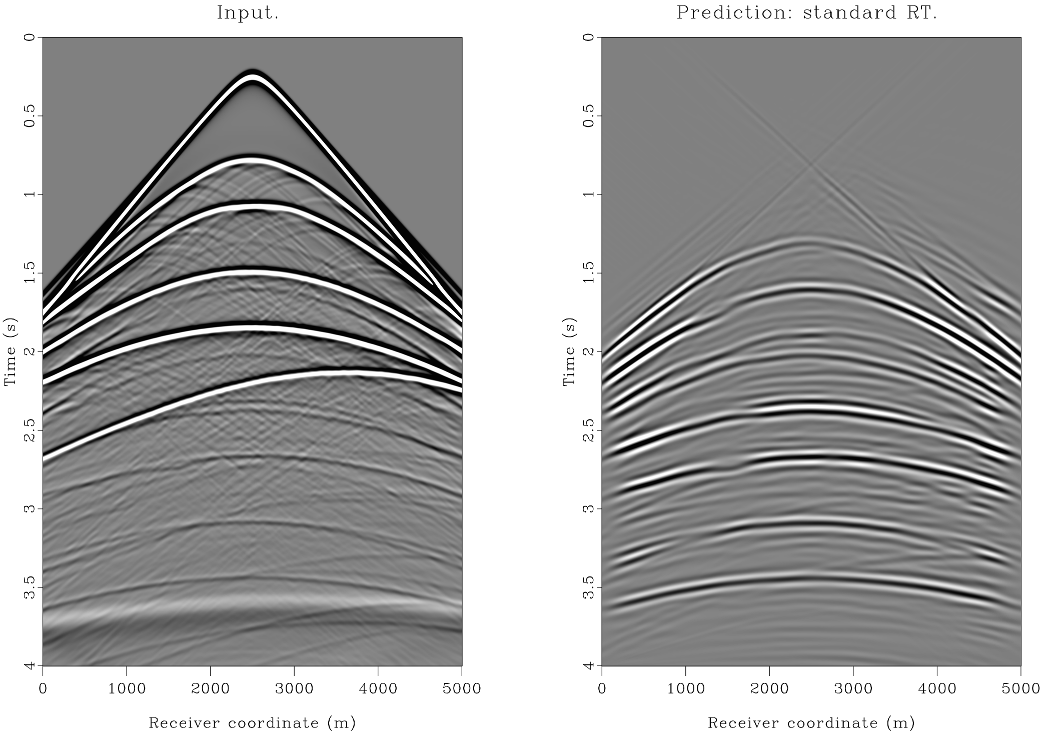
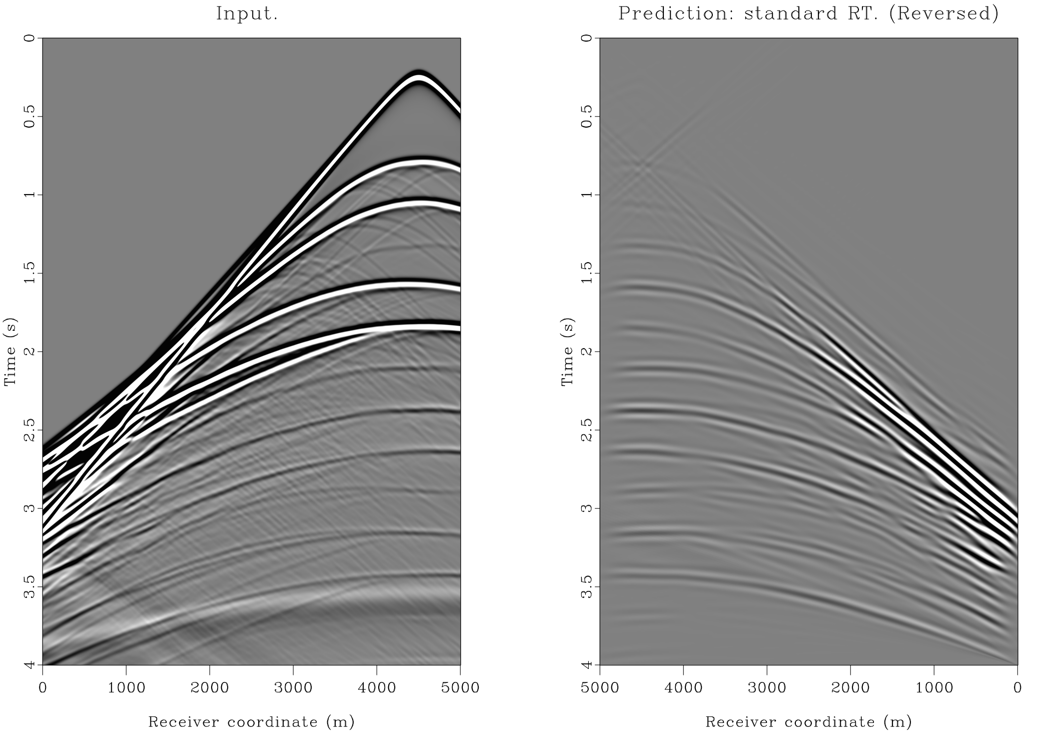


Figure 6.17 Comparison of input (left) and predicted internal multiples (right) using a standard Radon transform. Shot gathers extracted at $x_s = 4500\ m$.



6.2.2 Using high-resolution Radon transforms

In order to test the effect of using high-resolution transforms, the modelled shot gathers are again transformed to the coupled plane wave domain using algorithm 1. Only the last Radon transform in the algorithm (step III) was computed usying the sparse $\ell_2 - \ell_1$ hybrid time-frequency domain transform described in chapter 4. This approach was chosen in order to prevent over-sparseness in the coupled plane wave domain representation. Figure 6.18 compares the coupled plane wave domain data at $p_s = 0.0\ s/km$ using the standard and the sparse transforms. Unsurprisingly, the sparse transformed data show less smearing and butterfly-type aperture artifacts. It should be noted that graphical representation of data in this sparse domain, especially with sparseness promoting transforms, is particularly difficult with respect to colorbar levels and amplitude clipping.

The most evident challenge in using high-resolution variants of the linear Radon transform is to achieve a good compromise between artifact prevention and avoiding over-sparseness. Without routines that automatically and robustly estimate regularizarion parameters, this can be somewhat challenging. It would have been especially beneficial to have some automatic regularization because empirically the high-dip components in the coupled plane wave domain would benefit from more regularization compared to low-dip components. Because all source slownesses are calculated for each independent receiver slowness in step III of algorithm 1, automatic regularization routines would have been able to specify independent levels of regularization to the low and high ends of the receiver slowness range.

Similarly as in the previous subsection, the internal multiple predictions are compared directly to the input data in the shot gather domain. Figures 6.19 through 6.21 show the input compared to the internal multiple prediction at the same shot locations, namely $x_s \in \{500, 2500, 4500\}\ m$. The temporal support of the internal multiple predictions appear to give a good match of that in the input data, a virtue of the sparse Radon transform's ability to promote sparseness in time. Even more interesting is perhaps the observation that even fine-scale features in the internal multiples are accurately modelled. In places, up to four internal multiples coincide at the same arrival time at near offsets, while diverging at larger offsets. These types of very detailed features are accurately reproduced in the internal multiple prediction.

The internal multiple predictions using standard and sparse Radon transforms are also compared to each other. The comparisons are extracted at the very same shot locations studied previously, and are graphically depicted in figures 6.22 through 6.24. Kinematically, the two predictions appear very similar. The sparse transform prediction shows an improved frequency bandwidth. Therefore, many of the fine-grained features appear to be more accurately represented. At the shot gather at $x_s = 4500\ m$ in figure 6.24 one can also observe that the pseudo-events in the standard prediction are not present in the sparse prediction. Indeed, observing again the comparison of the input datasets in figure 6.18, it is readily observed that the waterbottom reflection shows no butterfly artifact and focuses very well. A slice through the shot gather depicted in figure 6.24 at receiver position $x_r = 5000\ m$ is shown in figure 6.25. The first 1.25 seconds have been muted in order to compare only the events not corresponding to obvious artifacts. The waveform of the standard prediction does not resemble a time-shifted zero-phase wavelet, rather many events are closer to a mixed-phase wavelet. It also shows significant energy in locations where there are no internal multiples. This might be an effect of numerical diffractions acting as a type of noise. The sparse RT prediction is more similar to a series of spiked events with a plausible, time-shifted zero-phase wavelet.

6.2.3 Results from adaptive subtraction

It is certainly interesting to study how the predictions, using standard and sparse Radon transforms on the input, fare in adaptive subtraction procedures. Ultimately, the end goal of internal multiple predictions to attenuate any internal multiples present in the input data. The input and adaptive subtraction result when using the prediction based on standard Radon transforms is shown in figure 6.26. Idem, the same figure when using the prediction based on sparse Radon transforms in figure 6.27. In general, both predictions yielded adaptive subtraction results with significant internal multiple attenuation. Recall that the internal multiples related to the steeply dipping underburden, i.e. the last primary, are in general not modelled, due to restriction of the dip-range in the internal multiple modelling according to the dips of the overburden. Their presence after adaptive subtraction is correct behaviour.

The adaptive subtraction filter used is estimated based on an ℓ_2 data norm. By using the ℓ_2 norm one implicitly assumes that the filtered data (consisting of the primaries) is orthogonal to the internal multiples, and has minimum energy (Guitton and Verschuur, 2004). This assumption may not be strictly true. Application of a subtraction filter optimal in the ℓ_2 sense can therefore cause internal multiple energy to leak into primaries and vice versa. These kind of artifacts are partly visible both in figures 6.26 and 6.27. When comparing the two adaptive subtraction results, in figure 6.28, one can observe that the result using the sparse transform prediction as internal multiple model is significantly more successful at internal multiple removal, with less artifacts.

The author wishes to thank Shruti Gupta for running the adaptive subtraction procedure.

Figure 6.18 Comparisons of input in the coupled plane wave domain using a standard Radon transform (left) and a sparse, hybrid time-frequency variant (right). Extracted at $p_s = 0.0\ s/km$

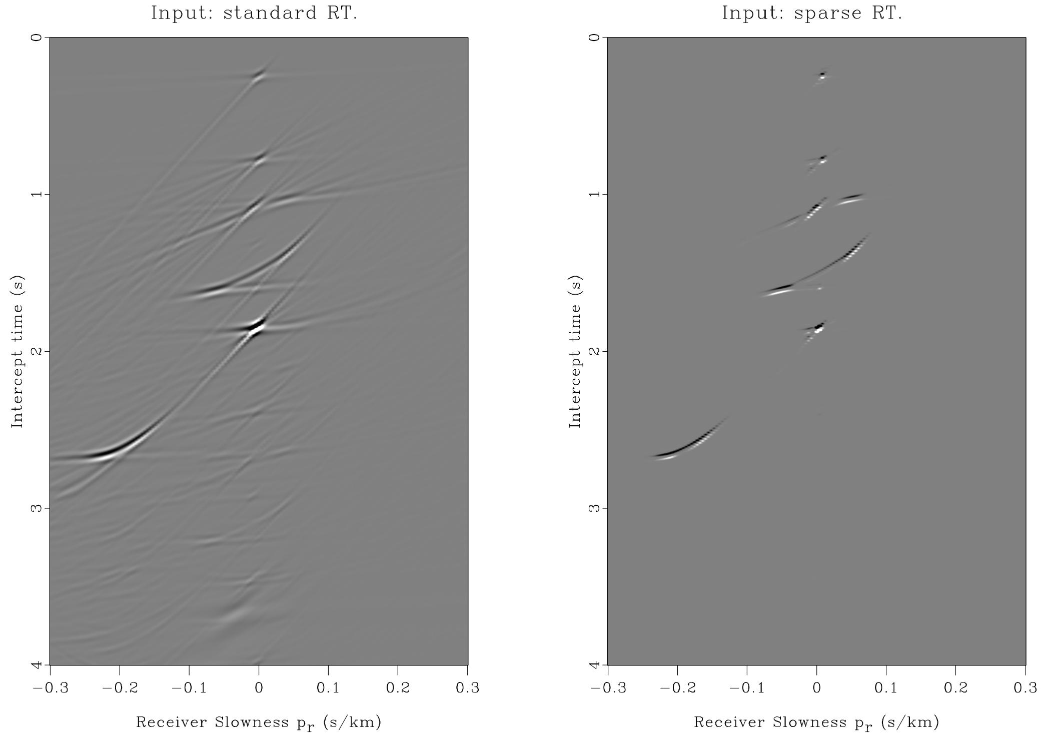


Figure 6.19 Comparison of input (left) and predicted internal multiples (right) using a sparse, hybrid time-frequency Radon transform. Shot gathers extracted at $x_s = 500\ m$.

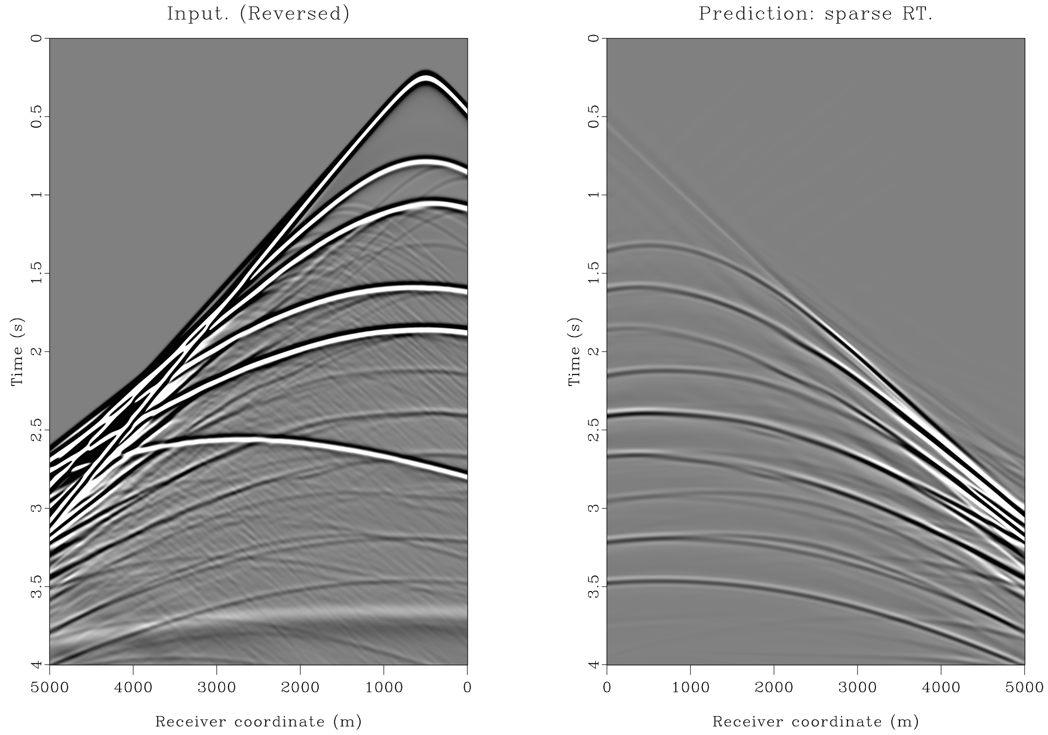


Figure 6.20 Comparison of input (left) and predicted internal multiples (right) using a sparse, hybrid time-frequency Radon transform. Shot gathers extracted at $x_s = 2500\,m$.

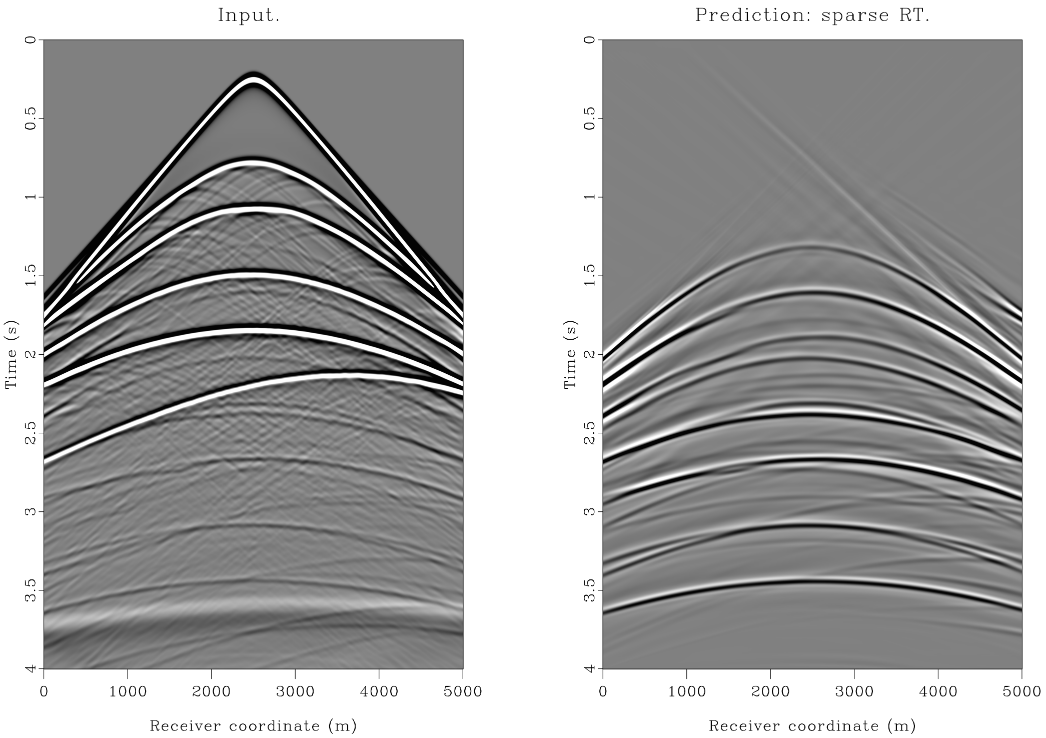


Figure 6.21 Comparison of input (left) and predicted internal multiples (right) using a sparse, hybrid time-frequency Radon transform. Shot gathers extracted at $x_s = 4500\,m$.

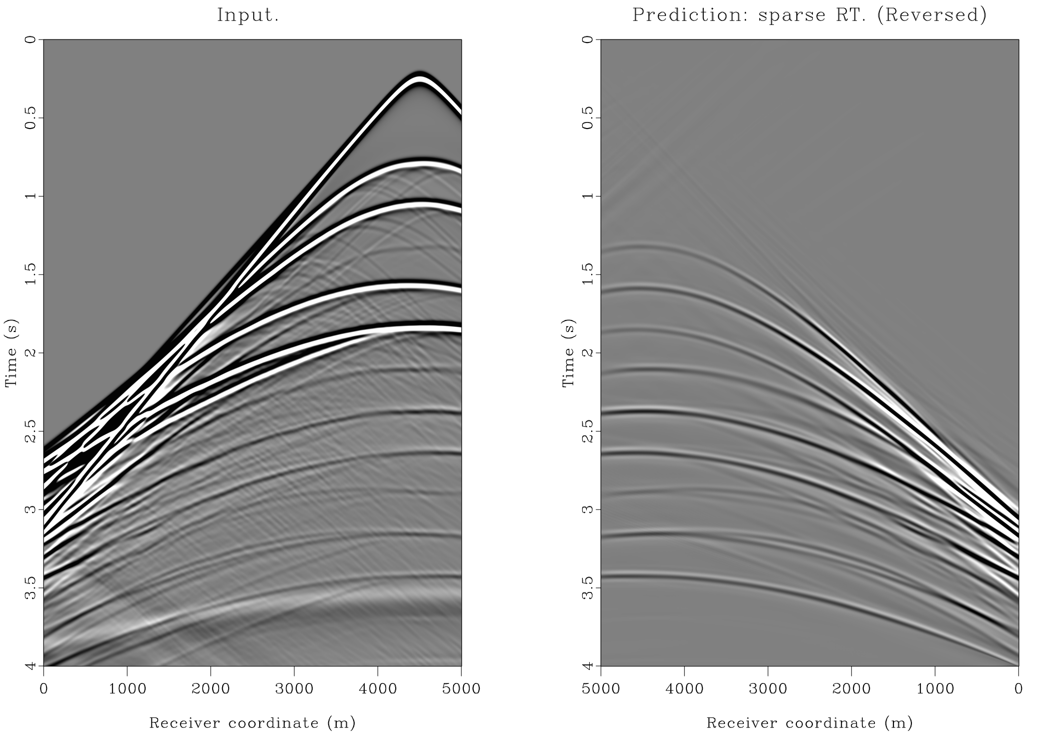


Figure 6.22 Comparison of internal multiple predictions using standard Radon transform (left) and sparse, hybrid time-frequency Radon transform (right). Shot gathers extracted at $x_s = 500\,m$.

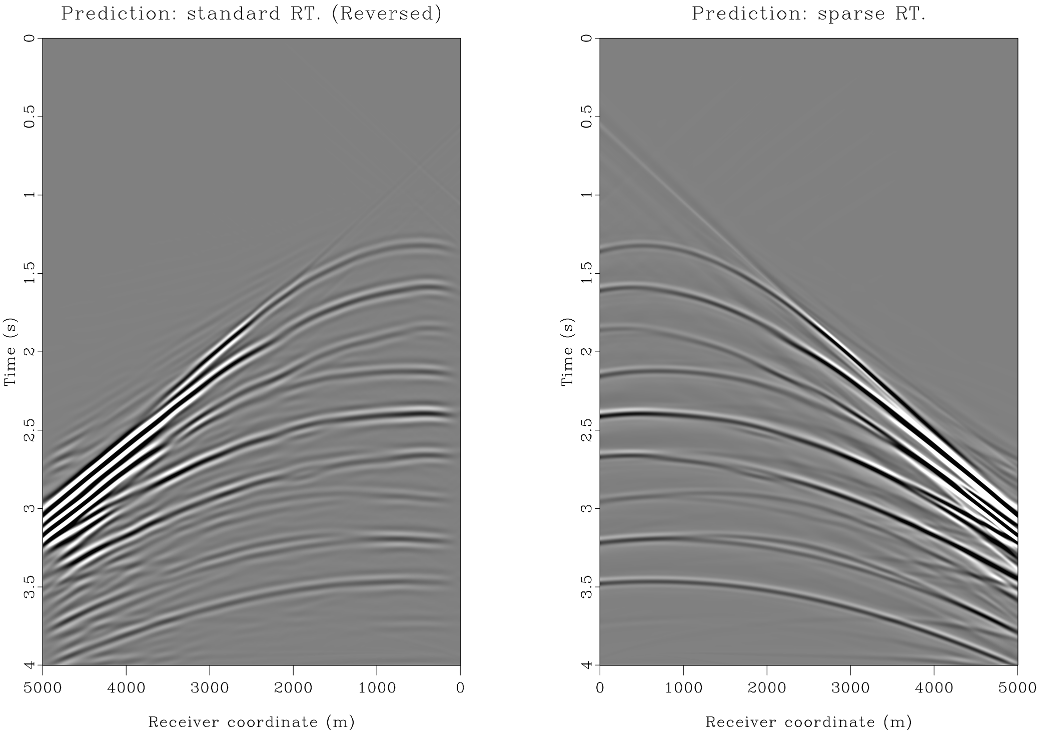


Figure 6.23 Comparison of internal multiple predictions using standard Radon transform (left) and sparse, hybrid time-frequency Radon transform (right). Shot gathers extracted at $x_s = 2500\,m$.

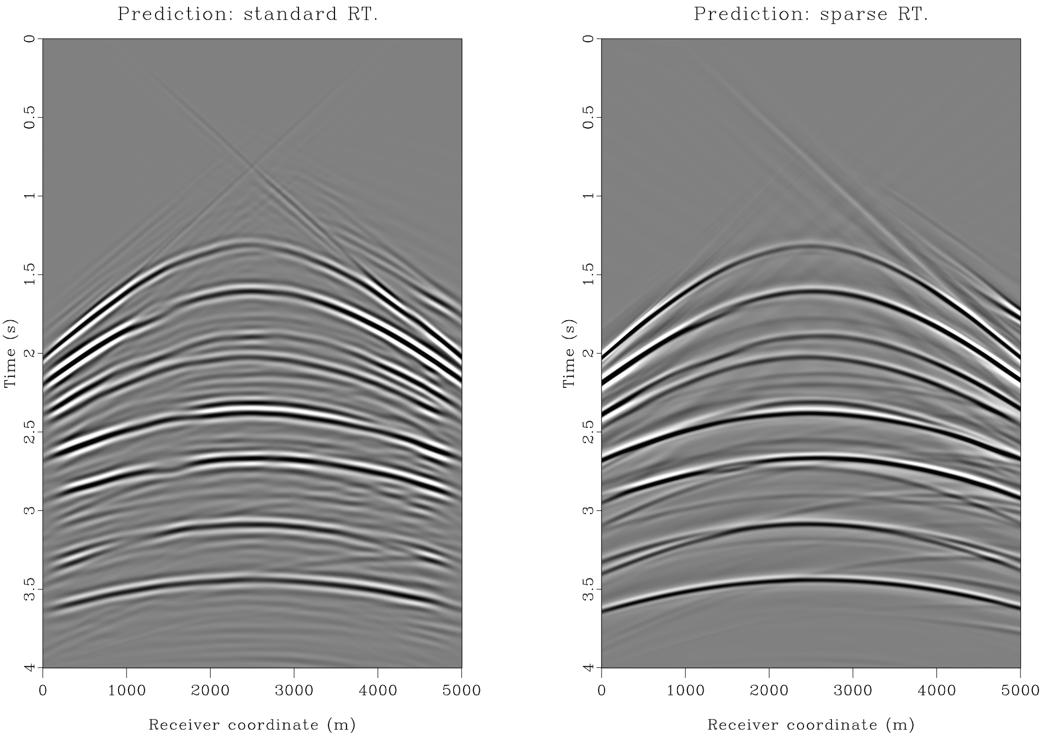


Figure 6.24 Comparison of internal multiple predictions using standard Radon transform (left) and sparse, hybrid time-frequency Radon transform (right). Shot gathers extracted at $x_s = 4500\,m$.

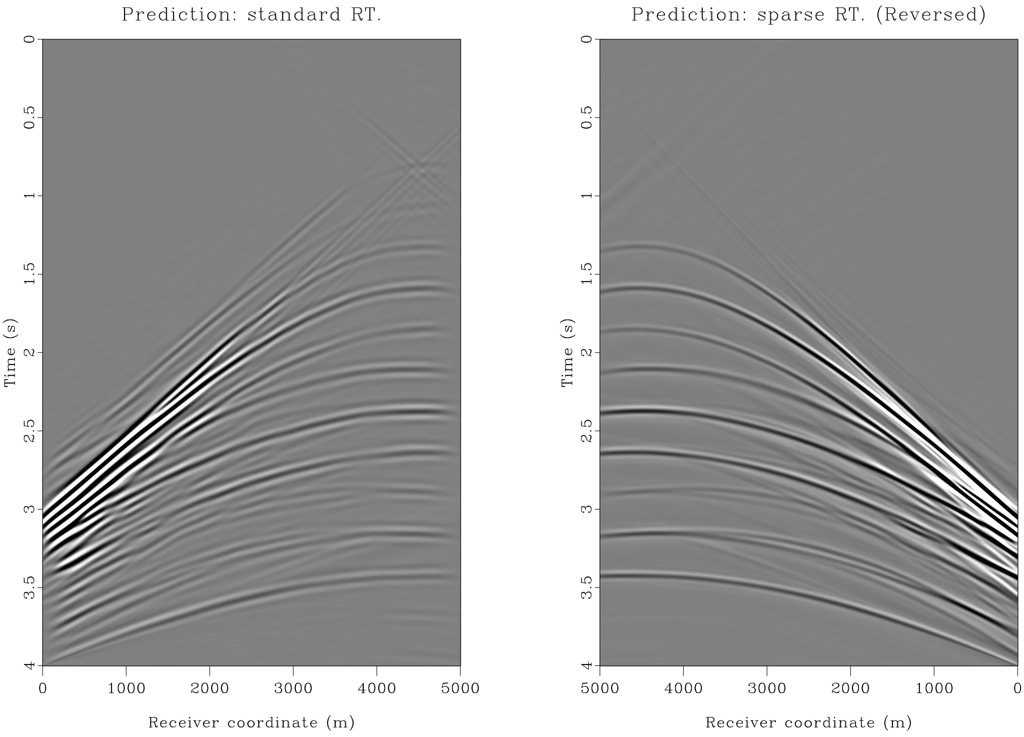


Figure 6.25 Trace comparison of internal multiple predictions using standard Radon transform (left) and sparse, hybrid time-frequency Radon transform (right). Extracted at $x_r = 5000\,m,\ x_s = 4500\,m$.

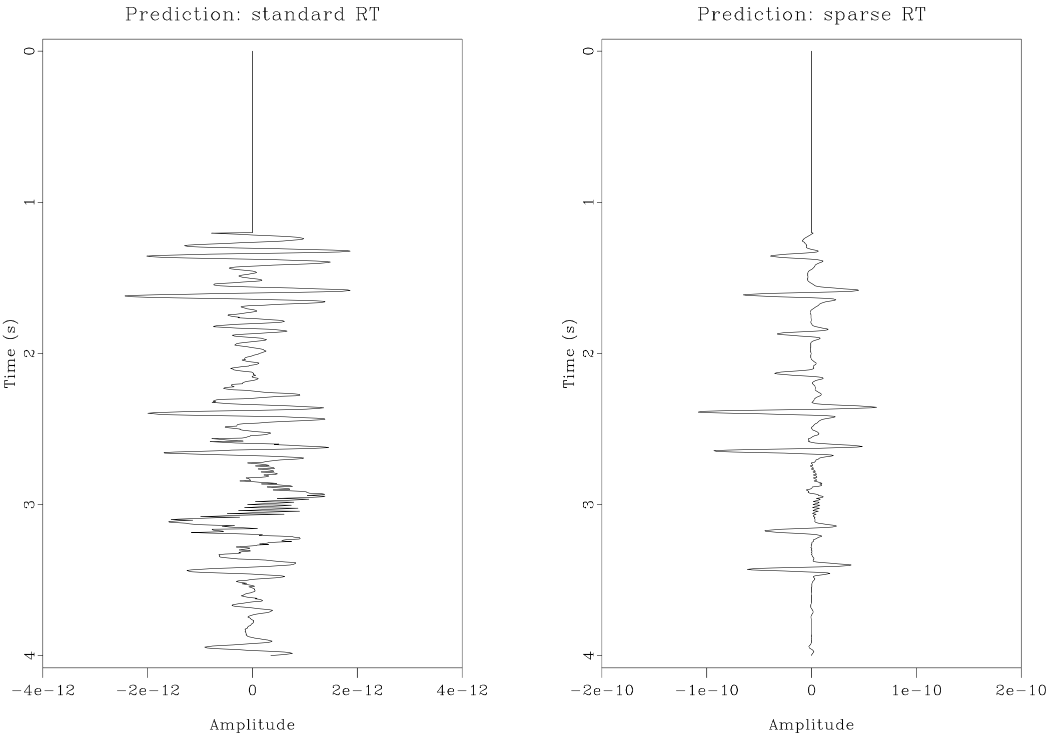


Figure 6.26 Input (left) and adaptive subtraction result (right), when using internal multiple model predicted from input data transformed with a standard linear RT. Extracted at $x_s = 1500\, m$.

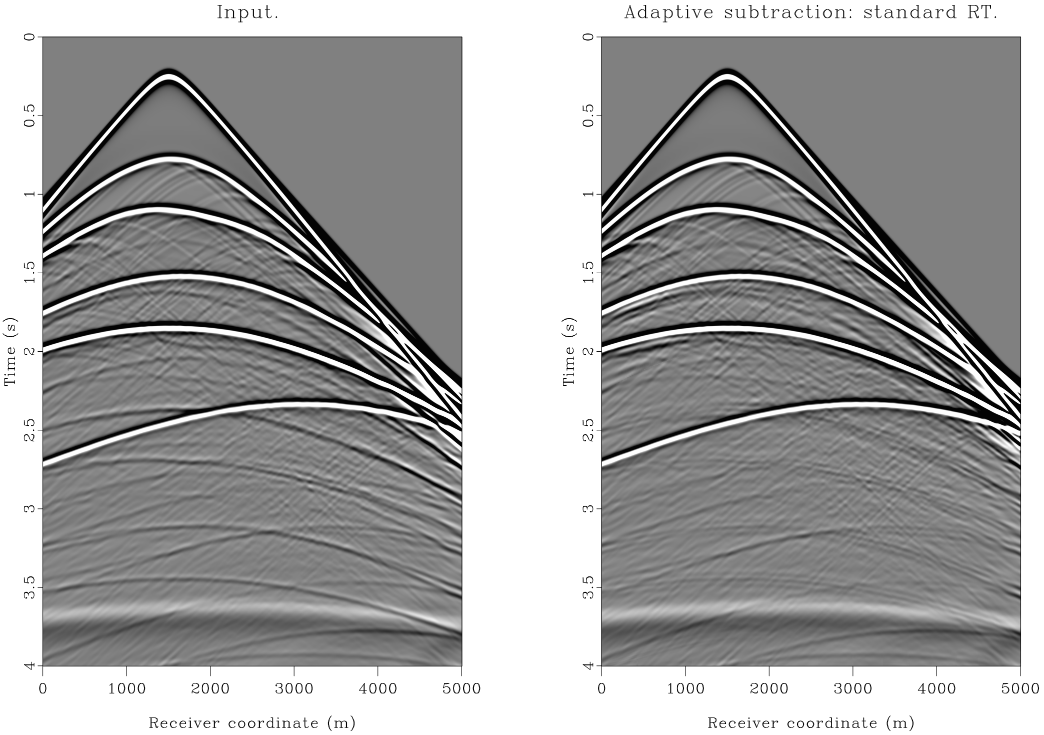


Figure 6.27 Input (left) and adaptive subtraction result (right), when using internal multiple model predicted from input data transformed with a sparse linear RT. Extracted at $x_s = 1500\, m$.

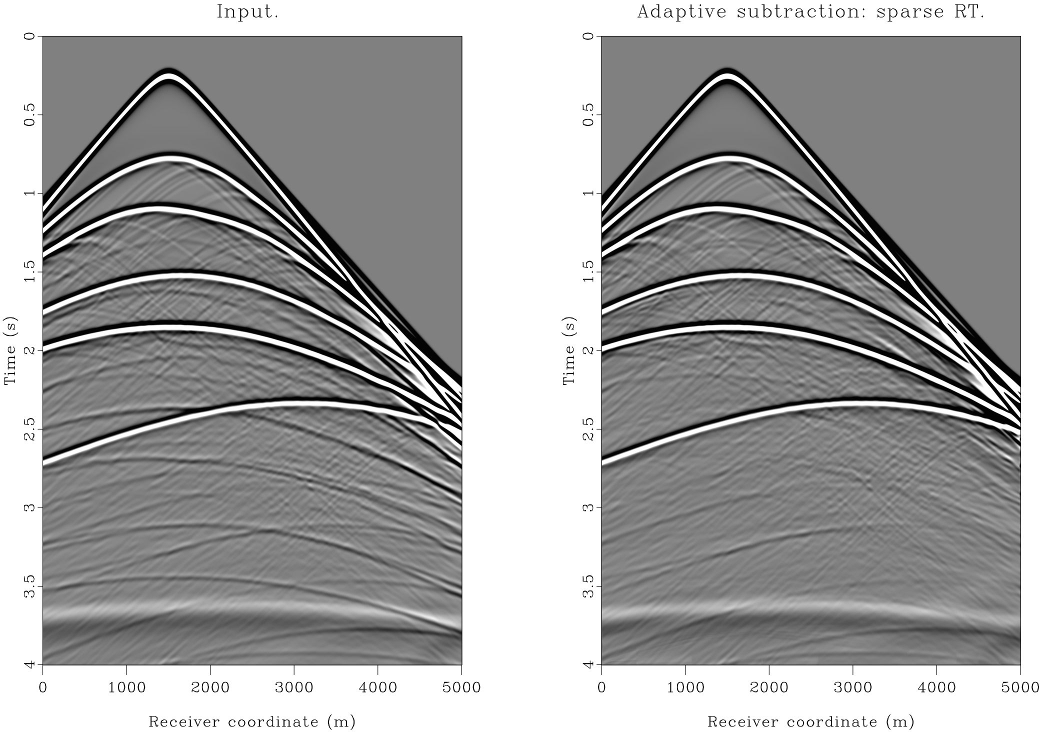
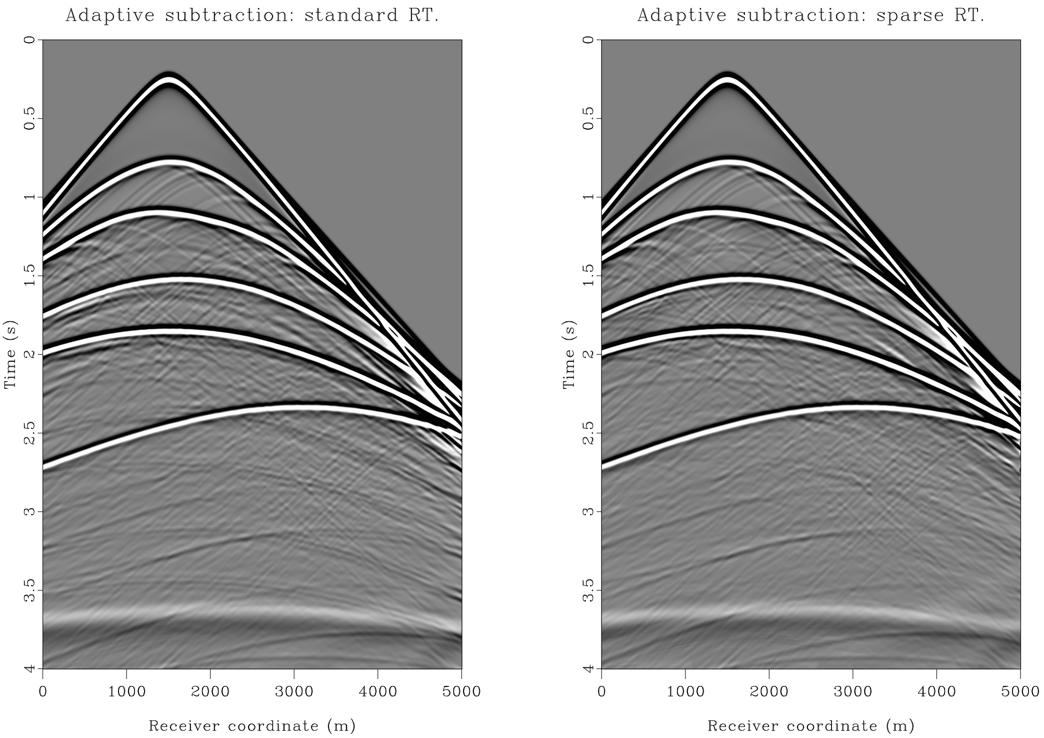


Figure 6.28 Comparison of adaptive subtraction results, using prediction with a standard linear RT (left) and prediction with a sparse linear RT (right). Extracted at $x_s = 1500\,m$.



6.3 Test on demigrated data

In order to utilize multidimensional internal multiple prediction using migrated data, Aaker (2017) presented a workflow utilizing a demigration-migration scheme. As a proof of concept of this idea, it is tested on synthetic, demigrated data.

The data were originally (i.e. pre-migration) modelled via a Finite Difference method utilizing Johan Sverdrup-based velocity and density models similar to those shown in figure 6.10. The migration procedure used was Kirchhoff based. Idem, the demigration procedure was similar to kinematic Kirchhoff modelling, yet with a spherical divergence factor applied. The implementation of this particular demigration procedure could not provide the acquisition geometry required for prediction in 2D. This trial-run was therefore limited to usage of the 1.5D predictor.

An example of a demigrated gather is shown in figure 6.29. Among the difficulties in using this for internal multiple prediction is that the mute applied is harsh, and in particular the waterbottom reflection is barely preserved. There are also some vertical stripes present, although their effect on the prediction did not appear to be critical. In general, some tweaking of amplitudes were required in order to reach satisfying internal multiple model, especially on the waterbottom reflection.

The prediction result using a standard linear Radon transform is shown in figure 6.30, where it is compared to the input gather. Similarly, the prediction result using the implemented sparse linear RT is shown in figure 6.31. In general, the predicted internal multiple models appears to match those present in the input in a quite decent manner. Note that discrepancies are in general expected as a 1.5D predictor has been utilized on a dataset originating from a 2D earth model. Furthermore, high dip components (often corresponding to larger offsets) are largely not available for modelling due to the mute present in the input gather. Therefore, these components of the internal multiples will not be retrieved entirely correctly. However, the resulting predictions are still very fair.

For completeness, the two internal multiple predictions are compared to each other in figure 6.32.

The author wishes to thank Emin Sadikhov and Shruti Gupta for providing the demigrated synthetic data.

Figure 6.29 Synthetic, demigrated gather.

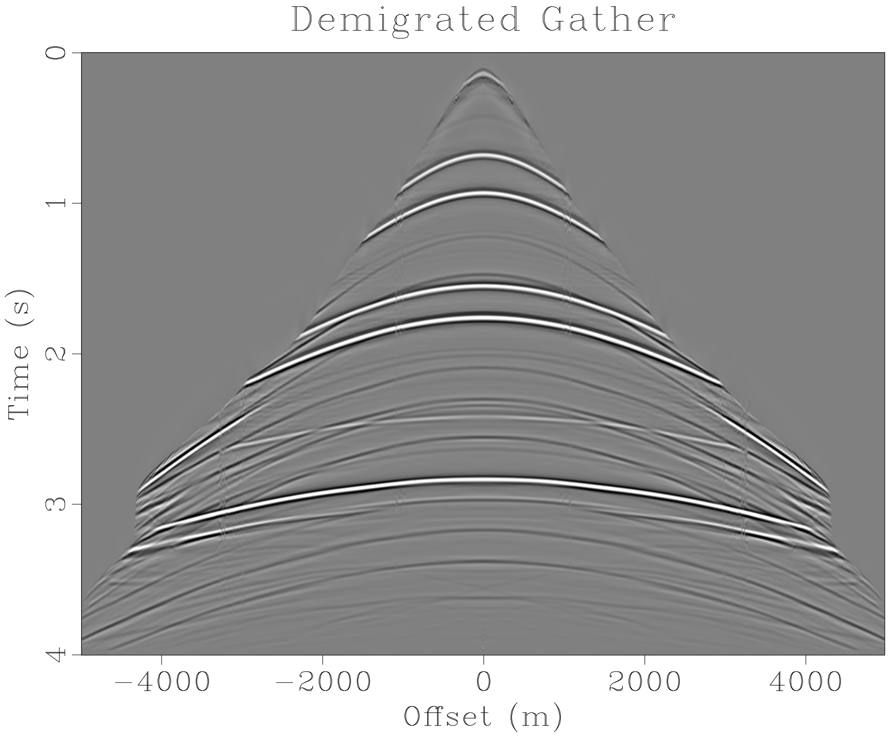


Figure 6.30 Comparison of input gather (left) and internal multiple prediction using a standard Radon transform on input (right).

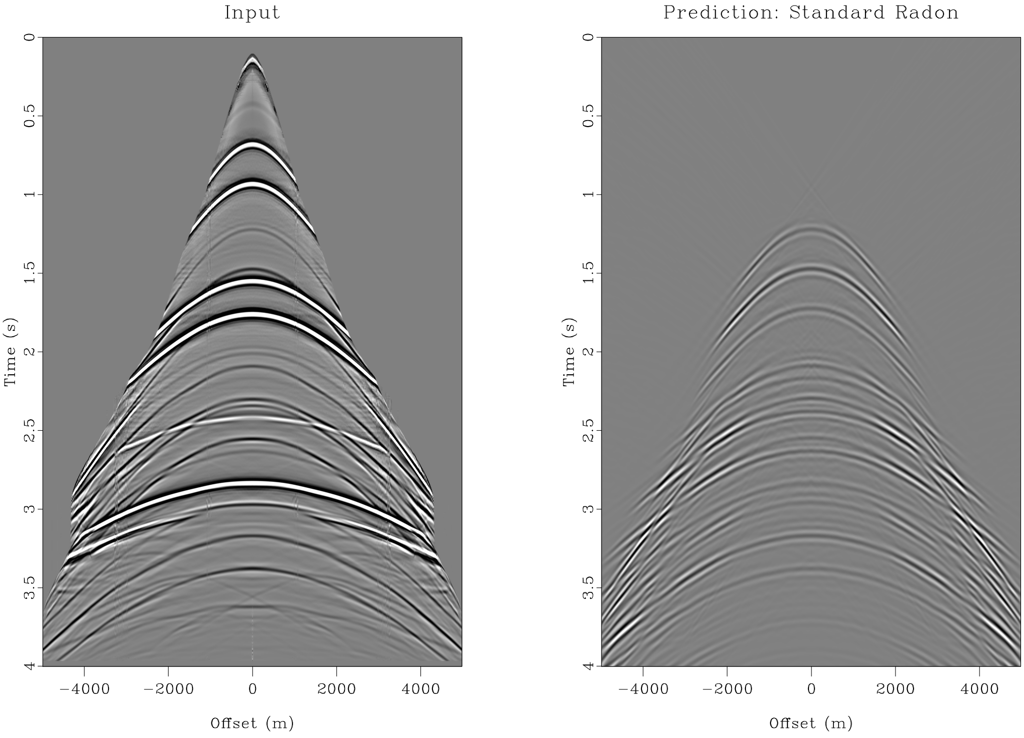


Figure 6.31 Comparison of input gather (left) and internal multiple prediction using a sparse Radon transform on input (right).

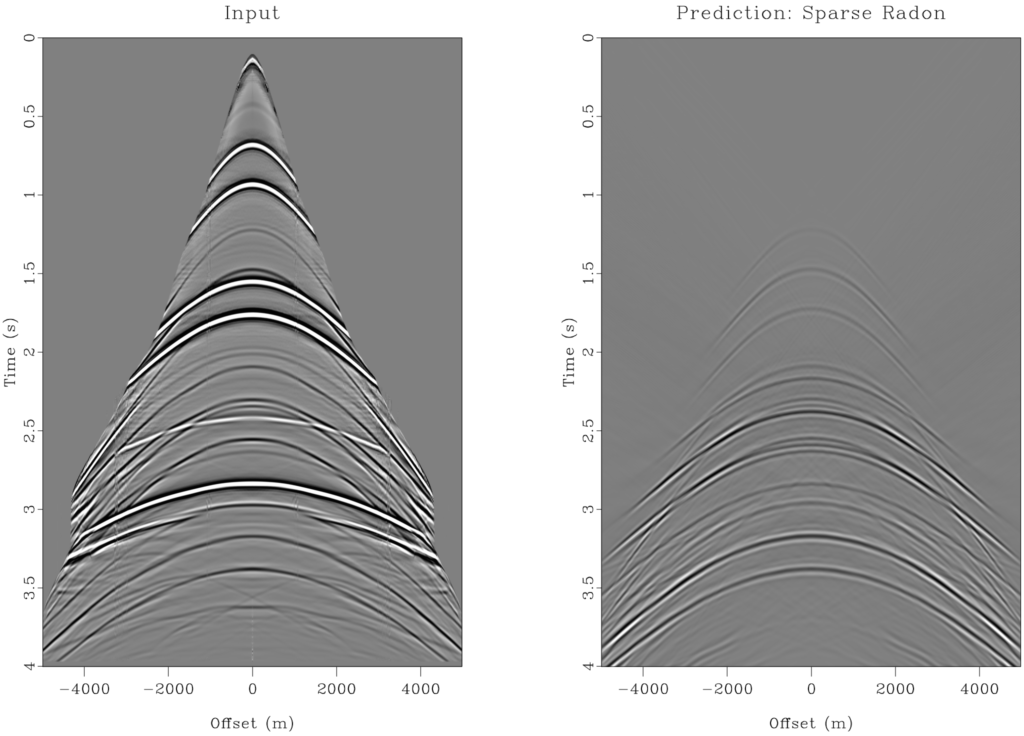
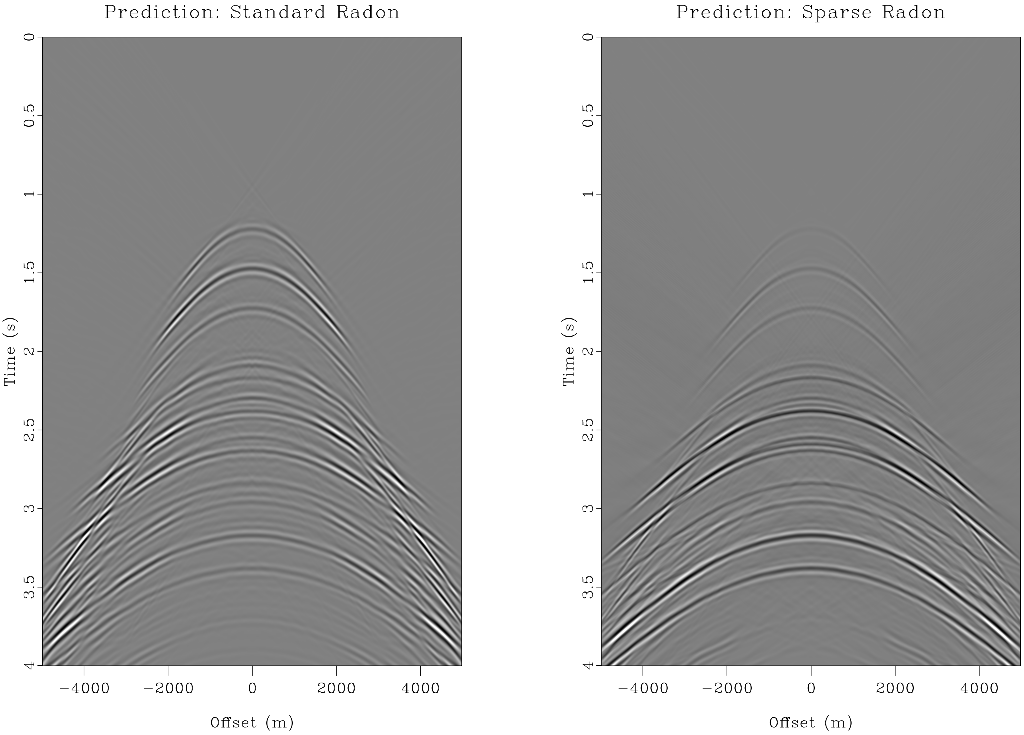


Figure 6.32 Comparison of internal multiple predictions using a standard transform (left) and a sparse transform (right) on input.



Chapter 7

Limitations and relation to other methods

7.1 Limitations of the inverse scattering series internal multiple predictor

The most profound limitation of the internal multiple predictors studied in this thesis seem to be their need of transform domains. Artifacts may arise in the pseudodepth-wavenumber formulation, as the parameter ϵ is not stationary across this domain (Sun and Innanen, 2015). The plane wave formulations do not explicitly suffer from this deficiency. However, due to the non-orthogonality of the linear Radon transform, the input data in plane-wave domains often include certain artifacts. The 1.5D plane wave domain predictor does seem somewhat robust to these transform limitations. In multidimensional plane wave formulations, however, artifacts may span large parts of the domain. On the contrary, the actual data tend to have a sparse representation. Artifacts and data may therefore combine to create significant artifacts in the internal multiple predictions. The use of transform domains also imposes the need of a special sampling geometry, in the case of multidimensional prediction. It should be noted that the demigration procedure in Aaker (2017) can effectively deal with this.

The amplitudes predicted by the internal multiple predictor b_3^{IM} are only approximate (Weglein et al., 2003). The predicted amplitude is always less than the true amplitude of the internal multiple. The so-called attenuation factor controlling this behaviour is due to the inclusion of extra transmission terms in the predicted internal multiple (Ramírez et al., 2005). Some waveform mismatches are also to be expected from this prediction algorithm. This arises because three subevents, with three wavelets, are combined to calculate contributions to the internal multiple prediction. Therefore, the frequency bandwidth and waveform of predicted internal multiples will not match what is found in their true counterparts. The amplitude and waveform mismatches require the use of an adaptive subtraction routine in order to suppress internal multiples in the input data.

The ISS internal multiple prediction criterion consists of a Lower-Higher-Lower relationship in pseudodepth or intercept time. Compared to general scattering coda, there will exist certain geological scenarios in which the LHL criterion will fail to describe an internal multiple. Examples of this includes models that may give rise to complex scattering, such as salt domes.

In its multidimensional formulations, the inverse scattering series internal multiple predictors have high computational cost functions. Chapter 5 gave an overview of algorithmic simplifications and implementational strategies intended to give practical value to the two-dimensional internal multiple predictor. Due to the computational cost, full three dimensional inverse scattering series internal multiple prediction is currently not viable with deterministic evaluation of the prediction integrals.

7.2 Relation to other methods

Among the key strengths of the internal multiple predictors derived from the inverse scattering series is that they are virtually entirely data driven. The only input needed is the velocity of the medium most proximal to the receivers and sources. In the marine case this corresponds to the waterspeed velocity. Several other methods proposed to predict and/or implicitly treat internal multiples rely on a subsurface velocity and/or reflectivity model. As an extension of the Delft approach to SRME, the Common Focus Point approach formulated by Berkhout and Verschuur (2005) requires a velocity model for inverse wavefield extrapolation and a reflectivity model to estimate the subsurface locations of internal multiple generators. The model requirements and restrictments in the method of Berkhout and Verschuur (2005) may be a key reason as to why it did not gain traction in the industry.

Internal multiples need not to be explicitly predicted and removed, they may also be correctly treated in redatuming and/or imaging. By solving the three dimensional Marchenko equations (Wapenaar et al., 2014) one can retrieve the full[1] scalar Green's function in a data driven manner. This enables the possibility to not just inverse extrapolate a recorded wavefield but also to focus it at an arbitrary location inside the (generally) unknown medium of interest. Applications of Marchenko derived methods to treat internal multiples have very recently appeared, e.g. in Staring et al. (2017). The current formulation of the Marchenko equations requires a velocity model in order to calculate the direct part of a transmitted Green's function. In this case, the need for a velocity model stems from the ill-posedness of the 3D Marchenko equation, which is treated by a translation of Marchenko's original ansatz to three dimensions. Note also that the validity of the Marchenko ansatz for arbitrary three-dimensional earth models may not hold. Non-scalar formulations, e.g. elastodynamic, are currently limited by the very same ansatz.

[1] I.e. not just the approximate Green's function due to invocation of the inverse Born approximation.

Löer et al. (2016) derived expressions for data driven internal multiple prediction using Source-Receiver Interferometry. Similar to the ISS approach, the SRI predictor can not provide the exact amplitude of the internal multiples. In its current formulation, the SRI internal multiple predictor does not require any knowledge of the medium of interest. The internal multiple generation criterion is here a Lower-Higher-Lower relationship in two-way traveltime, t. For 1D applications such a condition is by causality bound to hold. However, for multidimensional approaches it remains to be seen how much of a limitation this generation criterion will provide for reflection data aquired in arbitrarily inhomogenous earth models.

Chapter 8

Conclusions and further work

8.1 Conclusion

The work performed in relation to this thesis has uncovered the following:

- Within the work of this thesis the author has implemented, in the C and C++ programming languages, Inverse Scattering Series 2D and 1.5D internal multiple predictors, as well as a high resolution linear Radon transform.
- Implementation of the linear Radon transform was not part of the original problem specification, but rather implemented as a method to yield improved internal multiple prediction results.
- While the coupled plane wave formulation of the ISS internal multiple predictor shows quite stationary values of the search-limiting parameter ϵ, other difficulties may arise due to the non-orthogonality of the linear Radon transform.
- The usage of high-resolution linear RTs attempts to ameliorate this deficiency by minimizing amplitude smearing and aperture artifacts in the (coupled) plane wave domain.
- In a hybrid time-frequency domain formulation, the implemented transforms were able to compress the temporal support of the signal as well.
- Synthetic, multidimensional data examples appear to have demonstrated that the resulting internal multiple predictions are to a better extent able to reproduce small-scale features, with the usage of high resolution RTs. Observed waveforms also appear to have given an improved match.
- Results from adaptive subtraction demonstrated that the usage of high resolution linear RTs provided improved internal multiple removal and attenuation with less artifacts.

- Using synthetic demigrated data, the concept of the demigration-prediction-migration procedure proposed in Aaker (2017) has been demonstrated to work. The procedure can therefore be applied to real data.

- The internal multiple predictors studied have high computational cost functions. The implemented predictors therefore needed to be re-expressed through mathematical analysis in order to reduce the computational cost by a factor $\mathcal{O}(N^2)$. This part of the work was largely based on the analysis of (Kaplan et al., 2004).

- The coupled plane wave domain provided a natural framework to be able to restrict the required amount of calculations in situations where subsurface reflectors related to internal multiple generation have limited angular dips.

- Concepts from high-performance computing were applied in order to squeeze performance out of the available hardware. Speedups larger than two orders of magnitude were seen, using the exact same hardware and computational algorithm.

8.2 Recommendations for future work

The recommendations listed below point at obvious limitations of this work, and what would potentially be interesting and realistic to see in a future work.

- The coupled plane wave domain internal multiple predictors should be tested with real seismic data.

- Because the different subparts of the coupled plane wave domain benefite from distinct levels of regularization, it is conjectured that automatic regularization routines for the high-resolution linear Radon transform can yield further benefits.

- Due to its high computational cost function the full 3D coupled plane wave domain internal multiple predictor has not been implemented. Most likely it will need a method distinct from Riemann summation to evaluate the high-dimensional integrals involved. Such an approach is however not easily extendable from the code implemented in this thesis.

Bibliography

Aaker, O. E., 2017. Inverse scattering series based internal multiple prediction using migrated datasets.

Araújo, F., 1994. Linear and non-linear methods derived from scattering theory: Backscattered tomography and internal multiple attenuation. Ph thesis, Universidade Federal da Bahia, Brazil,(in Portuguese).

Araújo, F. V., Weglein, A. B., Carvalho, P. M., Stolt, R., 1994. Inverse scattering series for multiple attenuation: An example with surface and internal multiples. In: SEG Technical Program Expanded Abstracts 1994. Society of Exploration Geophysicists, pp. 1039–1041.

Berenger, J.-P., 1996. Perfectly matched layer for the fdtd solution of wave-structure interaction problems. IEEE Transactions on antennas and propagation 44 (1), 110–117.

Berkhout, A., Verschuur, D., 2005. Removal of internal multiples with the common-focus-point (cfp) approach: Part 1explanation of the theory. Geophysics 70 (3), V45–V60.

Bjorck, A., 1996. Numerical methods for least squares problems. Siam.

Björck, Å., Elfving, T., Strakos, Z., 1998. Stability of conjugate gradient and lanczos methods for linear least squares problems. SIAM Journal on Matrix Analysis and Applications 19 (3), 720–736.

Bryant, R. E., David Richard, O., David Richard, O., 2003. Computer systems: a programmer's perspective. Vol. 2. Prentice Hall Upper Saddle River.

Burg, J., 1975. Maximum entropy spectral analysis. Ph.D. Dissertation, Dept. of Geophysics, Stanford University.

Candès, E. J., Romberg, J., Tao, T., 2006. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. IEEE Transactions on information theory 52 (2), 489–509.

Carvalho, P., 1992. Free-surface multiple reflection elimination method based on nonlinear inversion of seismic data. Ph.D. thesis, Ph. D. thesis, Universidade Federal da Bahia.

Cary, P. W., et al., 1998. The simplest discrete radon transform. In: 1998 SEG Annual Meeting. Society of Exploration Geophysicists.

Claerbout, J., 2014. Geophysical image estimation by example. Lulu. com.

Coates, R., Weglein, A., 1996. Internal multiple attenuation using inverse scattering: Results from prestack 1 & 2d acoustic and elastic synthetics. In: SEG Technical Program Expanded Abstracts 1996. Society of Exploration Geophysicists, pp. 1522–1525.

Dragoset, B., Verschuur, E., Moore, I., Bisley, R., 2010. A perspective on 3d surface-related multiple elimination. Geophysics 75 (5), 75A245–75A261.

Favati, P., Lotti, G., Menchi, O., Romani, F., 2014. Generalized cross-validation applied to conjugate gradient for discrete ill-posed problems. Applied Mathematics and Computation 243, 258–268.

Fokkema, J. T., van den Berg, P. M., 2013. Seismic applications of acoustic reciprocity. Elsevier.

Frigo, M., Johnson, S. G., 2005. The design and implementation of fftw3. Proceedings of the IEEE 93 (2), 216–231.

Gholami, A., Aghamiry, H. S., 2017. Iteratively re-weighted and refined least squares algorithm for robust inversion of geophysical data. Geophysical Prospecting 65 (S1), 201–215.

Golub, G. H., Van Loan, C. F., 2012. Matrix computations. Vol. 3. JHU Press.

Guitton, A., Verschuur, D., 2004. Adaptive subtraction of multiples using the l1-norm. Geophysical Prospecting 52 (1), 27–38.

Haber, E., Oldenburg, D., 2000. A gcv based method for nonlinear ill-posed problems. Computational Geosciences 4 (1), 41–63.

Hansen, P. C., 2005. Rank-deficient and discrete ill-posed problems: numerical aspects of linear inversion. Vol. 4. Siam.

Harlan, W., 1982. Avoiding interpolation artifacts in stolt migration. SEP-30: Stanford Exploration Project 103, 110.

Hestenes, M. R., Stiefel, E., 1952. Methods of conjugate gradients for solving linear systems. Vol. 49. NBS Washington, DC.

Hokstad, K., Sollie, R., 2006. 3d surface-related multiple elimination using parabolic sparse inversion. Geophysics 71 (6), V145–V152.

Huber, P. J., 1973. Robust regression: asymptotics, conjectures and monte carlo. The Annals of Statistics, 799–821.

Ibrahim, A., Sacchi, M. D., 2013. Simultaneous source separation using a robust radon transform. Geophysics 79 (1), V1–V11.

Intel, 2011 (accessed February 2018). Introduction to intel advanced vector extensions. https://software.intel.com/en-us/articles/introduction-to-intel-advanced-vector-extensions.

Intel, 2016 (accessed February 2018). Using intel streaming simd extensions and intel integrated performance primitives to accelerate algorithms. https://software.intel.com/en-us/articles/using-intel-streaming-simd-extensions-and-intel-integrated-performance

Intel, 2017 (accessed February 2018). Intel performance counter monitor - a better way to measure cpu utilization. https://software.intel.com/en-us/articles/intel-performance-counter-monitor.

Jaynes, E. T., 1968. Prior probabilities. IEEE Transactions on systems science and cybernetics 4 (3), 227–241.

Kaplan, S., Innanen, K., Otnes, E., Weglein, A., 2004. Internal multiple attenuation code development and implementation. Mission-Oriented Seismic Research Program (M-OSRP) Annual Report, 83–103.

Kernighan, B., Ritchie, D. M., 2017. The C programming language. Prentice hall.

Kreutz-Delgado, K., 2009. The complex gradient operator and the cr-calculus. arXiv preprint arXiv:0906.4835.

Liu, B., Sacchi, M. D., 2004. Minimum weighted norm interpolation of seismic records. Geophysics 69 (6), 1560–1568.

Liu, F., Sen, M. K., Stoffa, P. L., 2000. Dip selective 2-d multiple attenuation in the plane-wave domain. Geophysics 65 (1), 264–274.

Löer, K., Curtis, A., Angelo Meles, G., 2016. Relating source-receiver interferometry to an inverse-scattering series to derive a new method to estimate internal multiples. Geophysics 81 (3), Q27–Q40.

Lomont, C., 2011 (accessed February 2018). Introduction to intel advanced vector extensions. https://software.intel.com/sites/default/files/m/d/4/1/d/8/Intro_to_Intel_AVX.pdf.

Lustig, M., Donoho, D., Pauly, J. M., 2007. Sparse mri: The application of compressed sensing for rapid mr imaging. Magnetic resonance in medicine 58 (6), 1182–1195.

Ma, J., Sen, M. K., Chen, X., 2009. Free-surface multiple attenuation using inverse data processing in the coupled plane-wave domain. Geophysics 74 (4), V75–V81.

Madagascar Development Team, 2012. Madagascar Software, Version 1.4. http://www.ahay.org/.

Matson, K., Weglein, A. B., 1996. Removal of elastic interface multiples from land and ocean bottom data using inverse scattering. In: SEG Technical Program Expanded Abstracts 1996. Society of Exploration Geophysicists, pp. 1526–1530.

Moses, H., 1956. Calculation of the scattering potential from reflection coefficients. Physical Review 102 (2), 559.

Nichols, D., 1997. A simple example of a null space and how to modify it. Stanford Exploration Project Report 82, 185–192.

Nita, B. G., Weglein, A. B., 2009. Pseudo-depth/intercept-time monotonicity requirements in the inverse scattering algorithm for predicting internal multiple reflections. Communications in Computational Physics 5 (1), 163.

Ramírez, A. C., 2007. I. inverse scattering subseries for removal of internal multiples and depth imaging primaries; ii. green's theorem as the foundation of interferometry and guiding new practical methods and applications. Ph.D. thesis, University of Houston.

Ramirez, A. C., Otnes, E., 2008. Forward scattering series for 2-parameter acoustic media: analysis and implications to the inverse scattering task-specific subseries. Comput. Phys 3, 136–159.

Ramirez, A. C., Sadikhov, E., Brunsvik, F., Sigernes, L.-T., Krishna, S., et al., 2017. Internal multiples–the details that matter!: Part 1–fast prediction and attenuation for quantitative interpretation. In: 2017 SEG International Exposition and Annual Meeting. Society of Exploration Geophysicists.

Ramírez, A. C., Weglein, A. B., et al., 2005. An inverse scattering internal multiple elimination method: Beyond attenuation, a new algorithm and initial tests. In: 2005 SEG Annual Meeting. Society of Exploration Geophysicists.

Razavy, M., 1975. Determination of the wave velocity in an inhomogeneous medium from the reflection coefficient. The Journal of the Acoustical Society of America 58 (5), 956–963.

Sacchi, M. D., Ulrych, T. J., 1995. High-resolution velocity gathers and offset space reconstruction. Geophysics 60 (4), 1169–1177.

Scales, J. A., Gersztenkorn, A., Treitel, S., 1988. Fast ip solution of large, sparse, linear systems: Application to seismic travel time tomography. Journal of Computational Physics 75 (2), 314–333.

Shewchuk, J. R., et al., 1994. An introduction to the conjugate gradient method without the agonizing pain.

Staring, M., Pereira, R., Douma, H., van der Neut, J., Wapenaar, C., et al., 2017. Adaptive double-focusing method for source-receiver marchenko redatuming on field data. In: 2017 SEG International Exposition and Annual Meeting. Society of Exploration Geophysicists.

Stoffa, P. L., Sen, M. K., Seifoullaev, R. K., Pestana, R. C., Fokkema, J. T., 2006. Plane-wave depth migration. Geophysics 71 (6), S261–S272.

Stolt, R. H., Jacobs, B., 1980. Inversion of seismic data in a laterally heterogeneous medium. SEP Rep 24, 135–152.

Sun, J., Innanen, K., 2015. 1.5d internal multiple prediction in the plane wave domain. CSEG GeoConvention.

Sun, J., Innanen, K. A., 2016. Inverse-scattering series internal-multiple prediction in the double plane-wave domain. In: SEG Technical Program Expanded Abstracts 2016. Society of Exploration Geophysicists, pp. 4555–4560.

Thorbecke, J. W., Draganov, D., 2011. Finite-difference modeling experiments for seismic interferometry. Geophysics 76 (6), H1–H18.

Trad, D., 2001. Implementations and applications of the sparse radon transform. Ph.D. thesis, University of British Columbia.

Trad, D., Ulrych, T., Sacchi, M., 2003. Latest views of the sparse radon transform. Geophysics 68 (1), 386–399.

Turner, G., 1990. Aliasing in the tau-p transform and the removal of spatially aliased coherent noise. Geophysics 55 (11), 1496–1503.

Van Den Berg, E., Friedlander, M. P., 2008. Probing the pareto frontier for basis pursuit solutions. SIAM Journal on Scientific Computing 31 (2), 890–912.

Vasconcelos, I., Snieder, R., Douma, H., 2009. Representation theorems and greens function retrieval for scattering in acoustic media. Physical Review E 80 (3), 036605.

Verschuur, D. J., 2006. Seismic multiple removal techniques: Past, present and future. EAGE publications Netherlands.

Walker, D. W., Dongarra, J. J., 1996. Mpi: A standard message passing interface. Supercomputer 12, 56–68.

Wang, Y., Houseman, G. A., 1997. Point-source τ-p transform: A review and comparison of computational methods. Geophysics 62 (1), 325–334.

Wapenaar, C. P. A., 2014. Elastic wave field extrapolation: Redatuming of single-and multi-component seismic data. Vol. 2. Elsevier.

Wapenaar, K., 2017. Unified two-way wave equation and its symmetry properties. arXiv preprint arXiv:1801.07728.

Wapenaar, K., Fokkema, J., 2006. Green s function representations for seismic interferometry. Geophysics 71 (4), SI33–SI46.

Wapenaar, K., Thorbecke, J., van der Neut, J., Broggini, F., Slob, E., Snieder, R., 2014. Green's function retrieval from reflection data, in absence of a receiver at the virtual source position. The Journal of the Acoustical Society of America 135 (5), 2847–2861.

Weglein, A., Boyse, W., Anderson, J., 1981. Obtaining three-dimensional velocity information directly from reflection seismic data: An inverse scattering formalism. Geophysics 46 (8), 1116–1120.

Weglein, A. B., Araújo, F. V., Carvalho, P. M., Stolt, R. H., Matson, K. H., Coates, R. T., Corrigan, D., Foster, D. J., Shaw, S. A., Zhang, H., 2003. Inverse scattering series and seismic exploration. Inverse problems 19 (6), R27.

Weglein, A. B., Gasparotto, F. A., Carvalho, P. M., Stolt, R. H., 1997. An inverse-scattering series method for attenuating multiples in seismic reflection data. Geophysics 62 (6), 1975–1989.

Williams, S., Waterman, A., Patterson, D., 2009. Roofline: an insightful visual performance model for multicore architectures. Communications of the ACM 52 (4), 65–76.

Appendix

A Derivation of the acoustic Kirchhoff-Helmholtz integral from the acoustic reciprocity theorem of correlation type

In the space-frequency domain, the acoustic pressure $\hat{p}(\mathbf{x}, \omega)$ and particle velocities $\hat{v}_i(\mathbf{x}, \omega)$ defined in a lossless arbitrary inhomogenous acoustic medium obey the stress-strain relationship:

$$j\omega\kappa\hat{p} + \partial_i\hat{v}_i \tag{A.1}$$

and the equations of motion:

$$j\omega\rho\hat{v}_i + \partial_i\hat{p} = \hat{f}_i \tag{A.2}$$

The source terms $\hat{q}$ and $\hat{f}_i$ represent sources of volume injection rate and external force, respectively. The medium parameters $\kappa(\mathbf{x})$ and $\rho(\mathbf{x})$ are adiabatic incompressibility and volumetric density. Notation of frequency dependence is consistently suppressed in this section.

Consider two wavefield states A and B, both defined in the same medium and each obeying the acoustic stress-strain relationship and equations of motion. The reciprocity theorem of correlation type reads (Wapenaar and Fokkema, 2006):

$$\int_{\mathbb{D}} \{\hat{p}_A^*\hat{q}_B + \hat{v}_{i,A}^*\hat{f}_{i,B} + \hat{q}_A^*\hat{p}_B + \hat{f}_{i,A}^*\hat{v}_{i,B}\}d^3\mathbf{x} = \\ \oint_{\partial\mathbb{D}} \{\hat{p}_A^*\hat{v}_{i,B} + \hat{v}_{i,A}^*\hat{p}_B\}n_i d^2\mathbf{x} \tag{A.3}$$

It relates the interaction of wavefield components across the two states at the boundary $\partial\mathbb{D}$ to the interaction of the wavefield components and the source distributions throughout the domain $\mathbb{D}$. When states A and B coincide, the boundary integral term is linearly proportional to the acoustic power flux[1] through the surface $\partial\mathbb{D}$ with normal vector n_i (Wapenaar, 2017). Given two identical states, the correlation type reciprocity theorem then describes that the generation of power by the sources within $\mathbb{D}$ must be balanced by the power flux through the boundary $\partial\mathbb{D}$. Equation (A.3) is exact under the assumption of non-evanescent linearized wave motion in a lossless acoustic medium.

From hereon no state will be considered in which the source of external force is present. From the equations of motion (A.2) in the acoustic medium, the particle velocity can then be written as:

$$\hat{v}_i = \frac{-1}{j\omega\rho}\partial_i\hat{p} \tag{A.4}$$

[1] Simply scale the integral with a factor $\frac{1}{4}$ to obtain the power flux through the surface.

Inserting (A.4) into (A.1), yields the following wave equation:

$$\frac{\omega}{c^2\rho}\hat{p} + \partial_i(\frac{1}{\rho}\partial_i\hat{p}) = -j\omega\hat{q}(\mathbf{x}, \mathbf{x}') \tag{A.5}$$

The quantity $\hat{p}$ represents, through the stress-strain relationship and the definition of $\hat{q}$, the acoustic pressure due to a source of volume injection rate. The propagation velocity is $c(\mathbf{x}) = \{\kappa(\mathbf{x})\rho(\mathbf{x})\}^{-\frac{1}{2}}$.

Rather than the source-term on the right-handside defined by $-j\omega\hat{q}$, we search for a slightly alternative representation. Define $\mathcal{P} = \frac{\hat{p}}{j\omega}$ and $\mathcal{V}_i = \frac{\hat{v}_i}{j\omega}$. In terms of the modified wavefield quantities, the acoustic equations of motion and stress-strain relationship now read:

$$\mathcal{V}_i = \frac{-1}{j\omega\rho}\partial_i\mathcal{P} \tag{A.6}$$

$$j\omega\kappa\mathcal{P} + \partial_i\mathcal{V}_i = \frac{\hat{q}(\mathbf{x}, \mathbf{x}')}{j\omega} =: \hat{\imath}_V(\mathbf{x}, \mathbf{x}') \tag{A.7}$$

Hence, $\mathcal{P}$ represents the acoustic pressure due to a source of volume injection, as opposed to volume injection rate. The structure of the resulting wave-equation is the same as (A.5), but due to a different source function:

$$\frac{\omega^2}{c^2\rho}\mathcal{P} + \partial_i(\frac{1}{\rho}\partial_i\mathcal{P}) = -\hat{q}(\mathbf{x}, \mathbf{x}') \tag{A.8}$$

The corresponding Green's function is denoted $\mathcal{G}(\mathbf{x}, \mathbf{x}')$. Idem, we define a further modified wavefield P that satisfies the constant-density Helmholtz wave equation, valid in a constant density medium:

$$\frac{\omega^2}{c^2}P + \partial_i\partial_i P = -\hat{q}(\mathbf{x}, \mathbf{x}') \tag{A.9}$$

By a similar exercise as for the pressure field $\mathcal{P}$ one can show that, physically, P represents the acoustic pressure due to a source of buyoancy injection[2]. The corresponding Green's function is denoted $G(\mathbf{x}, \mathbf{x}')$.

The following quantities will be used in the correlation type reciprocity theorem (A.3). For state A, we use the Green's function, $\mathcal{G}(\mathbf{x}, \mathbf{x}_A)$ in terms of acoustic pressure due to an impulsive source of volume injection, $\hat{\imath}_V(\mathbf{x}, \mathbf{x}_A) = \frac{\hat{q}(\mathbf{x}, \mathbf{x}_A)}{j\omega} = \frac{\delta(\mathbf{x}-\mathbf{x}_A)}{j\omega}$, located at $\mathbf{x}_A$ inside $\mathbb{D}$. For state B, consider the physical wavefield in terms of acoustic pressure $\mathcal{P}(\mathbf{x}, \mathbf{x}_B)$ due to a source of volume injection located at position $\mathbf{x}_B$ outside $\mathbb{D}$.

[2] Although in a constant density medium this represents but an arbitrary scaling of the wavefield or the source function.

Inserting for states A and B, the correlation type reciprocity theorem gives:

$$\int_{\mathbb{D}} \{ (\frac{\delta(\mathbf{x},\mathbf{x}_A)}{j\omega})^* \mathcal{P}(\mathbf{x},\mathbf{x}_B) \} d^3\mathbf{x} = \oint_{\partial\mathbb{D}} \frac{-1}{j\omega\rho} \Big\{ \mathcal{G}^*(\mathbf{x},\mathbf{x}_A)\partial_i \mathcal{P}(\mathbf{x},\mathbf{x}_B) \\ -(\partial_i \mathcal{G}^*(\mathbf{x},\mathbf{x}_A))\mathcal{P}(\mathbf{x},\mathbf{x}_B) \Big\} n_i d^2\mathbf{x} \tag{A.10}$$

$$\mathcal{P}(\mathbf{x}_A,\mathbf{x}_B) = \oint_{\partial\mathbb{D}} \frac{1}{\rho} \{ \mathcal{G}^*(\mathbf{x},\mathbf{x}_A)\partial_i \mathcal{P}(\mathbf{x},\mathbf{x}_B) - (\partial_i \mathcal{G}^*(\mathbf{x},\mathbf{x}_A))\mathcal{P}(\mathbf{x},\mathbf{x}_B) \} n_i d^2\mathbf{x} \tag{A.11}$$

Equation (A.11) is the Kirchhoff-Helmholtz integral for an acoustic pressure field due to a source of volume injection. Idem, for wavefield states $G(\mathbf{x},\mathbf{x}_A)$ and $P(\mathbf{x},\mathbf{x}_B)$ satisfying the constant-density Helmholtz-equation, equation (A.9), the slightly simplified Kirchhoff-Helmholtz integral reads:

$$P(\mathbf{x}_A,\mathbf{x}_B) = \oint_{\partial\mathbb{D}} \{ G^*(\mathbf{x},\mathbf{x}_A)\partial_i P(\mathbf{x},\mathbf{x}_B) - (\partial_i G^*(\mathbf{x},\mathbf{x}_A))P(\mathbf{x},\mathbf{x}_B) \} n_i d^2\mathbf{x} \tag{A.12}$$

Representations theorems derived from the (unified) reciprocity theorems of correlation type are extensively used for inverse wavefield extrapolation, i.e. extrapolating a wavefield recorded at the boundary backwards in time and into the medium. The Kirchhoff-Helmholtz integrals simply represent specific variants expressed in terms of scalar, acoustic fields.

B The regularizing effect of Conjugate Gradient iterations

Prelude: A review of Singular Value decomposition

Consider a matrix $\mathbf{A} \in \mathbb{R}^{M\times N}$ and for simplicity assume $M \geq N$. Then, the Singular Value Decomposition (SVD) of $\mathbf{A}$ is of the form;

$$\mathbf{A} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T = \sum_{i=1}^{N} \mathbf{u}_i \sigma_i \mathbf{v}_i^T \tag{B.1}$$

Where $\mathbf{U} = (\mathbf{u}_1, \cdots, \mathbf{u}_N) \in \mathbb{R}^{M\times N}$ and $\mathbf{V} = (\mathbf{v}_1, \cdots, \mathbf{v}_N) \in \mathbb{R}^{N\times N}$ are matrices with orthonormal columns s.t. $\mathbf{U}^T\mathbf{U} = \mathbf{I}_N = \mathbf{V}^T\mathbf{V}$. $\mathbf{\Sigma}$ is a diagonal matrix with non-negative entries σ_i appearing in non-increasing order s.t.:

$$\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_N \geq 0 \tag{B.2}$$

The scalars σ_i are termed the singular values of $\mathbf{A}$. They are defined as the stationary values of the expression $||\mathbf{Ax}||_2/||\mathbf{x}||_2$. The vectors $\mathbf{u}_i$ and $\mathbf{v}_i$ are the left and right singular vectors of $\mathbf{A}$, respectively, and form two sets of orthonormal basis vectors. In the case that $M < N$, apply the SVD to $\mathbf{A}^T$ and interchange $\mathbf{U}$ and $\mathbf{V}$.

The singular value decomposition of $\mathbf{A}$ is strongly related to the eigenvalue decomposition of the symmetric matrices $\mathbf{A}^T\mathbf{A}$ and $\mathbf{A}\mathbf{A}^T$. Suppose the former symmetric matrix can be expressed in an eigenvalue decomposition of the following:

$$\mathbf{A}^T\mathbf{A} = \mathbf{E}\mathbf{\Lambda}\mathbf{E}^T \tag{B.3}$$

Via the SVD Decomposition by equation (B.1), one can write:

$$\mathbf{A}^T\mathbf{A} = \mathbf{V}\boldsymbol{\Sigma}^T\mathbf{U}^T\mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{V}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{V}^T = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^T \tag{B.4}$$

Idem, suppose $\mathbf{A}\mathbf{A}^T$ can be expressed in an eigenvalue decomposition according to:

$$\mathbf{A}\mathbf{A}^T = \widehat{\mathbf{E}}\widehat{\boldsymbol{\Lambda}}\widehat{\mathbf{E}}^T \tag{B.5}$$

Then, we perform the same exercise using the SVD decomposition:

$$\mathbf{A}\mathbf{A}^T = \mathbf{U}\boldsymbol{\Sigma}^T\mathbf{V}^T\mathbf{V}\boldsymbol{\Sigma}\mathbf{U}^T = \mathbf{U}\boldsymbol{\Sigma}^T\boldsymbol{\Sigma}\mathbf{U}^T = \widehat{\mathbf{E}}\widehat{\boldsymbol{\Lambda}}\widehat{\mathbf{E}}^T \tag{B.6}$$

From equations (B.4) and (B.6) we can therefore state the following regarding the singular value decomposition $\mathbf{A} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T$:

- The right-singular vectors, i.e. the columns of $\mathbf{V}$, are the eigenvectors of $\mathbf{A}^T\mathbf{A}$.
- The left-singular vectors, i.e. the columns of $\mathbf{U}$, are the eigenvectors of $\mathbf{A}\mathbf{A}^T$.
- The singular values, i.e. entries of $\boldsymbol{\Sigma}$, are the square roots of the eigenvalues of $\mathbf{A}^T\mathbf{A}$ or $\mathbf{A}\mathbf{A}^T$.

SVD Decomposition of inverse matrices
For a square matrix $\mathbf{A} \in \mathbb{R}^{N\times N}$ of full rank, i.e. with all $\sigma_i > 0$, the singular value decomposition of its inverse is given by:

$$\mathbf{A}^{-1} = \mathbf{V}^{-T}\boldsymbol{\Sigma}^{-1}\mathbf{U}^{-1} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{U}^T = \sum_{i=1}^{N} \mathbf{v}_i \frac{1}{\sigma_i}\mathbf{u}_i^T \tag{B.7}$$

However, if the matrix is rectangular, not of full rank or both, then the solution to $\mathbf{A}\mathbf{x} = \mathbf{b}$ is given through application of its *pseudoinverse*[3] $\mathbf{A}^+$. The SVD of the pseudoinverse is given by:

$$\mathbf{A}^+ = \mathbf{V}\boldsymbol{\Sigma}^+\mathbf{U}^T = \sum_{i=1}^{\text{rank}(\mathbf{A})} \mathbf{v}_i \frac{1}{\sigma_i}\mathbf{u}_i^T \tag{B.8}$$

In relation to linear least squares inverse problems of the form $\mathbf{x} = \arg\min_{\mathbf{x}} ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2$, the least squares solution $\mathbf{x}_{LS}$ is given by:

$$\mathbf{x}_{LS} = \mathbf{A}^+\mathbf{b} = \sum_{i=1}^{\text{rank}(\mathbf{A})} \mathbf{v}_i \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i} \tag{B.9}$$

Division by the small singular values in the expression for $\mathbf{x}_{LS}$ amplifies high-frequency components in the data $\mathbf{b}$. Furthermore, the sensitivity of the least squares solution $\mathbf{x}_{LS}$ to

[3] Commonly it is also termed the Moore-Penrose generalized inverse.

perturbations in either $\mathbf{b}$ and/or $\mathbf{A}$ can be measured by the condition number of the matrix $\mathbf{A}$:

$$\text{Cond}(\mathbf{A}) := \frac{||\mathbf{A}||_2}{||\mathbf{A}^+||_2} = \frac{\sigma_1}{\sigma_{\text{rank}(\mathbf{A})}} \tag{B.10}$$

Very small relative singular values will evidently make the least squares solution very sensitive to perturbations. In the context of iterative matrix inversion algorithms, e.g. the CGLS algorithm, high condition numbers deteriorate the convergence rate due to an increased sensitivity to numerical round-off errors (Björck et al., 1998).

Rank deficient systems of equations

Consider the the theoretical situation where $\mathbf{A}$ and $\mathbf{b}$ are free of perturbation and all operations are performed on infinite precision hardware. Treating rank deficient problems can then be done easily by simply ignoring the components of the SVD where $\sigma = 0$ and computing the solution by means of the pseudoinverse, as in equation (B.9).

In practice, however, $\mathbf{A}$ is never *exactly* rank deficient. Rather, it may be numerically rank deficient such that in a canonical sense Rank$(\mathbf{A}) = N$, but one or more of the last σ_i are very small. The influence of such small singular values significantly perturb the solution. As a demonstration, the squared ℓ_2 norm of the least-squares solution $\mathbf{x}_{LS}$ is given by:

$$||\mathbf{x}_{LS}||_2^2 = \sum_{i=1}^{N} \left[\mathbf{v}_i \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\right]^T \left[\mathbf{v}_i \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\right] = \sum_{i=1}^{N} \left(\frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i}\right)^2 \tag{B.11}$$

Therefore, the solution norm will be very large, and the solution $\mathbf{x}_{LS}$ as a whole may be dominated by the SVD components corresponding to numerically rank deficient singular values.

One way to deal with rank deficiency, exact or numerical, is to project the matrix $\mathbf{A}$ onto a rank-k matrix $\mathbf{A}_k$ where the latter has its small, but nonzero, singular values $\sigma_{k+1}, \cdots, \sigma_N$ replaced with exact zeros. The solution to the corresponding least squares inverse problem reads:

$$\mathbf{x}_{LS,k} = \mathbf{A}_k^+\mathbf{b} = \sum_{i=1}^{k} \mathbf{v}_i \frac{\mathbf{u}_i^T\mathbf{b}}{\sigma_i} \tag{B.12}$$

The solution $\mathbf{x}_{LS,k}$ is referred to as the truncated SVD solution, abbreviated TSVD. Only the k first SVD components are allowed to contribute to the solution.

Alternatively, rank deficiency can be treated via regularization of the inverse problem. The effect of this will be demonstrated for simple, zero-th order Tikhonov regularization. The regularized inverse problem reads:

$$\mathbf{x} = \arg\min_{\mathbf{x}} \left\{||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2 + \mu^2||\mathbf{x}||_2^2\right\} \tag{B.13}$$

The regularized solution can be expressed as a filtered version of the SVD components of the least squares solution (Hansen (2005)):

$$\mathbf{x}_{LS,reg} = \sum_{i=1}^{N} \mathbf{v}_i f_i \frac{\mathbf{u}_i^T \mathbf{b}}{\sigma_i} \tag{B.14}$$

For zero-th order Tikhonov regularization the values of the implicitly introduced filter components f_i read:

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \mu^2} \tag{B.15}$$

Hence, for $\sigma_i^2 >> \mu^2$ the filter coefficients behave like $f_i \approx 1$, hence the filter is all-pass in this range of singular values. On the other end of the spectrum, situations may occur in which $\sigma_i^2 << \mu^2$. Then, the behaviour of the filter coefficents approach $f_i \approx \frac{\sigma_i^2}{\mu^2} \approx 0$.

The effect of Tikhonov regularization is to smoothly filter the singular value spectrum of the solution. The amount of contributing singular values are inversely proportional to the value of the regularization parameter.

Iterative regularization methods

In the context of regularization through application of iterative methods, each iteration vector $\mathbf{x}^{(k)}$ can be considered a regularized solution, where the iteration number k plays the role of the regularization parameter. For this to occur we must require iteration schemes that when applied to discrete, ill-posed problems initially pick up the SVD components that correspond to the largest singular values. One attempts to avoid the phenomenon of semiconvergence by stopping the iterative scheme (long) before convergence to a minimal residual solution. The phenomenon of semiconvergence entails the iteration history where the vector $\mathbf{x}^{(k)}$ converges towards the exact solution $\mathbf{x}_{\text{exact}}$ for small k but later on diverges and approaches the non-optimal $\mathbf{x}_{LS}$ least-squares solution as defined by (B.9) for large k. It should be noted that due to the ill-posedness of the inverse problem, the iterative solution may diverge from the true solution while concurrently the data residuals and residuals of the normal equations may convergence.

In order to make use of the regularizing properties of iterative methods, the regularized objective function $\varphi = ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2 + \mu^2 ||\mathbf{W}_x \mathbf{x}||_2^2$ is transformed into its standard form $\varphi = ||\mathbf{b} - \mathbf{A}\mathbf{W}_x^{-1}\tilde{\mathbf{x}}||_2^2 + \mu^2||\mathbf{x}||_2^2$ through the application of the preconditioner $\mathbf{W}_x^{-1}$. In order to let the iteration numbers play the role of the regularizing parameter, the parameter μ^2 must be set to zero. The preconditioner used here improves a part of the singular value spectrum of $\mathbf{A}$, in particular those singular values that contribute most to the regularized solution. Hence, this is in contrast to ”normal” preconditioning of an SPD system of linear equations $\mathbf{A}\mathbf{x} = \mathbf{b}$, whose aim is to choose a preconditioner $\mathbf{M}$ s.t. Cond$(\mathbf{MA}) <$ Cond$(\mathbf{A})$ in order to aid convergence.

Regularizing Conjugate Gradient Iterations

For conjugate gradient iterates $\mathbf{x}^{(k)}$ with residual vector $\mathbf{r}^{(k)} = \mathbf{b} - \mathbf{A}\mathbf{x}^{(k)}$, the residual of normal equations are given by $\mathbf{s}^{(k)} = \mathbf{A}^T\mathbf{b} - \mathbf{A}^T\mathbf{A}\mathbf{x}^{(k)} = \mathbf{A}^T\mathbf{r}^{(k)}$. Furthermore, they are mutually orthogonal, i.e.:

$$\mathbf{s}^{(k)} \perp \mathbf{s}^{(j)} \quad j \neq k \tag{B.16}$$

This orthogonality implies that if the initial solution is the zero vector, $\mathbf{x}^{(0)} = 0$, then the norm of the solution $||\mathbf{x}^{(k)}||_2^2$ increases monotonically with k (Hestenes and Stiefel, 1952). From the monotonic behaviour of the solution norm, one can intuitively sense a relationship between iteration number and which SVD components contribute to the solution, see e.g. (B.11) for reference.

Hansen (2005) states that the Conjugate Gradient method often produces iterative solutions in which the large eigenvalues of $\mathbf{A}$ converge faster than the smaller components. For least squares inverse problems, the eigenvalues of $\mathbf{A}^T\mathbf{A}$ are simply σ_i^2, as per equation (B.4). Hence, the singular value components associated with the large σ_i tend to converge the fastest. If the CG algorithm is stopped long before convergence to the Least-Squares solution $\mathbf{x}_{LS} = \mathbf{A}^+\mathbf{b}$ sets in, it should be equivalent to a regularized, LS solution for a particular choice of μ^2.

The k-th Conjugate Gradient iterate vector can be written as (Bjorck, 1996):

$$\begin{aligned} \mathbf{x}^{(k)} &= \arg\min_{\mathbf{x}} \left\{ ||\mathbf{b} - \mathbf{A}\mathbf{x}||_2^2 \right\} \\ \text{subject to } \mathbf{x} &\in \mathcal{K}_k(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b}) := \text{span}\{\mathbf{A}^T\mathbf{b}, \cdots, (\mathbf{A}^T\mathbf{A})^{k-1}\mathbf{A}^T\mathbf{b}\} \end{aligned} \tag{B.17}$$

$\mathcal{K}_k(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$ is the *Krylov subspace* of the k-th iteration of the CG algorithm. Hansen (2005) states that in certain[4] applications the Krylov subspace $\mathcal{K}_k(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$ can be considered an approximation to the subspace spanned by the first k right singular vectors of the matrix $\mathbf{A}$. Therefore, the iterative solution $\mathbf{x}^{(k)}$ can be considered an approximation to the TSVD solution $\mathbf{x}_{LS,k}$ given in (B.12), if the CG algorithm is applied to the standard form least-squares problem, i.e. if it is right preconditioned with $\mathbf{W}_x^{-1}$ and $\mu^2 = 0$.

In a more formal sense, the SVD component filtering properties of the CG algorithm is controlled by Ritz values. We denote the matrix $\mathbf{T}^{(k)}$ as the representation of $\mathbf{A}^T\mathbf{A}$ projected onto $\mathcal{K}_k(\mathbf{A}^T\mathbf{A}, \mathbf{A}^T\mathbf{b})$. The eigenvalues of $\mathbf{T}^{(k)}$ are $\theta_1^{(k)}, \cdots, \theta_k^{(k)}$ are called the Ritz values of $\mathbf{A}^T\mathbf{A}$ at the k-th step and converge to the square of the singular values σ_i when $k \rightarrow \infty$ (Favati et al., 2014). The filter factors of the Conjugate Gradient algorithm at the k-th step have the form (Hansen, 2005):

$$f_i^{(k)} = 1 - \prod_{j=1}^{k} \left(1 - \frac{\sigma_j^2}{\theta_j^{(k)}}\right) \tag{B.18}$$

[4] Note that Hansen (2005) does not discuss in which situations this analysis is applicable.

In regularizing Conjugate Gradient iterations the optimal number of iterations k_{opt} is therefore determined by the convergence behaviour of the Ritz values. If the number of iterations is chosen too large, all of the Ritz values converge and the iterative solution approaches the unregularized least-squares solution:

$$\lim_{k\to\infty} \mathbf{x}^{(k)} = \mathbf{x}_{LS} \tag{B.19}$$

C Proof of Lemma 1

To prove, for any integrable functions $f(t)$ and $g(t)$:

$$\int_{-\infty}^{\infty} dt \quad f(t) \int_{-\infty}^{t-\epsilon} dt' \quad g(t') = \int_{-\infty}^{\infty} dt\, g(t) \int_{t+\epsilon}^{\infty} dt' f(t') \tag{C.1}$$

We define the Heaviside step function: $\Theta(x)$:

$$\Theta(x) = \begin{cases} 0, & \text{for } x < 0 \\ 1, & \text{for } x \geq 0 \end{cases} \tag{C.2}$$

Equation (C.1) can then be re-written as:

$$\int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{t-\epsilon} dt'\, g(t') = \int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{\infty} dt'\, g(t')\Theta[(t-\epsilon)-t']$$

Then by renaming: $t \to t'$, $t' \to t$:

$$\int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{t-\epsilon} dt'\, g(t') = \int_{-\infty}^{\infty} dt'\, f(t') \int_{-\infty}^{\infty} dt\, g(t)\Theta[(t'-\epsilon)-t]$$

$$\int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{t-\epsilon} dt'\, g(t') = \int_{-\infty}^{\infty} dt'\, f(t') \int_{-\infty}^{\infty} dt\, g(t)\Theta[t'-(t+\epsilon)]$$

$$\int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{t-\epsilon} dt'\, g(t') = \int_{-\infty}^{\infty} dt\, g(t) \int_{-\infty}^{\infty} dt'\, f(t')\Theta[(t'-(t+\epsilon)]$$

$$\int_{-\infty}^{\infty} dt\, f(t) \int_{-\infty}^{t-\epsilon} dt'\, g(t') = \int_{-\infty}^{\infty} dt\, g(t) \int_{t+\epsilon}^{\infty} dt'\, f(t') \qquad \blacksquare \tag{C.3}$$