SEISMIC DATA INTERPOLATION USING SPARSITY CONSTRAINED INVERSION

by

Lucas José Andrade de Almeida

© Copyright by Lucas José Andrade de Almeida, 2017

All Rights Reserved

A thesis submitted to the Faculty and the Board of Trustees of the Colorado School of Mines in partial fulfillment of the requirements for the degree of Master of Science degree (Geophysics).

Golden, Colorado

Date _____

Signed: _____
Lucas José Andrade de Almeida

Signed: _____
Dr. Paul C. Sava
Thesis Advisor

Golden, Colorado

Date _____

Signed: _____
Dr. Roel Snieder
Professor and Head
Department of Geophysics

ABSTRACT

Missing data reconstruction is an ongoing challenge in seismic processing for incomplete and irregular acquisition. This problem needs to be adequately addressed, especially because it negatively affects several important processing steps such as migration. While many methods have been developed to address this problem, most of the recent research on the subject focuses on transform domain approaches due to their low computational cost and independence of medium property estimation, such as velocity and density. Because of the underdetermined nature of the interpolation problem, transform domain techniques use sparsity constraints in order to obtain better solutions than the conventional least squares approach. Motivated by the compressive sensing framework, which proves that sparse signals can be recovered from a highly incomplete set of measurements, two approaches for sparsity constrained inversion are commonly used: the synthesis and analysis approaches. Although similarly posed, they present differences when the transform domain used in the inversion is redundant. Due to the high dimensionality and aliasing that occurs in the interpolation problem, such approaches are not able to handle large gaps between traces and display artifacts in the reconstruction due to the usage of the L_1 norm as a proxy for sparsity. Weighted algorithms that approximate the L_0 norm have been proposed as a way to mitigate such problems. In this thesis, I show the problems associated with the synthesis and analysis approaches when applied to seismic interpolation and propose an approach based on a reweighted L_1 algorithm that estimates weights recursively, i.e., from previous weighted steps. Experiments, carried out on 2D and 3D data using different undersampling schemes, show that the proposed approach is able to improve estimation inside large gaps, decrease artifacts and obtain better SNR when compared with conventional analysis and synthesis sparsity constrained inversion.

TABLE OF CONTENTS

ABSTRACT iii

LIST OF FIGURES vi

LIST OF ALGORITHMS x

ACKNOWLEDGMENTS xi

DEDICATION xiii

CHAPTER 1 INTRODUCTION 1

CHAPTER 2 SEISMIC INTERPOLATION USING SYNTHESIS AND ANALYSIS CONSTRAINTS 5

2.1 The underdetermined inverse problem 5

2.2 Sparsity constraint for underdetermined inverse problems 5

2.2.1 Conditions for sparse recovery 7

2.3 Sparsity in transform domains 8

2.4 Analysis and synthesis approaches 9

2.4.1 Two toy examples 10

2.4.2 Analysis vs. synthesis 14

2.5 The seismic interpolation as an inverse problem 15

2.5.1 Sparsity in seismic wavefields 16

2.5.2 The restriction matrix 18

2.6 Results 20

2.7 Discussion 26

CHAPTER 3 SEISMIC INTERPOLATION USING A REWEIGHTED ANALYSIS APPROACH 30

3.1 Prior information in sparsity constrained inversion 30

3.2 L_0 norm via reweighted L_1 inversion 30

3.2.1 Effect of weights on L_1 norm minimization 31

3.2.2 Algorithm for reweighted L_1 minimization 32

3.3 Effects of the parameter ϵ 34

3.4 An alternative solution 46

3.4.1 Modification of the L_1 gradient 46

3.4.2 Implications and examples 47

3.5 3D Examples 52

3.6 Discussion 54

CHAPTER 4 CONCLUSIONS 59

REFERENCES CITED 61

CHAPTER 5 APPENDIX - NESTA 65

LIST OF FIGURES

Figure 2.1 (a) Example of a sparse signal. Even though the signal dimension is 400 samples, the number of non-zero coefficients is only 30. 6

Figure 2.2 Compressive sensing toy problem: (a) Input signal, which is denoted as $\mathbf{x}$. (b) The measured data $\mathbf{y} = \mathbf{\Phi x}$. (c) Real part of the frequency spectrum of the sine in (a). (d) Reconstruction result, which shows perfect estimation from the data $\mathbf{y}$. 11

Figure 2.3 (a) Input signal, which is denoted as $\mathbf{x}$. (b) Forward Gabor transform of the signal in (a). 12

Figure 2.4 Estimated and true signals in red and blue, respectively: (a) In the physical domain as a result of the synthesis approach. (b) In the transform domain as a result of the synthesis approach. (c) In the physical domain as a result of the analysis approach. (d) In the transform domain as a result of the analysis approach. 13

Figure 2.5 Curvelets of different scales: (a), (c) In the space domain. (b), (d) In the frequency domain. Note that curvelets are very similar to seismic events, i.e, a band-limited wavefront. 17

Figure 2.6 (a) Signal containing three different frequencies. (b) Corresponding frequency spectrum. 18

Figure 2.7 Figures displaying different undersampling schemes. The solid line indicates the original signal and the dots indicate the selected samples. Signals in time and corresponding spectra for (a), (b) Uniform sampling, (c), (d) Random sampling. 19

Figure 2.8 (a) Shot gather from the Sigsbee 2A model. (b) Corresponding f-k spectrum. 22

Figure 2.9 (a) Shot gather after uniform undersampling. (b) Corresponding f-k spectrum. 23

Figure 2.10 Uniform undersampling: (a), (b) Synthesis and analysis solutions. SNR of 13.29 dB and 14.83 dB, respectively. (c), (d) Corresponding f-k spectra. 24

Figure 2.11 Detail from the solutions in Figure 2.10 (a) and (b): (a) Synthesis solution. (b) Analysis solution. Notice the artifacts present in (a) at coordinates $x = 4.5$km and $t = 6.35$s. 25

Figure 2.12 (a) Shot gather after random undersampling. (b) Corresponding f-k spectrum. 27

Figure 2.13 Random undersampling: (a), (b) Synthesis and analysis solutions. SNR of 16.24 dB and 17.87 dB, respectively. (c), (d) Corresponding f-k spectra. 28

Figure 2.14 Detail from the solutions in Figure 2.13 (a) and (b): (a) Synthesis solution. (b) Analysis solution. Notice the artifacts present in (a) at coordinates $x = 4.5$km and $t = 6.35$s. 29

Figure 3.1 Geometric solutions of a weighted two-dimensional inversion example: (a) Unweighted solution. (b) Arbitrarily large weight in x_1. (c) Arbitrarily large weight in x_2. The blue and orange lines indicate the constraint and the L_1 ball centered at the origin, respectively. 33

Figure 3.2 Illustration of the different norms involved in this chapter. The lines in blue, red and yellow represent the L_1 norm, the L_0 norm and the function $f(x) = \sum_{i=1}^{n} log(|\mathbf{x}_i| + \epsilon)$, plotted for $\epsilon = 10^{-2}$. Note that the proposed function approximates the L_0 norm better than the L_1 norm for large values. 35

Figure 3.3 Binned weight matrices for the true model in Figure 2.8(a) for several values of ϵ: (a) $\epsilon = 10^2$. (b) $\epsilon = 10$. (c) $\epsilon = 1$. (d) $\epsilon = 10^{-2}$. (e) $\epsilon = 10^{-6}$. . 37

Figure 3.5 Toy example using Algorithm 1: (a) $\epsilon = 10^2$ (b) $\epsilon = 1$. (c) $\epsilon = 10^{-2}$. (d) $\epsilon = 10^{-6}$. The true model and the estimated solution are displayed in blue and orange, respectively. 39

Figure 3.6 Interpolation results for the minimization problem using Algorithm 1 and equation 3.5: (a) $\epsilon = 10^2$, SNR of 17.11dB. (b) $\epsilon = 1$, SNR of 18.96. (c) $\epsilon = 10^{-2}$, SNR of 18.62 dB. (d) $\epsilon = 10^{-6}$, SNR of 18.54 dB. 40

Figure 3.7 Binned weight matrix for $\epsilon = 10^{-6}$: (a) first and (b) second reweighted iterations. (c), (d) Detail of the top part of (a) and (b), respectively. Note the increase in number of high amplitude weights, which correspond to small values in the coefficients, in (d). 41

Figure 3.8 Binned weights using equation 3.5: (a) True model in Figure 2.8(a). (b) Estimated model in Figure 3.6(d). (c), (d) Detail of the top part of (a) and (b), respectively. Note the increase in number of high amplitude weights, which correspond to small values in the coefficients, in (d). . . . 43

Figure 3.9 Results of Algorithm 1 for different number of iterations and $\epsilon = 1$: (a) 25 iterations, 19.76 dB. (b) 50 iterations, SNR fof 19.78 dB. (c) 75 iterations, SNR of 19.78 dB. (d) 100 iterations, SNR of 19.78 dB. 44

Figure 3.10 Binned weights using equation 3.5: (a) Figure 3.9(a). (b) Detail of the top part of (a). Note that the number of high amplitude weights, which correspond to the small values in the coefficients, decrease when compared to Figure 3.8(b). 45

Figure 3.11 Reweighted iterations for the toy problem using equation 3.9: (a) First iteration, (b) second iteration, (c) third iteration, (d) fourth iteration, (e) fifth iteration. The true model and the estimated solution are displayed in blue and orange, respectively. 49

Figure 3.12 Reweighted iterations for the toy problem using equation 3.10: (a) First iteration, (b) second iteration, (c) third iteration, (d) fourth iteration, (e) fifth iteration. The true model and the estimated solution are displayed in blue and orange, respectively. 50

Figure 3.13 (a) Solution using the modified gradient in equation 3.10, SNR of 18.70 dB. 51

Figure 3.14 Binned weight vectors for minimization procedure that estimates Figure 3.13: (a) first and (b) second reweighted iterations. Notice how the weights are close to each other in the second reweighted iteration. The colors indicate the density of points inside a given bin. 51

Figure 3.15 Results of Algorithm 1 using Figure 3.13 as starting model and $\epsilon = 1$: (a) 25 iterations, 19.80 dB. (b) 50 iterations, SNR fof 19.85 dB. (c) 75 iterations, SNR of 19.97 dB. (d) 100 iterations, SNR of 19.98 dB. 53

Figure 3.16 (a) Convergence curves for the experiments in Figure 3.9, in red, and Figure 3.15, in blue. The normalized difference is calculated as $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$, where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are the true model and the estimated solution, respectively. 54

Figure 3.17 100 shots from the 2D Gulf of Mexico dataset. 55

Figure 3.18 (a) Detail illustrating uniformly undersampled data. (b), (c) Synthesis and analysis solutions, SNR of 6.23 dB and 6.27 dB, respectively. (d) Reweighted solution with the inverse, SNR of 7.15 dB. 56

Figure 3.19 (a) Detail illustrating randomly undersampled data. (b), (c) Synthesis and analysis solutions, SNR of 17.27 dB and 17.21 dB, respectively. (d) Reweighted solution with the inverse, SNR of 18.94 dB. 57

LIST OF ALGORITHMS

Algorithm 1 Reweighted analysis L_1 procedure 32

Algorithm 2 General NESTA Solver 65

Algorithm 3 NESTA Solver for Orthogonal Projections 66

ACKNOWLEDGMENTS

First, I would like to thank my advisor Paul Sava for granting me the opportunity of pursuing this degree. Even though my research interests were not very close to his expertise, he was always patient and ready to point me in the directions he believed were right. He also helped me a lot with my difficulties, both in research and life, and for that I will always be grateful. I am sure his teachings will be always valuable in my life.

My research would not be that interesting without the help of the remaining members of my thesis commitee, Paul Constantine and Michael Wakin. Paul was always ready to share his wisdom about inverse problems and dimensionality reduction with me, which helped me greatly in my research. Mike is an expert in my area of research, and I learned a lot from him about Compressive Sensing during our meetings. I hope he learned something from me, too.

Life in CWP would not be as good if I didn't have so many interesting colleagues to talk to. I especially thank my iTeam friends, Daniel Rocha, Alicia Johnson, Ivan Lim, Iga Pawelec, Thomas Rapstine, Esteban Diaz and Yuting Duan for their company and assistance. We shared too many seminar hours, but you guys made it fun. I also thank Nishant Kamath, Aaron Prunty and Elias Arias for every weird conversation and discussion we had during work breaks (be it about books, classes or psychometrics).

While CWP is recognized mostly because of its students work, none of our work would have been possible without the best faculty and staff we could possibly have. I thank Dawn Umpleby, Pam Kraus and Shingo Ishida for their commitment in making our life at CSM easier by dealing with all the bureaucratic procedures we wish would not exist. Diane Witters taught me a lot about how to be an effective writer, and proved to be a good friend when I was away from home in a long time. Roel Snieder and Ilya Tsvankin are two incredible professors. Roel's class was very important to me as a scientist, while Ilya taught me everything I know

about seismology and anisotropy.

Finally, I would like to thank my family for the support during my time at CSM, both with research and in my life. Every time I thought I couldn't do it, they were always there to push me and keep me straight on my goals. Thank you for beIieving in me more than I do. I am sure that wherever I end up after the completion of this master's degree, I will be in very good company with them by my side.

To my grandfather, Ernesto.

CHAPTER 1

INTRODUCTION

Seismic acquisition seeks to sample seismic wavefields densely and regularly in every spatial direction and time, with the intent of obtaining signals that adequately represent data observed at the surface. Regular sampling has important consequences for many applications, such as reverse-time migration and multiple removal (Trad, 2009). However, acquisition costs and field obstacles can easily make the acquisition both irregular and sparse. The purpose of missing trace reconstruction is to fill the data gaps and to resample the data as accurately as possible.

Some of the most traditional approaches for missing trace reconstruction are based on wave-equation physics. In general, these approaches use operators built according to a wave equation to map the seismic wavefields to a physical domain (such as a migrated image). This modeled domain is subsequently transformed back to the data space and produces the data that would have been acquired with regular acquisition (Stolt, 2002; Ronen, 1987; Clapp et al., 2006; Trad, 2003). While wave-equation approaches to seismic interpolation are important, they make assumptions about the knowledge of the velocity field of the medium and use modelling as the interpolation tool, which may impose large computational costs.

A different group of techniques for seismic interpolation is based on signal processing and require no knowledge of the properties of the medium. For example, the usage of prediction error filters to reconstruct seismic data has been proposed in the $t-x$ domain, where one can predict the dips of the data and reconstruct it (Claerbout, 2014; Fomel, 2002). Prediction error filters were also proposed in the $f-x$ domain by Spitz (1991) and further improved by Porsani (1999). In this domain, the low frequency component of the data is used as prior information to estimate the high frequency components. More recently, the assumption of sparsity for seismic signals in different transform domains to estimate the missing traces has

been proposed by several authors, such as Hennenfent and Herrmann (2008) and Naghizadeh and Sacchi (2010). This approach assumes that the analyzed wavefield is sparse in some domain and involves the minimization of a convex function with an L_1 norm, which is known to promote sparsity. Examples of transforms that are used in this setting are the Fourier, curvelet (Candes et al., 2006), seislet (Fomel and Liu, 2010) and data-driven transforms (Yu et al., 2015).

Transform domain approaches are closely related to the compressive sensing problem (Candès et al., 2006), which provides theoretical guarantees of exact recovery for underdetermined inverse problems where the estimated solution is sparse. In the common compressive sensing formulation, one tries to find the sparsest representation that can best describe the signal to be estimated in a certain transform domain dependent on the signal. This formulation is known as the **synthesis approach** to the L_1 minimization problem and it has been widely used for several geophysical problems, such as full-waveform inversion (Li et al., 2012), deblending (Wason et al., 2011), salt body detection (Ramirez et al., 2016), microseismic source retrieval (Rodriguez and Sacchi, 2014) and missing trace reconstruction (Hennenfent and Herrmann, 2008), often producing superior results compared to traditional techniques in the field.

When the transform domain is represented by a frame (Mallat, 2008), several representations of the same signal are allowed in the transform domain. In these cases, the **analysis approach** provides a different way of solving the L_1 minimization problem with the constraint that the forward transform of the signal to be estimated is sparse. The analysis approach and its geometry have been studied theoretically from the perspective of compressive sensing (Candes et al., 2011; Nam et al., 2013). Empirically, the analysis approach has been found to be superior to the synthesis approach when used to recover specific types of signals (Elad et al., 2007). Improvements over the synthesis approach were observed especially for the denoising problem on piecewise smooth images (Selesnick and Figueiredo, 2009; Elad et al., 2007). However, the differences between the analysis and synthesis approaches

are highly dependent on the application, dataset and transform domain, and a definitive conclusion regarding which approach is better remains elusive (Elad et al., 2007). In geophysics, the analysis approach has been used sparingly to solve problems such as denoising with data-driven frames (Chen et al., 2016), multiple removal and interpolation (Yang and Fomel, 2015).

Although both approaches have been used in several geophysical problems, they often present issues, such as dimming of amplitudes, when dealing with large gaps in seismic data and also may lead to artifacts in the reconstruction (Lin and Herrmann, 2013; Hennenfent and Herrmann, 2008; De Almeida et al., 2016). Weighted algorithms that use prior information in the frequency domain about the signal to be estimated, such as using the location of the non-zero coefficients of the low frequency slices to interpolate high frequency ones, were proposed in literature and show some success in dealing with such issues when the prior information is accurate (Liu and Sacchi, 2004), (Mansour et al., 2013). However, a high number of frequency slices may be needed to accurately estimate the location of the relevant coefficients from the frequency slices, which can quickly increase the computational cost of the method.

A different realm of reweighted algorithms attempt to approximate the L_0 norm solution via weighted matrices with the objective of enhancing sparsity of the solution. In this case, the assumption is that enhancing sparsity helps the inverse problem to correctly estimate large transform coefficients as opposed to small ones. In this work, I devise an approach to deal with the issues of the synthesis and analysis approaches based on the reweighted L_1 norm technique proposed by Candes et al. (2008) that approximates the L_0 norm. The approach uses a modified gradient to produce solutions that better represents seismic data inside large gaps and decrease artifacts in the estimated solution. This approach does not use prior information and uses weights derived from the previous weighted iterations. I illustrate this approach along with the synthesis and analysis approaches in a complex synthetic shot gather from the Sigsbee 2A model and 3D field data. The results show that the proposed

method helps find models with less artifacts and higher signal to noise ratio than conventional synthesis and analysis approaches.

CHAPTER 2

SEISMIC INTERPOLATION USING SYNTHESIS AND ANALYSIS CONSTRAINTS

2.1 The underdetermined inverse problem

The seismic interpolation problem aims at filling the gaps in seismic records such that the recovered data are plausible from a seismic point of view and agree with the recorded observations. In this problem, the number of data samples available is smaller than the number of model parameters to be recovered. Thus, the interpolation of seismic data can be understood as the equation

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x}, \tag{2.1}$$

where $\mathbf{\Phi}$ is a linear operator that maps a high dimensional vector $\mathbf{x} \in \Re^n$ to a low dimensional vector $\mathbf{y} \in \Re^m$. In general, this problem is related to the embedding of a signal, which belongs to a high dimensional subspace, into a lower dimensional subspace. Several problems currently pursued by the signal processing community, such as denoising and edge detection (Selesnick and Figueiredo, 2009), can be formulated in a way that resembles equation 2.1. In the seismic interpolation case, $\mathbf{y}$ represents the data with missing gaps, while $\mathbf{x}$ represents the data without missing gaps to be recovered.

In general, the problem of obtaining a high dimensional signal from a low dimensional representation has infinite solutions. Solving equation 2.1 in a least-squares sense yields only an approximate solution, which may differ significantly from the original signal due to the non-uniqueness of the problem.

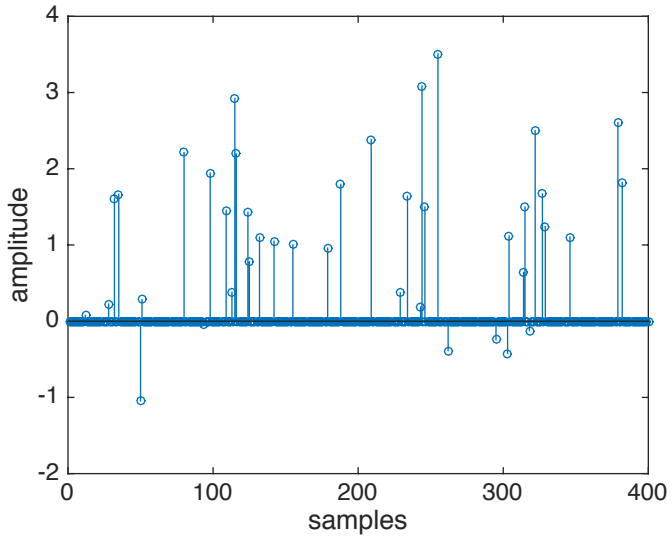
2.2 Sparsity constraint for underdetermined inverse problems

Equation 2.1 requires some sort of regularization in order to reduce the amount of possible solutions based on desirable characteristics (such as smoothness, blockiness or even petrophysical plausibility). A popular way to constrain the aforementioned problem is to

assume that the signal to be recovered is sparse, which means that even though the signal is high dimensional, it contains a small number of non-zero elements. Formally, sparsity is defined, for a given signal $\mathbf{x}$, as

$$\{\mathbf{x} : \|\mathbf{x}\|_0 = k\}. \tag{2.2}$$

In other words, sparsity is defined by the number of non-zero elements in a given signal: a signal that has k non-zero elements is called k-sparse. Figure 2.1 shows an example of a sparse signal. Note that even though the signal dimension is 400 samples, the number of non-zero samples is only 30, making it 30-sparse.



(a)

Figure 2.1: (a) Example of a sparse signal. Even though the signal dimension is 400 samples, the number of non-zero coefficients is only 30.

The strength of sparsity as a constraint for underdetermined inverse problems is readily perceived: it effectively reduces the number of model parameters to be recovered because several of those are zero. Therefore, even though the number of data samples is smaller than the number of model samples, if a sparse solution exists, it may be found through an underdetermined inversion procedure with a sparsity constraint, such as

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_0 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 < \gamma. \tag{2.3}$$

Notice that equation 2.3 is a constrained inverse problem, which is composed by a data fitting term and a model fitting term with the L_0 norm (associated with equation 2.2), which enforces sparsity in the model domain. While equation 2.3 promotes sparse solutions, it is difficult to minimize due to the presence of the L_0 norm, which is not convex and thus possess several local minima (Boyd and Vandenberghe, 2004). Instead, a relaxation of the L_0 norm by the L_1 norm is commonly used, which can be written as

$$\min_{\mathbf{x}} \quad \|\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi x}\|_2 < \gamma. \tag{2.4}$$

Notice that the L_1 norm in equation 2.4 is convex, and therefore can be minimized using existent technology, such as linear programming and interior point methods (Boyd and Vandenberghe, 2004).

2.2.1 Conditions for sparse recovery

Given the inverse problem defined in equation 2.3 and its approximation in equation 2.4, it is natural to ask under which conditions such problems have unique sparse solutions, or even under which conditions do equation 2.4 provide the same solution as equation 2.3. Recall that an inverse problem has a unique solution if the columns of $\mathbf{\Phi}$ form an orthogonal system, so that the null space of this matrix is only the null vector. In this setting, the mapping between model and data space is one-to-one. Because an underdetermined inverse problem implies that $\mathbf{\Phi}$ is not square, its columns can also not form an orthogonal system.

In their seminal paper, Candès et al. (2006) observe that, if a sparse solution is sought, it is only necessary that subsets of the columns in $\mathbf{\Phi}$ behave as an orthogonal system, with the size of the subset related to the sparsity of the solution. In particular, their key derivation, called the Restricted Isometry Property (RIP), establishes that the inverse problem from equation 2.4 has a unique k-sparse solution if $\mathbf{\Phi}$ obeys

$$(1 - \delta_{2k})\|\mathbf{x}\|_2 \leq \|\mathbf{\Phi x}\|_2 \leq (1 + \delta_{2k})\|\mathbf{x}\|_2. \tag{2.5}$$

The validity of the RIP rests on the constant δ_{2k}, where $2k$ denotes the sparsity of $\mathbf{x}$. Therefore, δ_{2k} is taken as the smallest number so that equation 2.5 is satisfied for every $2k$-

sparse vector $\mathbf{x}$. Intuitively, because $\delta_{2k} = 0$ implies that $\mathbf{\Phi}$ is an orthogonal matrix, small δ_{2k} conveys that every subset of $2k$ columns of $\mathbf{\Phi}$ behave approximately as an orthogonal system. Further analysis by Candes (2008) provides stricter guarantees for sparse recovery in terms of the RIP, namely

- if $\delta_{2k} < 1$, then the inverse problem in equation 2.3 has a unique k-sparse solution;
- if $\delta_{2k} < \sqrt{2} - 1$, then equation 2.3 and equation 2.4 provide the same solution.

While equation 2.5 is an important theoretical tool to analyze the behavior of sparse constraints subject to a given $\mathbf{\Phi}$, the computation of this property for even moderately sized $\mathbf{\Phi}$ is not straightforward due to the combinatorial character of the RIP. Therefore, apart from specific types of matrices, such as sub-Gaussian (Do et al., 2008), Fourier-like (Candès and Wakin, 2008) and some structured matrices (Duarte and Eldar, 2011), it is difficult to observe if a general matrix obeys the RIP or not.

I also remark that the RIP is only one of the ways one can analyze the theoretical performance of sparsity constrained underdetermined inverse problems. Other properties can be used to achieve that goal, such as probabilistic theory (Candes and Plan, 2011) and coherence-based analysis (Candes and Romberg, 2007). However, those properties suffer from the same problem as equation 2.5: they are not scalable to high dimensional problems.

2.3 Sparsity in transform domains

As previously observed in equation 2.2, sparsity is defined by the number of non-zero elements in a given signal. While Figure 2.1 shows an example of a naturally sparse signal, in general, very few signals are strictly sparse by itself. However, most signals and images possess structures that can be exploited by transforms. Therefore, several transforms have the ablility to act on these signals in such a way that their representation in the transform domain is more concise.

As expected, no single transform can make every signal sparse. Indeed, certain classes of signals are sparse using different transforms. For instance, sufficiently smooth signals are

sparse in the Fourier domain (Mallat, 2008), while piecewise smooth signals (such as natural images) are better represented in a wavelet basis (Daubechies et al., 1992). When performing sparsity-constrained inversion, it is necessary to carefully choose the transform domain in order to obtain a representation that is as sparse as possible in order to improve the quality of the signal reconstruction (Donoho, 2006).

2.4 Analysis and synthesis approaches

When using a transform domain for which the solution is assumed to be sparse, equation 2.4 needs to be rewritten to accommodate the transform domain. Defining the transform domain as the operator $\mathbf{\Theta}$, equation 2.4 can be reformulated as

$$\min_{\mathbf{u}} \quad \|\mathbf{u}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Theta}^*\mathbf{u}\|_2 < \gamma, \tag{2.6}$$

where $\mathbf{u}$ is the model in the transform domain. Note that equation 2.6 involves finding the sparse model in the transform domain, and requires an adjoint transformation to the physical domain described by

$$\mathbf{x} = \mathbf{\Theta}^*\mathbf{u}. \tag{2.7}$$

Because this formulation involves synthesizing the signal from its representation in the transformed domain, it is commonly referred to as the synthesis approach. Note that equation 2.6 is very similar to the usual preconditioning scheme used in several geophysical applications (Claerbout, 2014). However, while preconditioning is used to improve convergence of an inverse problem, here we use a transform domain to enable a sparse solution.

Alternatively, the sparsity constrained inversion can also be formulated to retrieve the solution directly in the physical domain, such as

$$\min_{\mathbf{x}} \quad \|\mathbf{\Theta}\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 < \gamma. \tag{2.8}$$

Instead of estimating a sparse representation of the signal in the transformed domain and subsequently synthesizing the signal using the inverse transform as in the problem defined in equation 2.7, equation 2.8 estimates a signal in the physical domain with the constraint

of a sparse forward transform. For this reason, it is commonly referred to as the analysis approach (Elad et al., 2007). Unlike the synthesis approach, equation 2.8 is similar to the model shaping technique commonly used in geophysical applications (Fomel, 2007).

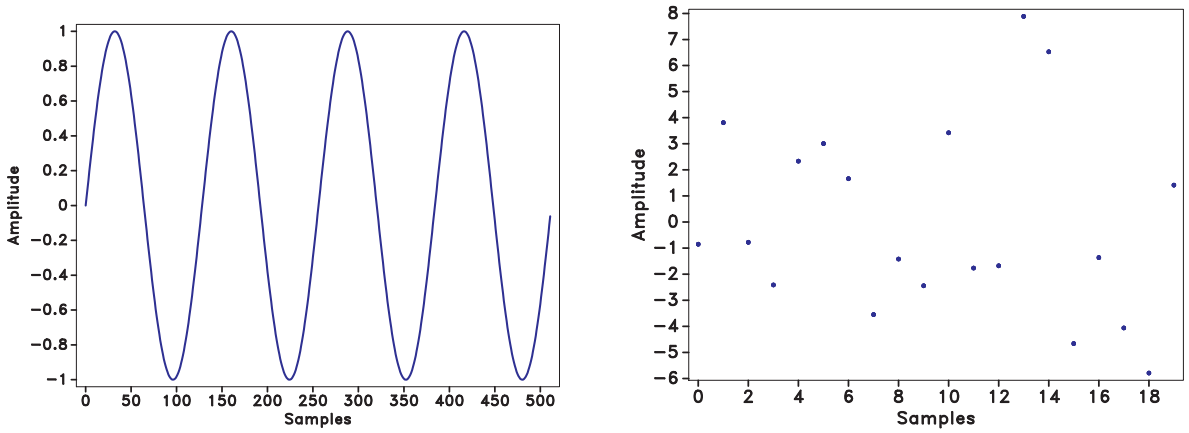
Equation 2.6 and equation 2.8 can be intuitively interpreted in similar ways, although they seem to be very different. Equation 2.6 asks how can one build a model $\mathbf{u}$ such that it satisfies the data fitting constraint using as few columns of $\mathbf{\Phi\Theta}^*$ as possible, so that the retrieved solution is sparse in the transform domain. Likewise, equation 2.8 asks which model $\mathbf{x}$, that satisfies the data fitting constraint, is the sparsest when compared (i.e., via dot products) to the elements of $\mathbf{\Theta}^*$.

2.4.1 Two toy examples

The following example illustrates the capability of sparsity constrained inversion when the signal to be estimated is sparse in a transform domain. Consider the sinusoid with length $n = 512$, shown in Figure 2.2(a), and suppose $\mathbf{\Phi}_{m\times n}$ is a matrix whose values are taken from a Gaussian distribution with zero mean and variance $1/m$, which satisfies the RIP (Candès and Wakin, 2008). I set $m = 20$, which means that the aforementioned sinusoid is embedded in a low dimensional signal $\mathbf{y} \in \Re^{20}$ through the action of $\mathbf{\Phi}$. In general, it is impossible to recover the original signal $\mathbf{x}$ from $\mathbf{y}$, shown in Figure 2.2(b).

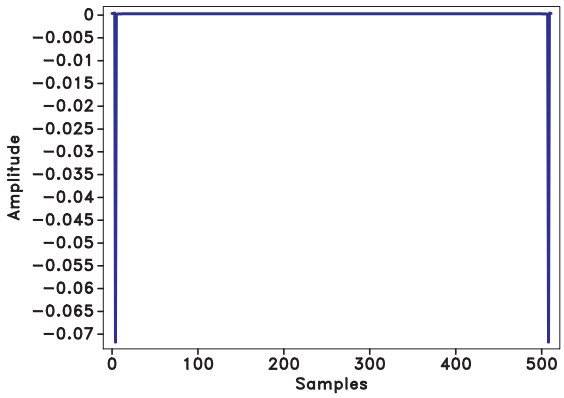
However, a sinusoid is sparse in the Fourier domain. Note that the real part of the frequency spectrum of said sinusoid, shown in Figure 2.2(c), contains only two non-zero elements. Setting $\mathbf{\Theta}_{n\times n}$ equal to the discrete Fourier transform matrix (which is an orthogonal basis) and using equation 2.6 or equation 2.8, we obtain a perfect reconstruction from $\mathbf{y}$, as observed in Figure 2.2(d). This means that even though the number of data samples in $\mathbf{y}$ is much smaller than the number of model parameters, sparsity constrained inversion can recover the true model exactly by using a sparsity measure.

Although $\mathbf{\Theta}$, in the previous example, was set as the Fourier transform, several other transformations can be expressed in terms of redundant dictionaries as opposed to orthonormal bases. Such dictionaries promote several representations of a given data in the transform

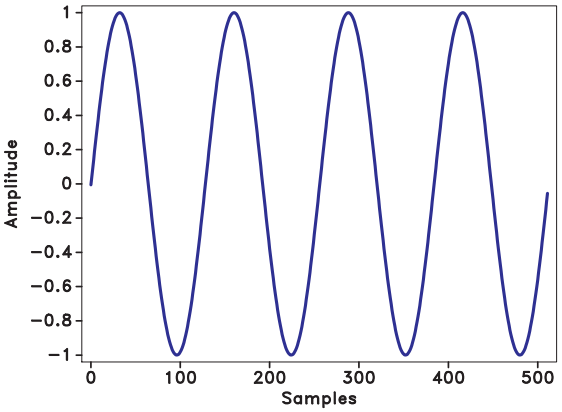


(a)

(b)



(c)

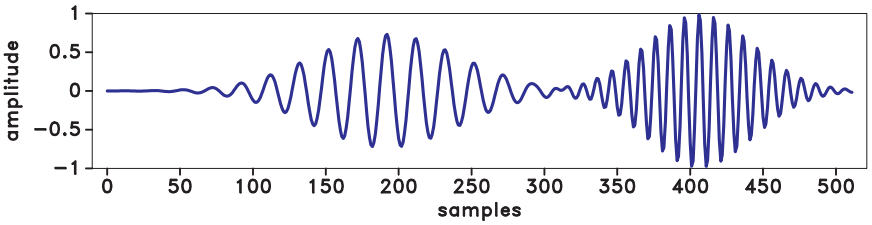


(d)

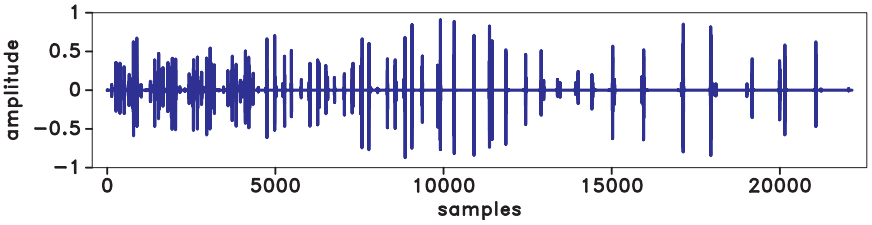
Figure 2.2: Compressive sensing toy problem: (a) Input signal, which is denoted as $\mathbf{x}$. (b) The measured data $\mathbf{y} = \mathbf{\Phi x}$. (c) Real part of the frequency spectrum of the sine in (a). (d) Reconstruction result, which shows perfect estimation from the data $\mathbf{y}$.

domain, i.e., a given model $\mathbf{x}$ may have several sequences in the transform domain $\mathbf{u}$ associated with it. In the following example, I reconstruct the signal in Figure 2.3(a) using a Gabor transform as $\mathbf{\Theta}$ in order to observe how the estimated signals from both approaches behave in the transform domain. Figure 2.3(b) shows the input signal in the Gabor domain, which implies that it is reasonably sparse in this transform domain.

Figure 2.4 shows the results of the analysis and synthesis approach compared with the true signal in the physical and Gabor domains. Notice that both approaches reasonably recovered the true signal in the physical domain, although the analysis approach seems to have a better fit to the true signal. However, in the transform domain, the solutions of both approaches are wildly different: the synthesis approach finds a much sparser representation of the signal, while the analysis approach conforms with the forward transform of the true signal. This indicates that the estimated solutions depend on the characteristics of the sparsest $\mathbf{u}$ for the synthesis approach, and the sparsest forward transform for the analysis approach, inside the constraint domain when redundant dictionaries are used.



(a)



(b)

Figure 2.3: (a) Input signal, which is denoted as $\mathbf{x}$. (b) Forward Gabor transform of the signal in (a).

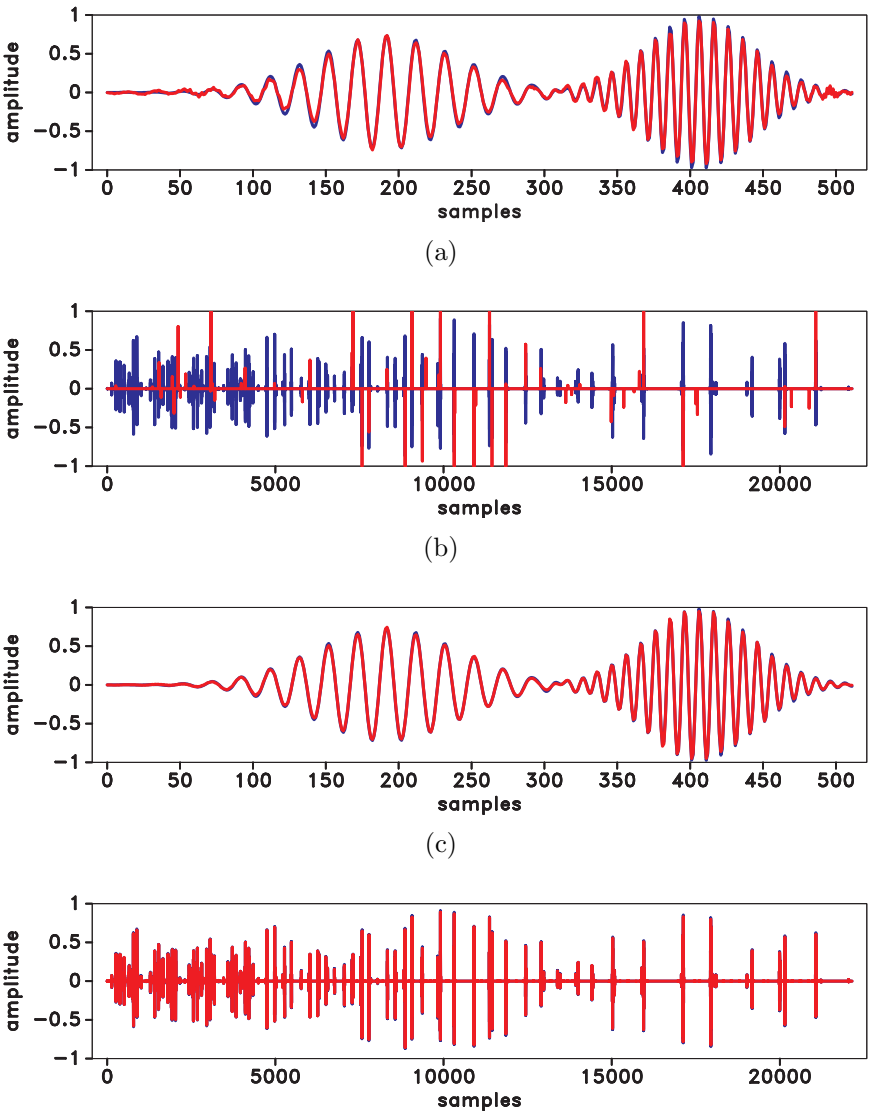


Figure 2.4: Estimated and true signals in red and blue, respectively: (a) In the physical domain as a result of the synthesis approach. (b) In the transform domain as a result of the synthesis approach. (c) In the physical domain as a result of the analysis approach. (d) In the transform domain as a result of the analysis approach.

2.4.2 Analysis vs. synthesis

While the analysis and synthesis formulations may appear substantially different - their feasible sets and objective functions lie in distinct domains - the ultimate goal of both formulations is the same: finding a sparse representation of the model that fits the measurements. In fact, using the substitution $\mathbf{x} = \mathbf{\Theta}^*\mathbf{u}$ in equation 2.8, we can rewrite the analysis approach as

$$\min_{\mathbf{u}} \quad \|\mathbf{\Theta}\mathbf{\Theta}^*\mathbf{u}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{\Theta}^*\mathbf{u}\|_2 < \gamma. \tag{2.9}$$

Notice that the formulation in equation 2.9 is equivalent to equation 2.8, in the sense that both find the same solution. Additionally, it is more similar to the synthesis approach than equation 2.8: both share the same feasible set and the solution of both formulations are subsequently synthesized into the ambient domain. In this sense, equations 2.6 and 2.9 can be considered as two types of synthesis approaches characterized by different objective functions, which enables better comparison of both approaches.

For example, if $\mathbf{\Theta}$ is an orthonormal basis (e.g. Fourier transform (Mallat, 2008)), then $\mathbf{\Theta}\mathbf{\Theta}^* = \mathbf{I}$. Consequently, for this case, equations 2.6 and 2.9 are the same, and both of these formulations are equivalent to equation 2.8. Therefore, for the case of an orthonormal basis, there is no distinction between the synthesis and analysis approaches. However, there are cases where the transform operator is a frame, which implies that it is redundant. An example of such Θ is the Gabor transform, used in the example in Figure 2.4. In those cases, the product $\mathbf{\Theta}\mathbf{\Theta}^*$ cannot be full rank, and thus, it is different from $\mathbf{I}$. It follows that the synthesis and analysis approaches are not equivalent and yield different solutions.

Furthermore, the difference between the solutions of equations 2.6 and 2.9 in the transform domain can be understood from their objective functions. The synthesis approach, as observed in equation 2.6, looks for the sparsest solution that satisfies the constraint. On the other hand, the analysis approach looks for a solution that, when shaped by the operator $\mathbf{\Theta}\mathbf{\Theta}^*$, is sparse. Clearly, not every sparse vector $\mathbf{u}$ obeys that condition. This means that

the objective function of the analysis approach may be more restrictive than that of the synthesis approach with respect to which solution they consider feasible.

If the frame used in both approaches is formed by rows vectors in general position, then the solution $\mathbf{u}$ of the synthesis approach is much sparser than that of the analysis approach, as shown in the toy example shown in Figure 2.4. This is an advantage for the synthesis approach in terms of descriptive power, i.e., it needs a smaller number of coefficients to represent a given data when compared to the analysis approach. However, it can also be detrimental in cases where the estimation of the coefficients is not accurate, i.e., in the presence of noise. This is because a smaller number of coefficients makes every coefficient carry much more significance. If, for instance, the support or magnitude of (some of) the coefficients of the synthesis solution are erroneously estimated, then the final signal $\mathbf{x} = \mathbf{\Theta}^*\mathbf{u}$ might be very different from the desired signal. Since the solution of the analysis approach is not as sparse in the transform domain, erroneous estimation may not affect the estimated signal as much. This motivates us to study the performance of both approaches on noise related geophysical problems.

In general, the exact distinction between the synthesis and analysis approaches when it comes to specific applications and data is difficult to pinpoint. Elad et al. (2007) observe that there is indeed a gap between the geometries of both problems and that each approach has its own favorable set of signals for which the estimation is optimal. Furthermore, it is highly likely that the differences between the approaches also depend on aspects of the chosen frame, such as redundancy (Becker et al., 2011). That said, it is seldom possible to assure which approach is better without testing both of them.

2.5 The seismic interpolation as an inverse problem

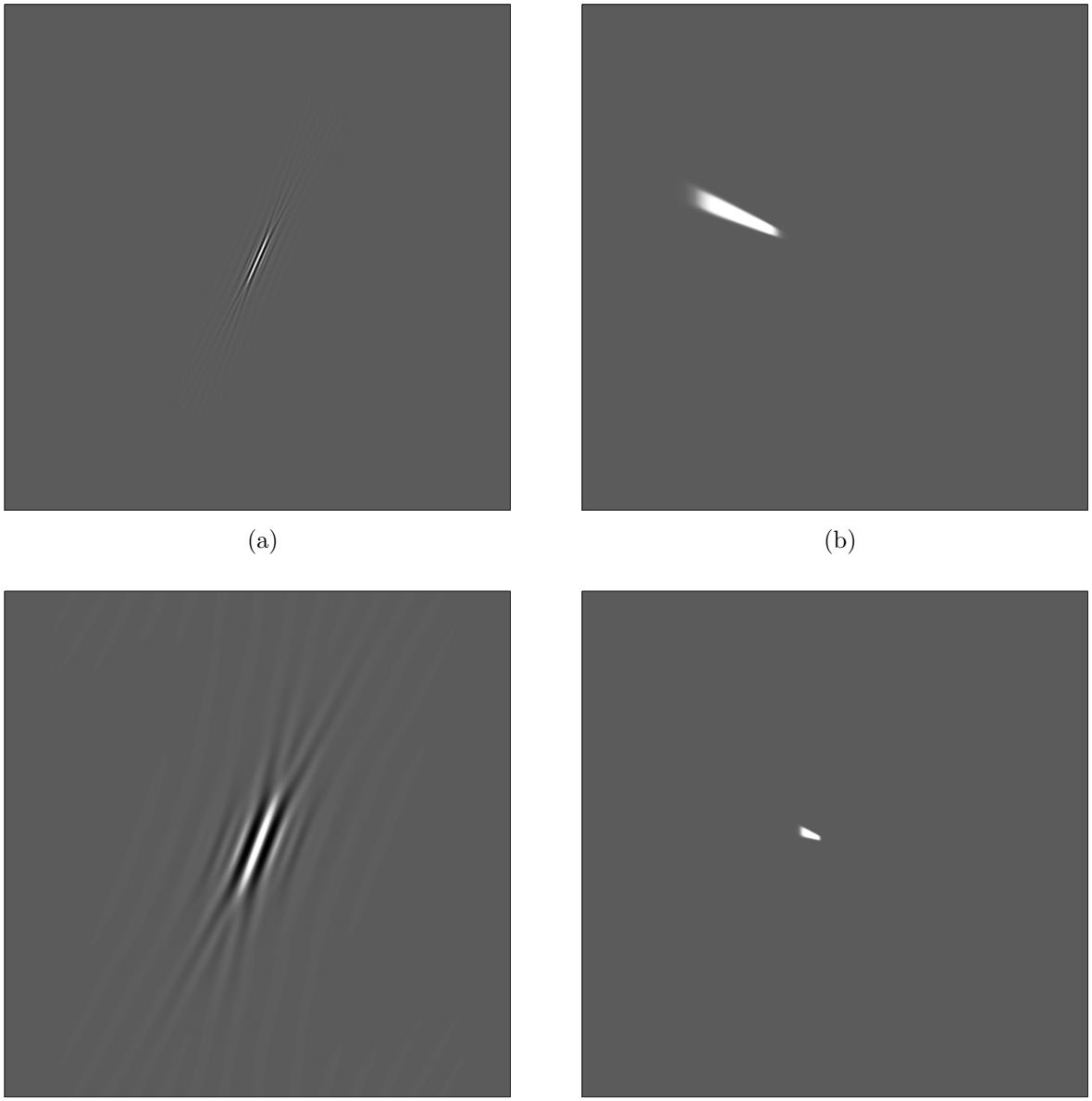
A partial goal of this thesis is to compare the performance of equation 2.8 and equation 2.6 when used to perform seismic interpolation. To achieve this, it is needed to define two very important aspects of the inverse problem: the sparsifying operator $\mathbf{\Theta}$ and the kernel $\mathbf{\Phi}$. In this section, I discuss the effects of such operators inside the seismic interpolation problem.

2.5.1 Sparsity in seismic wavefields

Seismic wavefields are usually composed of curves, i.e., seismic events. Candès and Donoho (2004) show that curvelets optimally represent piecewise smooth images with C^2 curves as discontinuities, which can be taken as the seismic events. Hence, curvelets are an attractive non-adaptive transform to represent seismic events and also can be efficiently applied to large problems (Candes et al., 2006). The curvelet transform is a frame that can be roughly seen as a dyadic-parabolic division of the Fourier spectrum of an image, which essentially means that curvelets are anisotropic and efficiently capture the directionality of events in the ambient domain. Figure 2.5 shows curvelets of different scales in the space and frequency domains. Note that using curvelets of different scales and orientations, one can represent the frequency domain entirely using curvelets. Because it is a frame, the curvelet transform is redundant and enables different results when using the synthesis and analysis approaches, like the Gabor transform used in the example Figure 2.4.

Adaptive transforms have also been proposed to represent seismic data (e.g. Zhu et al. (2015); Yu et al. (2015)), and often lead to good results. The power of an adaptive transform comes from the fact that it is trained using the available data, so that it produces a specific transform that provides a sparse representation of that specific data. More often than not, this representation is sparser than that of a non-adaptive transform. However, an adaptive transform lacks an efficient way of applying it to the data. This makes its application and training expensive when compared to non-adaptive transforms, specially on large datasets.

Although an adaptive transform could be learned using a dictionary learning algorithm such as K-SVD (Aharon et al., 2006), I restrict our discussion in this paper to non-adaptive transforms for both cost efficiency and generality. I adopt the curvelet frame as the transform domain $\mathbf{\Theta}$ in our examples and results. I use the convention $\mathbf{C}$ and $\mathbf{C}^*$ to represent the curvelet frame and the curvelet dual frame, respectively.



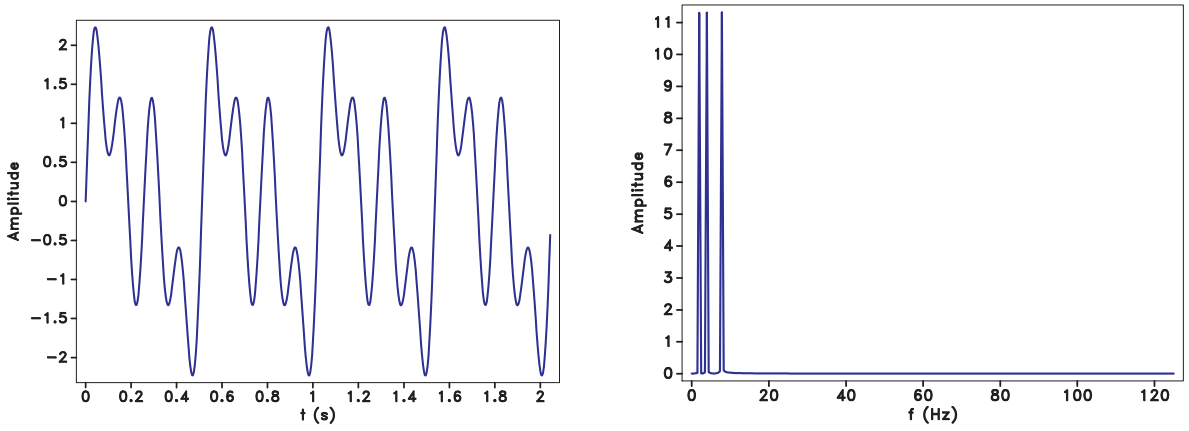
(a) (b)

(c) (d)

Figure 2.5: Curvelets of different scales: (a), (c) In the space domain. (b), (d) In the frequency domain. Note that curvelets are very similar to seismic events, i.e, a band-limited wavefront.

2.5.2 The restriction matrix

One of the components of the minimization problems described in equations 2.6 and 2.8 is the kernel $\mathbf{\Phi}$, which is a mapping for the model to the data space. For the case of seismic interpolation, $\mathbf{\Phi}$ is a restriction matrix with 1's on the positions where samples are taken in the seismic record and 0's elsewhere. Because the acquired data are incomplete, their Fourier spectrum varies depending on the pattern with which the seismic record is undersampled. This variation is important for our purposes because we assume that the seismic wavefield is sparse in the curvelet domain, which is a Fourier-related transform. In this section, I illustrate some undersampling schemes and their effects on the frequency spectrum of the simple signal in Figure 2.6(a).



(a) (b)

Figure 2.6: (a) Signal containing three different frequencies. (b) Corresponding frequency spectrum.

A common undersampling scheme selects traces regularly as a function of space. This pattern is known as the uniform undersampling scheme, illustrated in Figure 2.7(a), for the case where one out of every two samples is removed. Figure 2.7(b) shows the frequency spectrum of the uniformly undersampled data. Note that the usual aliasing effects are present due to insufficient sampling in the spatial direction.

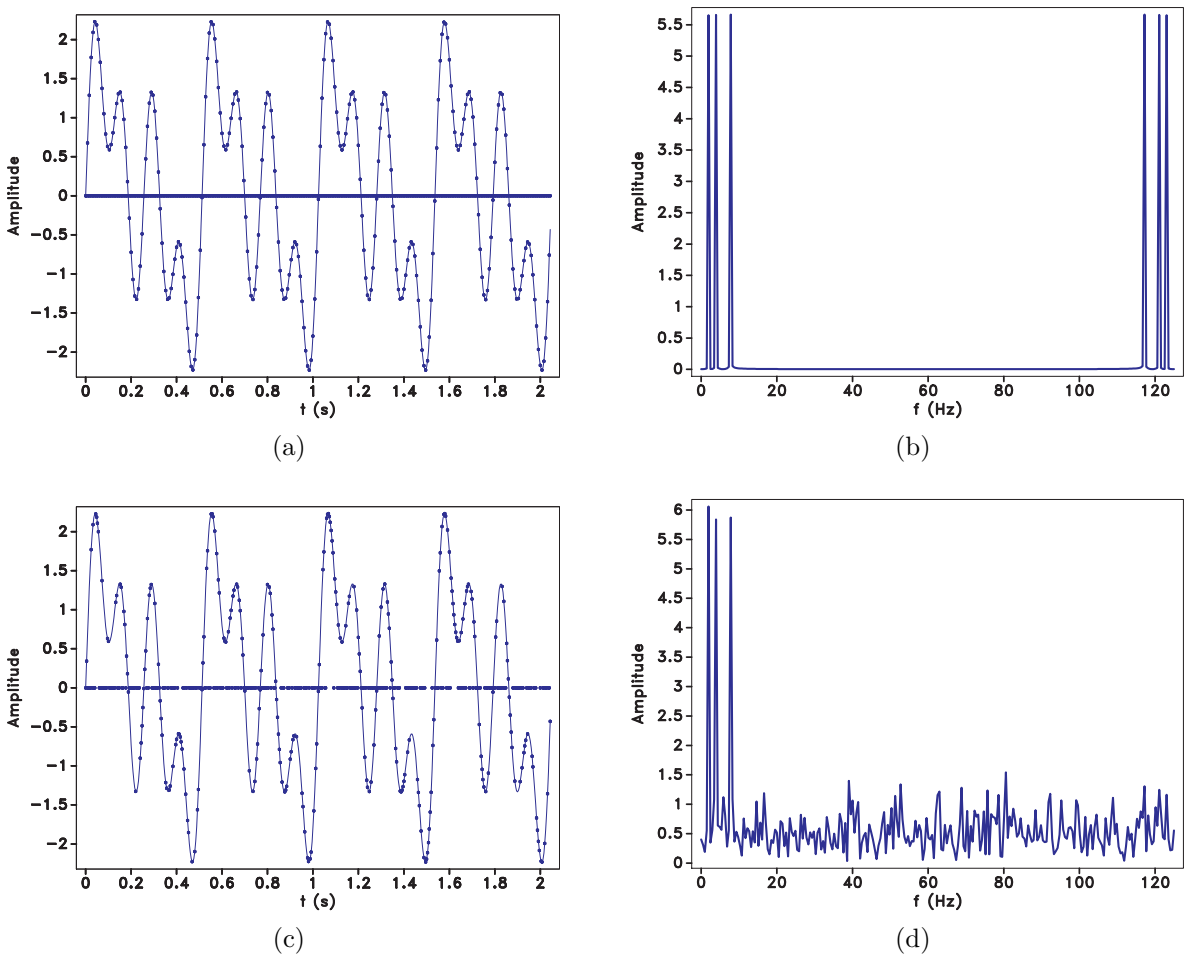


Figure 2.7: Figures displaying different undersampling schemes. The solid line indicates the original signal and the dots indicate the selected samples. Signals in time and corresponding spectra for (a), (b) Uniform sampling, (c), (d) Random sampling.

This type of undersampling renders minimization problems based on sparsity priors more challenging because the aliases and the original spectrum of the data have similar structure. Therefore, the aliases are sparse and likely to be a part of the solution of the minimization problem. Authors such as Hennenfent and Herrmann (2008) and Xu et al. (2005) encounter the same issue as they try to obtain a soluton for the missing trace reconstruction problem using uniform undersampling.

A more general undersampling scheme is obtained when the traces are taken out following a uniform probability distribution. This is known as the random undersampling scheme and is illustrated in Figure 2.7(c). The frequency spectrum of the randomly undersampled data is shown in Figure 2.7(d). Note that instead of the aliasing that usually appears in the f-k spectrum of uniformly undersampled data, the spectral leakage turns into somewhat incoherent artifacts (Donoho, 2006).

The randomly undersampling scheme is quite favorable for minimization problems that involve sparsity priors because the incoherent artifacts produced by the scheme are commonly not sparse in any reasonable transform domain. Consequently, the inversion should be able to "denoise" the f-k spectrum effectively, thus providing a good solution. This undersampling scheme also provides a more general way to undersample the data because it allows for varying gap sizes between traces in the decimated data.

2.6 Results

In this section, I evaluate the reconstruction results of the missing trace reconstruction using

$$\min_{\mathbf{u}} \quad \|\mathbf{u}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{R}\mathbf{C}^*\mathbf{u}\|_2, \tag{2.10}$$

and

$$\min_{\mathbf{x}} \quad \|\mathbf{C}\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{R}\mathbf{x}\|_2, \tag{2.11}$$

where $\mathbf{R}$ represents the restriction matrix and $\mathbf{C}$ is the curvelet transform. I propose to evaluate the quality of the results through the signal-to-noise equation

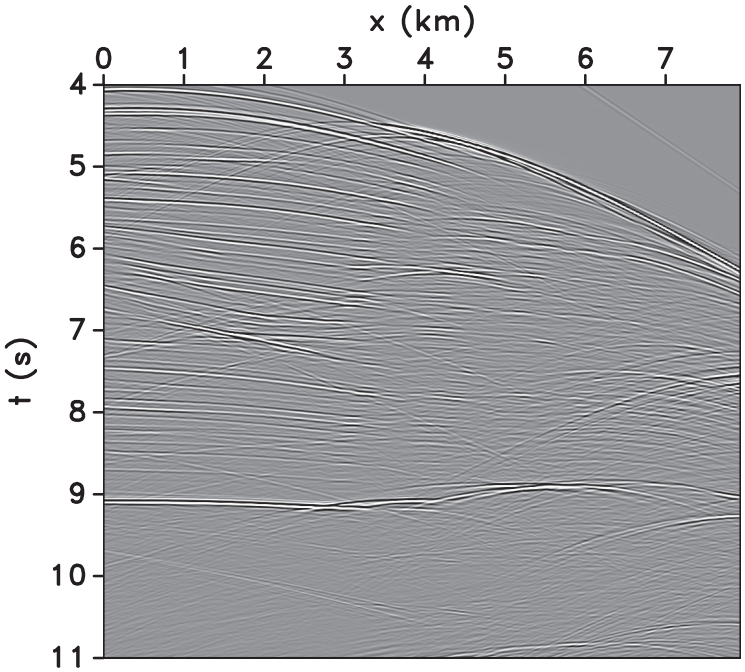
$$SNR = 20log\left(\frac{\|\mathbf{x}\|_2}{\|\mathbf{x}-\tilde{\mathbf{x}}\|_2}\right), \tag{2.12}$$

where $\mathbf{x}$ is the original data and $\tilde{\mathbf{x}}$ is the estimated data using either equation 2.10 or equation 2.11.

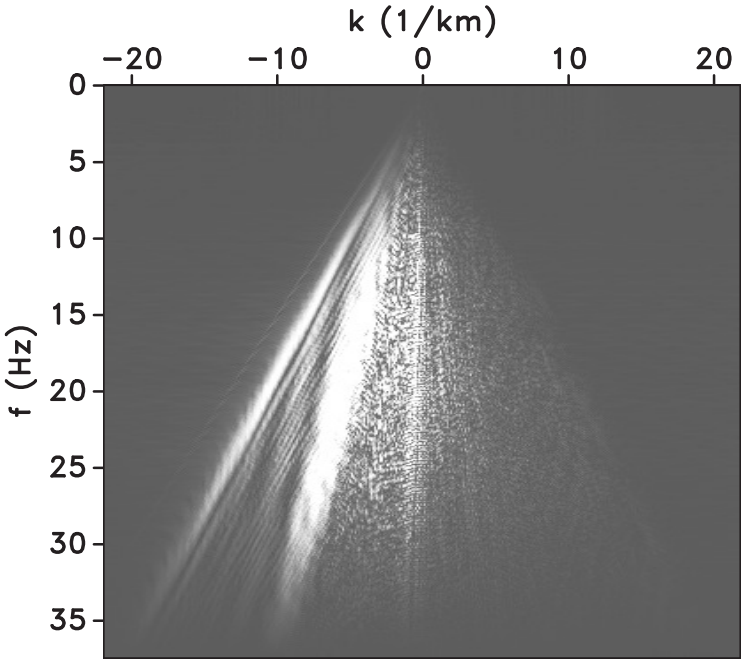
I use a shot gather from the Sigsbee 2A model, shown in Figure 2.8(a). Figure 2.8(b) shows the f-k spectrum of the shot gather. This model is appropriate for testing both approaches because it features complex geology, which generates shots with complex events with a wide range of amplitudes and conflicting dips due to the salt body. Note that the spectrum is not symmetric and features varying amplitudes.

I evaluate the results of the missing trace reconstruction for the shot gather in Figure 2.8(a) using the two different undersampling schemes proposed earlier. For every experiment, I undersample the data by 50%. In my first experiment, I uniformly undersample the original shot gather by removing one out of every two traces. Figure 2.9(a) shows the shot gather after the undersampling, while Figure 2.9(b) shows the corresponding f-k spectrum. Note that the usual aliasing is present, as expected due to the uniform undersampling.

In Figure 2.10, I show the result, difference plot and f-k spectrum for both approaches. I can see that the estimated shot gathers have some artifacts, which is expected because the uniform undersampling scheme does not favor the recovery through sparsity constraints, i.e., this type of undersampling produces a high number of aliases in the constraint domain. However, Figure 2.11 shows that such artifacts are more prominent in the synthesis approach result, which indicates that equation 2.11 may be able to better deal with such high amount of aliases in the constraint domain. The f-k spectra also shows that the f-k spectrum of the analysis solution still features less residual alias than that of the synthesis approach. These facts, along with the SNR difference, indicates that the analysis estimate is closer to the original shot gather than that of the synthesis.

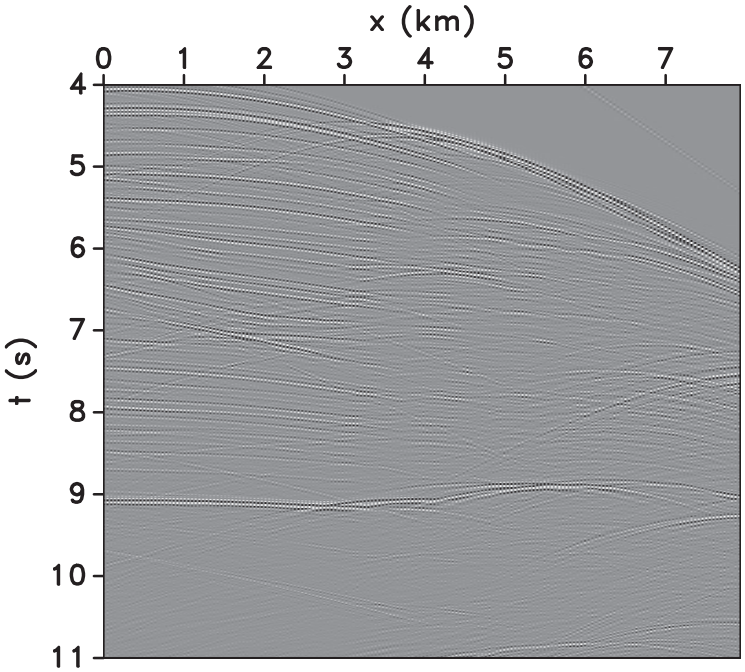


(a)

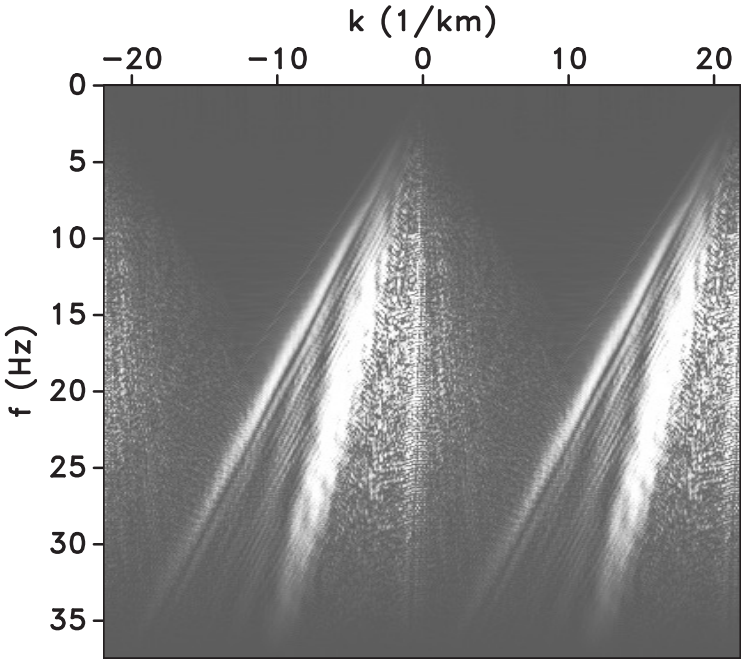


(b)

Figure 2.8: (a) Shot gather from the Sigsbee 2A model. (b) Corresponding f-k spectrum.



(a)



(b)

Figure 2.9: (a) Shot gather after uniform undersampling. (b) Corresponding f-k spectrum.

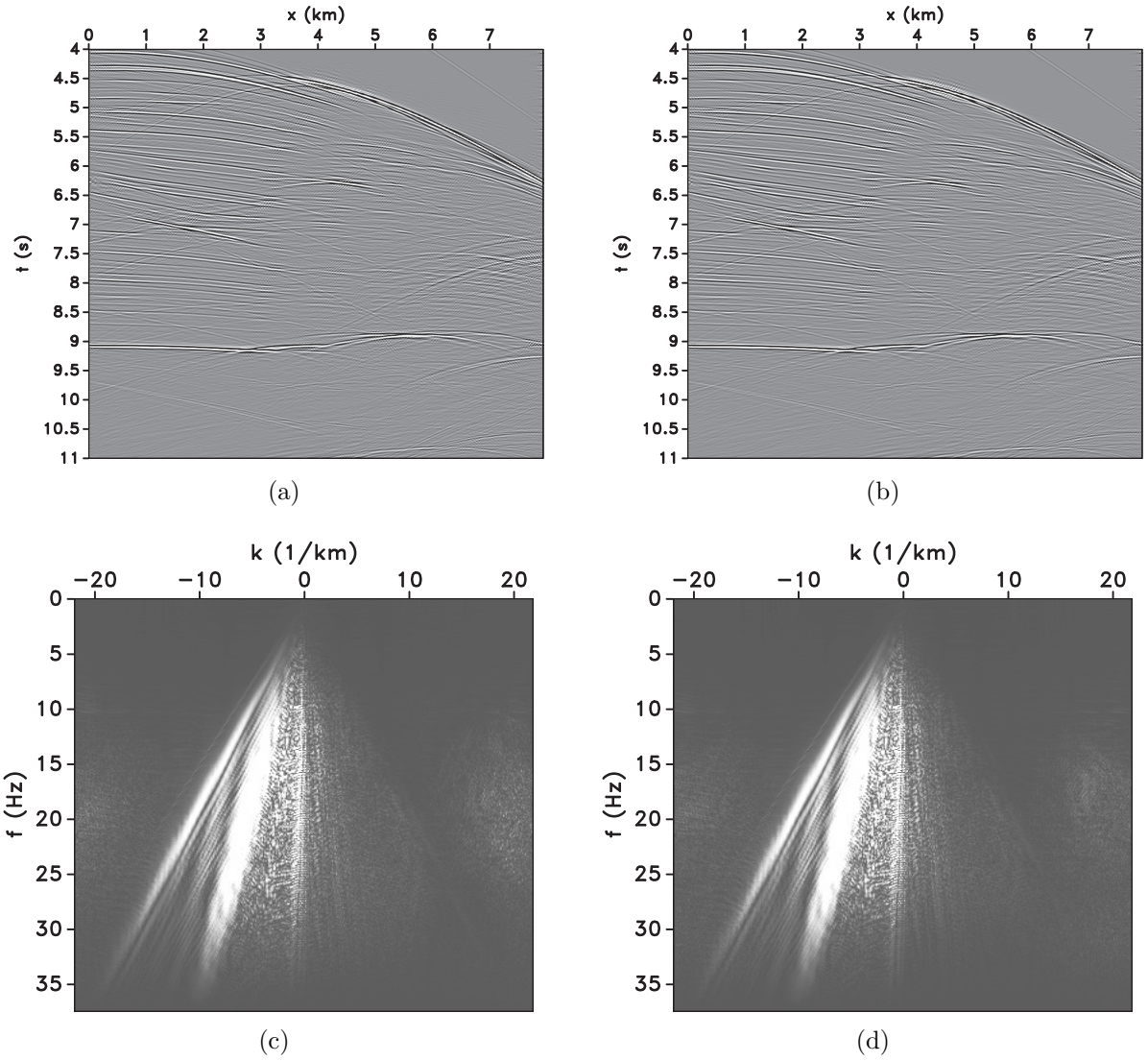


Figure 2.10: Uniform undersampling: (a), (b) Synthesis and analysis solutions. SNR of 13.29 dB and 14.83 dB, respectively. (c), (d) Corresponding f-k spectra.

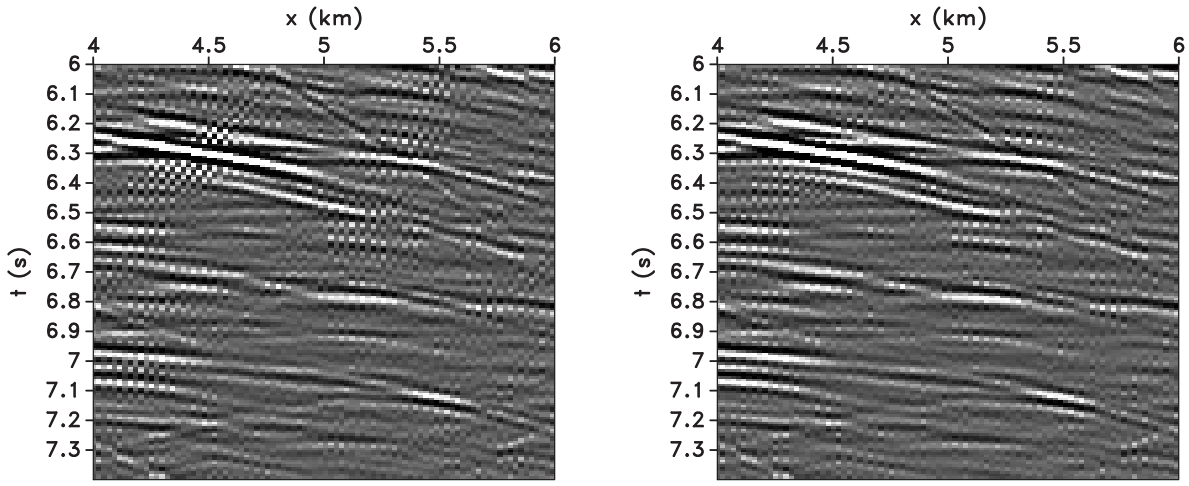


Figure 2.11: Detail from the solutions in Figure 2.10 (a) and (b): (a) Synthesis solution. (b) Analysis solution. Notice the artifacts present in (a) at coordinates $x = 4.5$km and $t = 6.35$s.

In the second experiment, I randomly undersample the original shot gather as a function of receiver position. Figure 2.12(a) shows the shot gather after random undersampling, and Figure 2.12(b) shows the corresponding f-k spectrum. Note that the undersampled shot gather features gaps of varying size. Also, the f-k spectrum shows the spectral leakage as somewhat incoherent artifacts, as expected from my previous example in Figure 2.7. Although more incoherent than the aliases present in the f-k spectra of Figure 2.10, these artifacts are not completely incoherent (i.e., as random noise). This might be because the undersampling is only performed in one dimension of the two dimensional image, in contrast with the one dimensional examples in Figure 2.7(c) and 2.7(d).

Figure 2.13 shows the estimates, and f-k spectrum for both approaches. These solutions are closer to the true model when compared to the one obtained using the uniform undersampling scheme, as evidenced by the SNR. Also, the artifacts that are present in the solutions using the undersampling scheme are mostly gone. This is accurate because the random undersampling produces incoherent aliasing, which favors sparsity constrained inversion. However, notice that in the areas where the gaps between traces are large, the quality

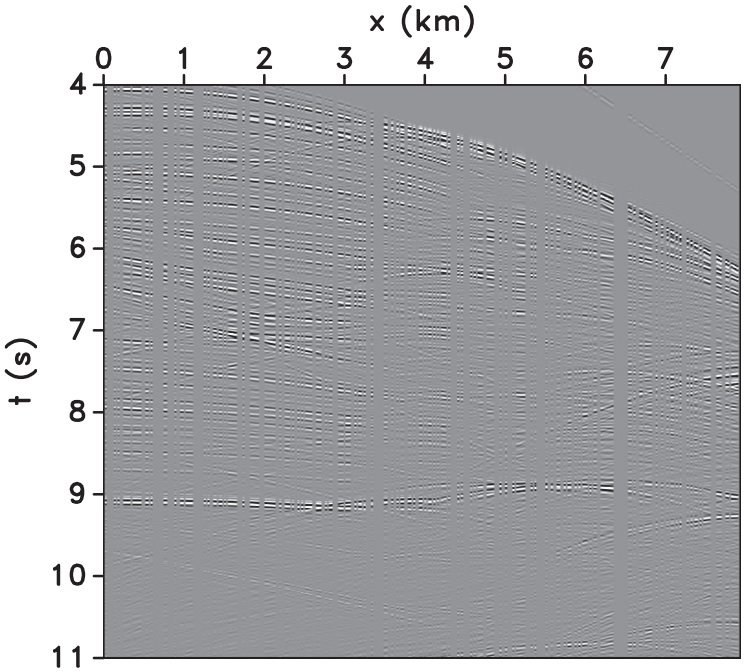
of the estimated solution is worse. Figure 2.14 also shows that some of the artifacts present in the results of the uniform undersampling experiment persist in the synthesis approach, which may indicate that although the random undersampling reduces the amount of aliasing in the constraint domain, it is not totally removed. In this case, the objective function of the analysis approach does a better job at providing a sparse solution with fewer artifacts. The f-k spectra of the solutions show that most of the incoherent alias is attenuated for both approaches, but the spectrum of the analysis solution shows less spectral leakage. This assures that the solution estimated by the analysis approach is more accurate than that of the synthesis approach for this undersampling scheme.

2.7 Discussion

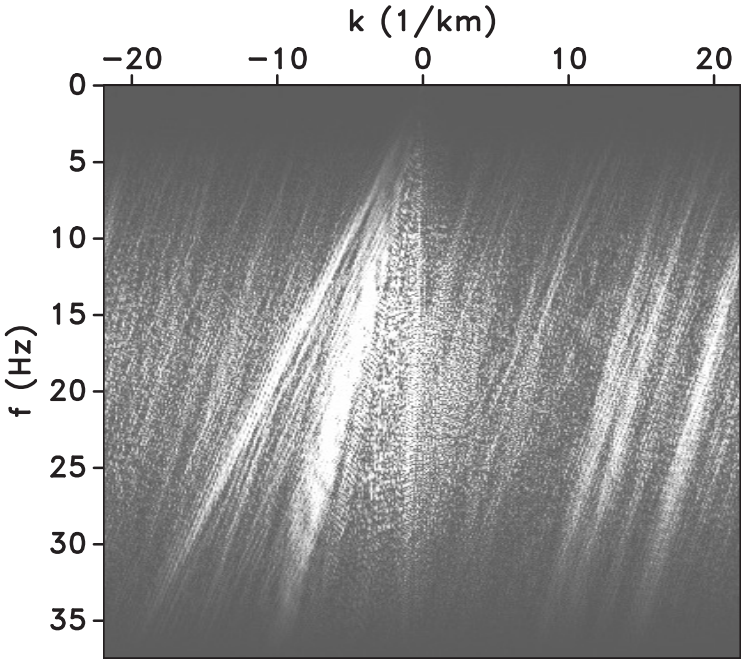
In this chapter, I discuss how to use sparsity constrained inversion to solve the problem of seismic interpolation. To that end, I introduce two different formulations: the analysis and synthesis approaches. Although similar when the sparsifying operator is an orthogonal basis, both approaches return different solutions in the transform domain when the sparsifying operator is a frame. This is the case when dealing with seismic data, because such data are optimally sparse in frames such as the curvelet or seislet transforms.

The synthetic examples show that the analysis approach performs better than the synthesis approach when it comes to interpolating seismic data. This is directly shown in terms of SNR and also due to the artifacts present in the synthesis approach solution when using both the uniform and random undersampling scheme. Such difference in performance may come from the relative lack in sparsity from the solution of the analysis approach, which means that it can be more robust to misestimations than that of the synthesis approach. Also, because the objective function of the analysis approach focuses on a smaller subset of the transform domain when compared to that of the synthesis approach, it may be more selective with respect to aliases in the constraint domain.

While both approaches display reasonable results for the interpolation problem, it is important to note that both of them still display artifacts in the estimation, especially when



(a)



(b)

Figure 2.12: (a) Shot gather after random undersampling. (b) Corresponding f-k spectrum.

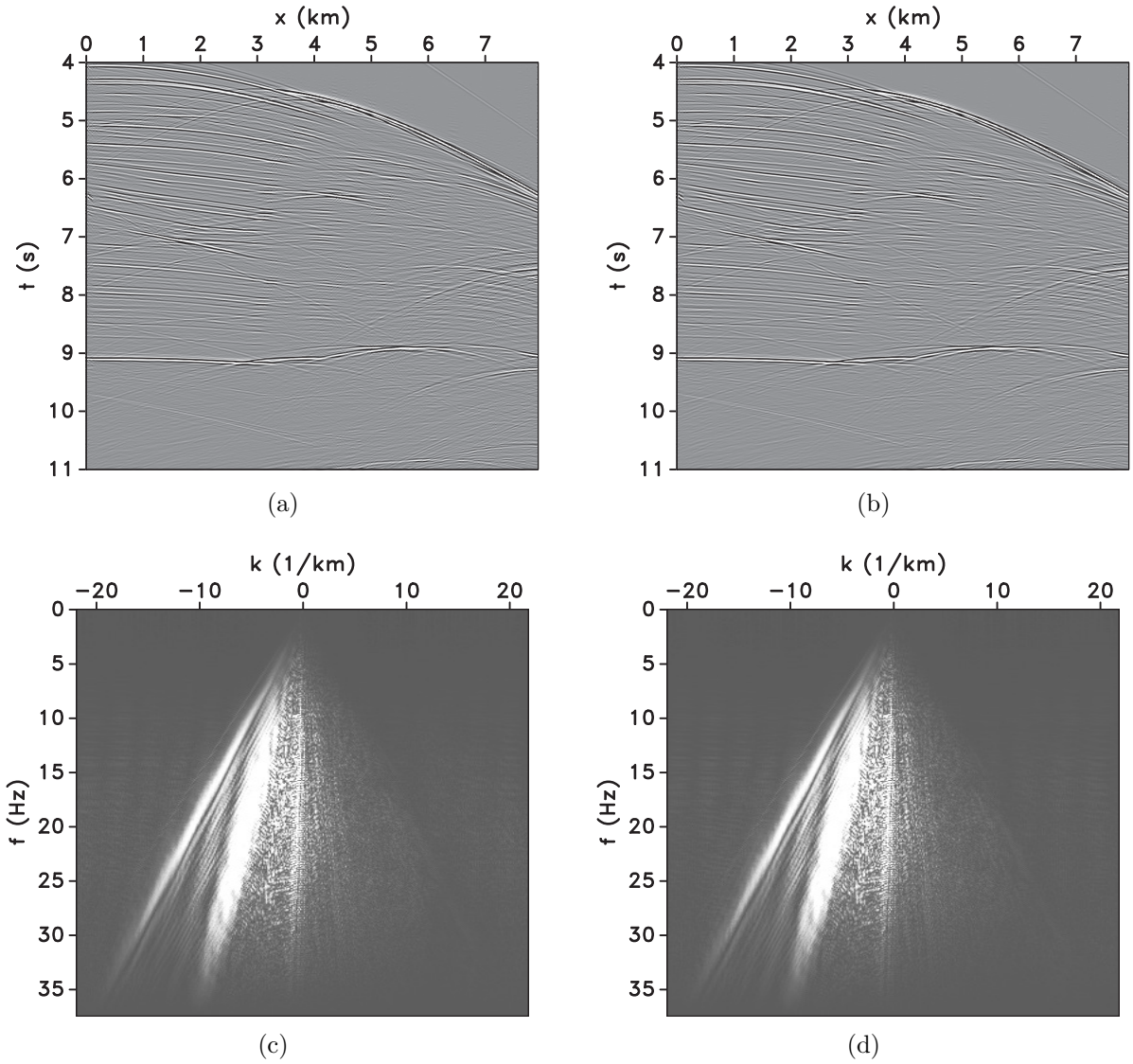


Figure 2.13: Random undersampling: (a), (b) Synthesis and analysis solutions. SNR of 16.24 dB and 17.87 dB, respectively. (c), (d) Corresponding f-k spectra.

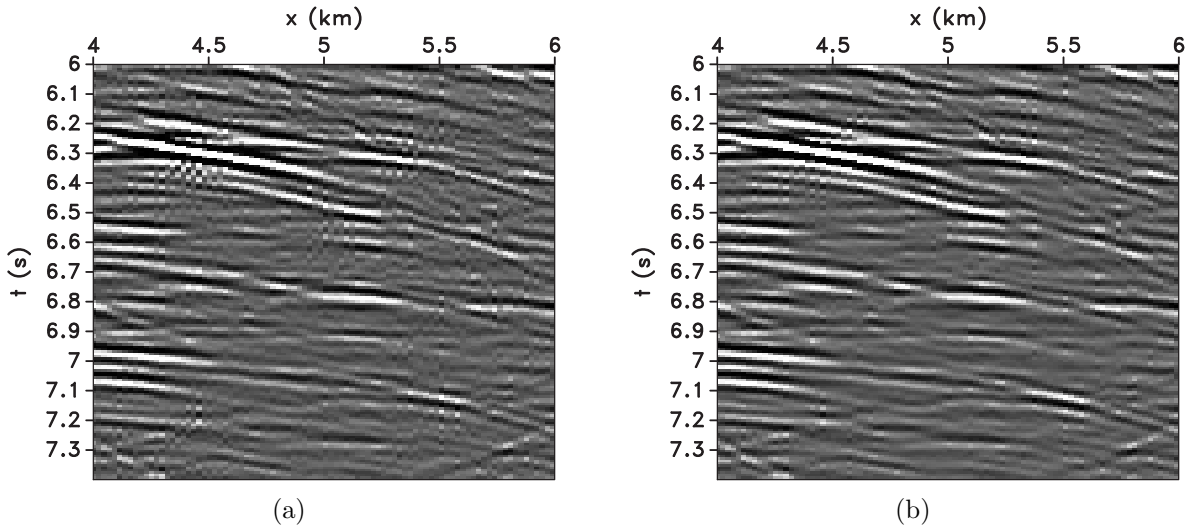


Figure 2.14: Detail from the solutions in Figure 2.13 (a) and (b): (a) Synthesis solution. (b) Analysis solution. Notice the artifacts present in (a) at coordinates $x = 4.5$km and $t = 6.35$s.

it comes to areas with big data gaps and when using the uniform undersampling. This is heavily related to the presence of aliases inside the constraint domain, i.e., models that have the same (or lower) L_1 norm as the true model, but are different from it. In such case, it may be beneficial to switch from an objective function that takes into account the magnitudes of the solution coefficients to one that does not have such influence, such as the L_0 norm in Equation 2.3. In the next chapter, I discuss an approximation to the L_0 objective function to try to attenuate such artifacts in the case of uniform undersampling and obtain better results inside large gaps between traces.

CHAPTER 3

SEISMIC INTERPOLATION USING A REWEIGHTED ANALYSIS APPROACH

3.1 Prior information in sparsity constrained inversion

In Chapter 2, I have examined the performance of two sparsity constrained inversion approaches that perform differently when $\mathbf{\Theta}$ is a redundant dictionary because of the difference in their objective function. Although both approaches estimate reasonable results in the context of seismic interpolation, both of them have problems when dealing with large gaps between traces and also because of artifacts, specially in the case of uniform undersampling. Such problems come from the presence of aliases inside the inverse problem, i.e., solutions that satisfy the data fitting term and have the same (or lower) L_1 norm of the true solution.

A common approach to solve this problem involves using weighting matrices to guide the inversion towards favored models by using prior information on the locations of the non-zero coefficients of the solution. For example, Mansour et al. (2013) divide the decimated data in several frequency slices and interpolate from the lowest frequency to the highest frequency. Because the low frequency slices are less aliased, they observe that weights built from these slices can aid the interpolation of heavily aliased frequency slices. However, this implies that the number of inverse problems to be solved is equal to the number of frequency slices, which can be many. In this chapter, I look at a more general scheme based on the findings of Candes et al. (2008), a reweighted L_1 norm approximation of the L_0 norm that does not require separation of the data in frequency slices and thus may interpolate the seismic data using a small number of inversion procedures.

3.2 L_0 norm via reweighted L_1 inversion

The errors of estimation inside large gaps of the seismic record and the artifacts the occured in the estimated solution may come from the difficulty of estimating every non-zero coefficient of the data accurately in the transform domain. Especially in the case where the

model is not strictly sparse, such as seismic data in the curvelet domain, there are several coefficients that are very small in amplitude as well as large amplitude coefficients that contribute to the solution (Zhu and Wakin, 2015). In such cases, a more accurate solution can be found by focusing on reliably estimating the amplitudes of the largest coefficients (i.e., those that contribute the most to the solution), instead of trying to estimate all of them.

3.2.1 Effect of weights on L_1 norm minimization

In practice, the situation described in section 3.2 can be achieved by using a weighting matrix that prohibits coefficients with small amplitudes to change in the inversion procedure, thus letting the inversion scheme focus on the largest coefficients. For example, suppose there is a correct solution in the transform domain, $\mathbf{u}$, for the interpolation problem. If we knew in advance the exact values of the solution, we could condition the analysis (or synthesis) approach to find such solution by setting weights as, for example Candes et al. (2008)

$$\mathbf{w}_i = \begin{cases} \frac{1}{|\mathbf{u}_i|}, & if \quad \mathbf{u}_i \neq 0 \\ \infty, & if \quad \mathbf{u}_i = 0 \end{cases} . \tag{3.1}$$

Using such weights into the objective function of equations 2.6 or 2.8, I penalize the L_1 norm of the coefficients of the solution that should be small, while lightly penalizing the coefficients that should be large. In other words, the L_1 norm, $\sum_{i=1}^{n} |x_i|$, is weighted by equation 3.1, that is, $\sum_{i=1}^{n} w_i|x_i|$. To understand better the effects of such weights, consider the following linear system:

$$\min_{\mathbf{x}} \qquad |x_1| + |x_2| \quad s.t. \quad \frac{1}{2}x_1 + x_2 = \frac{3}{2}. \tag{3.2}$$

The linear system in equation 3.2 has two possible sparse solutions: $(3, 0)$ and $(0, 3/2)$. However, $(0, 3/2)$ is the solution to the minimization problem, since it has a smaller L_1 norm. The solution of equation 3.2 is shown geometrically in Figure 3.1(a). Consider an arbitrarily large parameter λ, which scales the first term of the L_1 norm of equation 3.2,

$$\min_{\mathbf{x}} \qquad \lambda|x_1| + |x_2| \quad s.t. \quad \frac{1}{2}x_1 + x_2 = \frac{3}{2}. \tag{3.3}$$

In this case, the solution is the same as in equation 3.2. However, Figure 3.1(b) shows that the solution is the same because a large weight in the first term of the L_1 norm prohibits the x_1 component to move much during the inversion procedure. Likewise, shifting the weight λ to the second term of the L_1 norm, leads to the inverse problem

$$\min_{\mathbf{x}} \quad |x_1| + \lambda|x_2| \quad s.t. \quad \frac{1}{2}x_1 + x_2 = \frac{3}{2}, \tag{3.4}$$

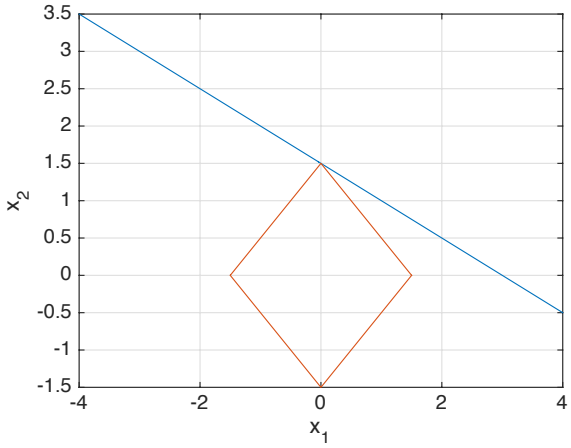
whose solution is $(3, 0)$, shown geometrically in Figure 3.1(c). This example shows the value of placing high weights in some coordinates of the objective function: it can steer the inversion procedure to a desirable solution if one knows, in advance, which coordinates to weight higher than the others.

3.2.2 Algorithm for reweighted L_1 minimization

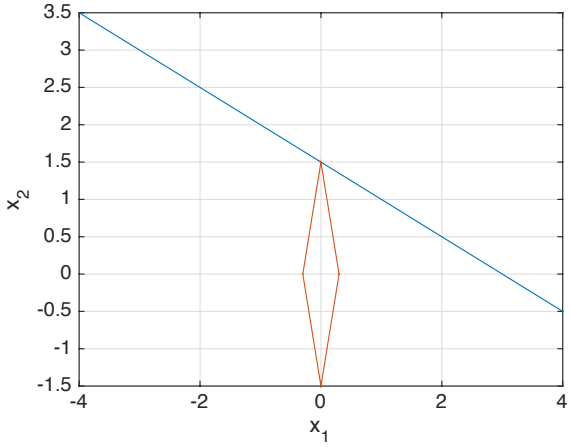
Algorithm 1 introduces a general procedure for reweighted L_1 minimization, which involves solving several L_1 programs with a different set of weights $\mathbf{W}$ at each step. Usually, there is no information about the location of the non-zero coefficients of the solution in advance, therefore it is common to start the iteration process with the unweighted problem and use the weights provided by this solution as prior information for the next reweighted iteration. The minimization problem in Algorithm 1 is written in the analysis form (equation 2.8), but it could also take the form of the synthesis approach (equation 2.6). Here, I use the analysis approach due to the superior results obtained in the last chapter. Note that a suitable stopping criterion is needed to stop the reweighted iterations.

1: $\mathbf{W}^{(0)} = \mathbf{I}$
2: **for** each iteration i **do**
3: $\quad \mathbf{x}^i = \min_{\mathbf{x}} \quad \|\mathbf{W}^{(i)}\mathbf{\Theta}\mathbf{x}\|_1 \quad s.t. \quad \|\mathbf{y} - \mathbf{\Phi}\mathbf{x}\|_2 < \gamma$
4: $\quad$ Set $\mathbf{W}^{i+1}$
5: $\quad$ If some stopping criterion is met, halt the algorithm.
6: **end for**
7: **return** $\mathbf{x}^i$

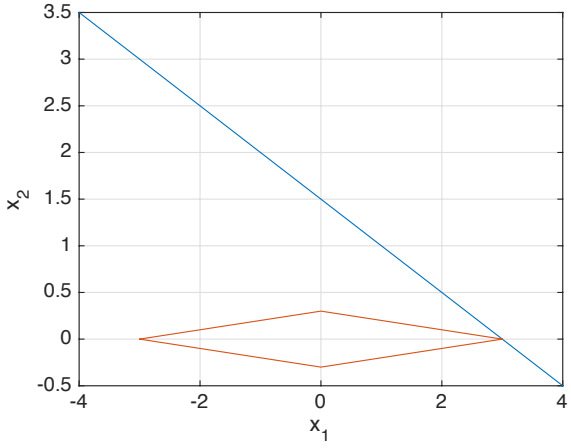
Algorithm 1: Reweighted analysis L_1 procedure



(a)



(b)



(c)

Figure 3.1: Geometric solutions of a weighted two-dimensional inversion example: (a) Unweighted solution. (b) Arbitrarily large weight in x_1. (c) Arbitrarily large weight in x_2. The blue and orange lines indicate the constraint and the L_1 ball centered at the origin, respectively.

There are multiple ways of building the weight matrix $\mathbf{W}$, which is always diagonal. For example, one can devise a binary matrix with an appropriately large value for components of the model that are supposed to be small, and small values otherwise. Another possibility is to automatically set the weighting matrix using a rule based on the amplitudes of the coefficients in previous reweighted steps. Candes et al. (2008) show that, with the right choice of weights, the reweighted algorithm approximates the L_0 norm of the objective function. In particular, setting the weights as

$$\mathbf{W}^{i+1} = diag\left(\frac{1}{|\mathbf{\Theta}\mathbf{x}^i(j)| + \epsilon}\right), \tag{3.5}$$

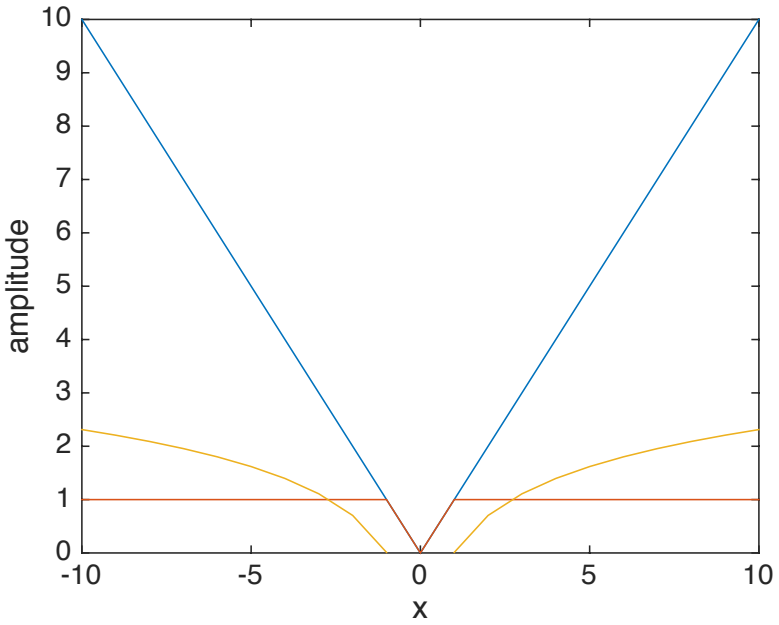
where the index j indicates the coefficient of the sequence x^i, ensures that each step of the reweighted algorithm approximates the function $f(x) = \sum_{j=1}^{n} log(|\mathbf{\Theta}\mathbf{x}^i(j)| + \epsilon)$, which very closely resembles the L_0 norm function as $\epsilon \to 0$ (Candes et al., 2008), as shown in Figure 3.2. Notice that equation 3.5 describes weights with the same behavior as equation 3.1, but with a regularization parameter ϵ that prevents that weights at locations where $\mathbf{\Theta}\mathbf{x}^i$ is strictly zero to assume infinity value. Other functions can also be used as an approximation for the L_0 norm, such as $f(x) = \sum_{j=1}^{n} atan(|\mathbf{\Theta}\mathbf{x}^i(j)|/\epsilon)$, which is achieved using the weights

$$\mathbf{W}^{i+1} = diag\left(\frac{1}{|\mathbf{\Theta}\mathbf{x}^i(j)|^2 + \epsilon^2}\right). \tag{3.6}$$

In this thesis, I focus on the effects of approximating the L_0 norm using the reweighted scheme above in the context of seismic interpolation. Since either equation 3.5 and 3.6 satisfy this purpose, I do not go into a comparison of such weight schemes and use equation 3.5 as the update rule for the weight matrix $\mathbf{W}$ throughout the chapter, for simplicity.

3.3 Effects of the parameter ϵ

When building the weight matrix $\mathbf{W}$, the parameter ϵ heavily influences the outcome of the inverse problem, since it modifies the amplitude of the weights. Intuitively, in order to preserve the structure of the weights as established in equation 3.1, ϵ should be set to a small value such that it does not contribute much to the weights at large components and



(a)

Figure 3.2: Illustration of the different norms involved in this chapter. The lines in blue, red and yellow represent the L_1 norm, the L_0 norm and the function $f(x) = \sum_{i=1}^{n} log(|\mathbf{x}_i| + \epsilon)$, plotted for $\epsilon = 10^{-2}$. Note that the proposed function approximates the L_0 norm better than the L_1 norm for large values.

also does not overshadow the weights at small components.

In order to examine the effects of the parameter ϵ in the structure of the weight matrix $\mathbf{W}$, I transform the Sigsbee data in Figure 2.8(a) to the curvelet domain and use the rule in equation 3.5 to calculate the weights associated with the transform domain sequence for several values of ϵ. Because the transform domain sequence associated with the Sigsbee data is very high dimensional (about 3 million coefficients), I bin the results in a 100×1000 grid representing amplitude versus samples, present in Figure 3.3. Note that the colors denote the density of values in a given bin, with red and blue describing high and low density, respectively.

Figure 3.3 expressly shows that while large values of ϵ confine the weights to a narrow band of values, where each of the solution is assigned weights of similar values. In such cases, because the weights are very close in every coefficient, the effects of weighted inversion shown in Figure 3.1 may not be seen. As the value of ϵ varies from large to small, the weights become more distant in magnitude. In particular, the smaller the value of ϵ, the more small weights are present in the weight matrix. Drawing from the example in Figure 3.1, smaller values of ϵ let more coordinates of the solution change during the inverse problem, while larger values of ϵ contribute to make most of the coordinates zero.

The following toy example using the model in Figure 2.1, which is a signal that is sparse in the physical domain, illustrates the importance of ϵ in the inversion procedure. The operator $\mathbf{\Theta}$ is the identity matrix due to the signal being sparse in the physical domain. I take $\mathbf{\Phi}$ to be a Gaussian matrix, as in the sine example, because it satisfies the RIP. Figure 3.4 shows the solution of the analysis approach, i.e., the solution that is used to construct the first weight matrix. Note that it is very different from the true model, which means that it is not an acceptable solution. Figure 3.5 shows the solutions of Algorithm 1 for several values of ϵ and 5 reweighted iterations. As expected, the solutions for different ϵ improve over that of the analysis approach. However, most improvement is shown when the ϵ value promotes mild sparsity, i.e., only a portion of the coordinates of the solution possess small weights.

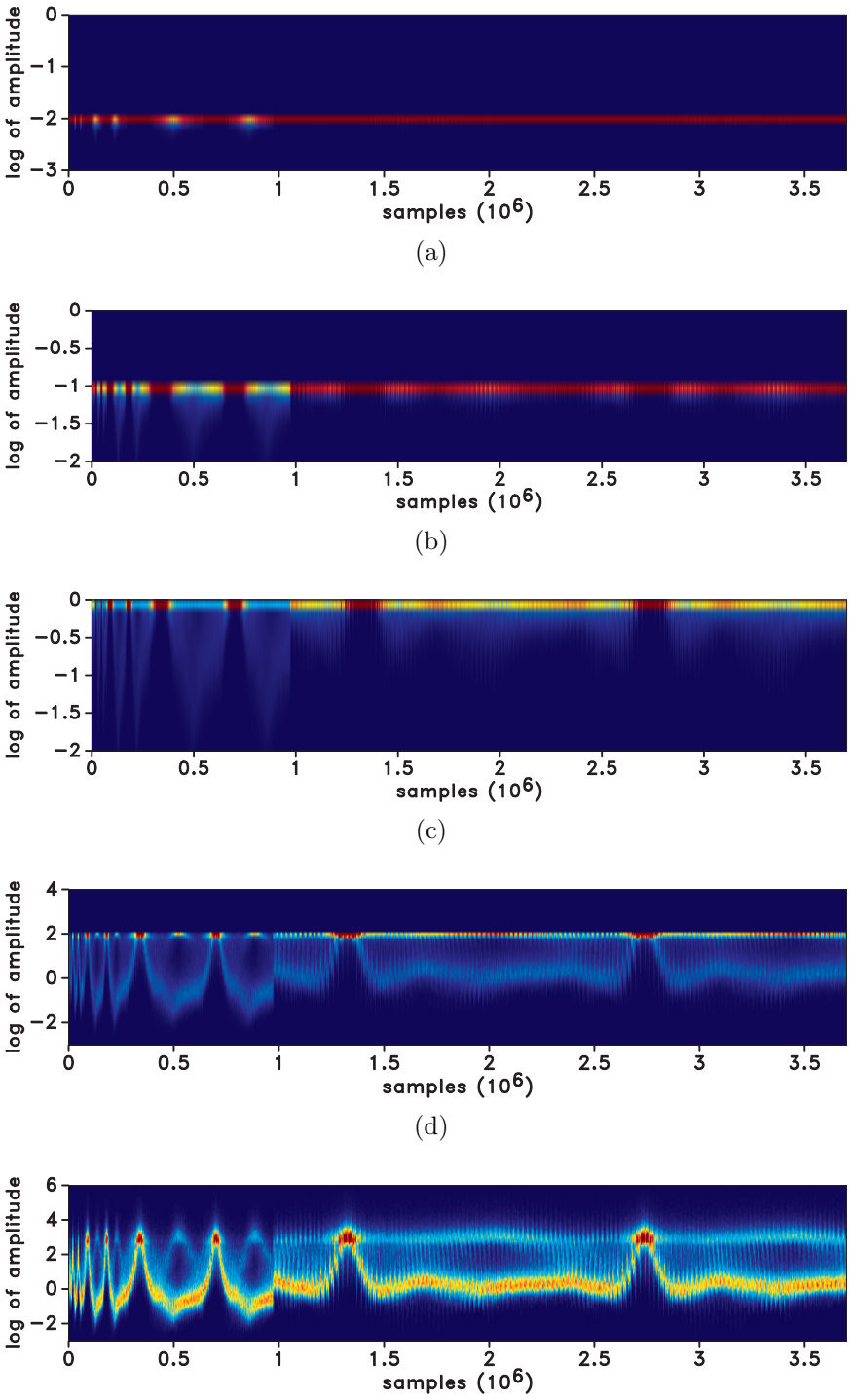


Figure 3.3: Binned weight matrices for the true model in Figure 2.8(a) for several values of ϵ: (a) $\epsilon = 10^2$. (b) $\epsilon = 10$. (c) $\epsilon = 1$. (d) $\epsilon = 10^{-2}$. (e) $\epsilon = 10^{-6}$.

For relatively large and small values of ϵ, the obtained solution is not as accurate.

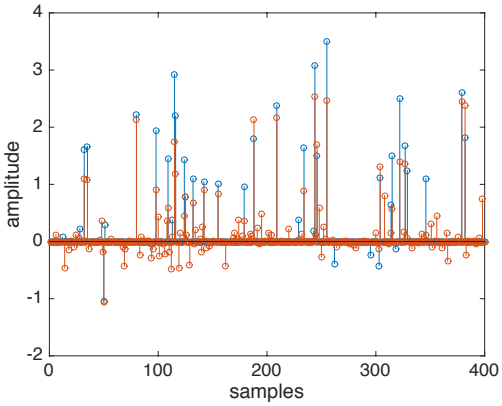


Figure 3.4: Toy example using Algorithm 1: Analysis solution. The true model and the estimated solution are displayed in blue and orange, respectively.

I also examine the results of Algorithm 1 when used to interpolate the randomly undersampled Sigsbee data described by Figure 2.12(a) for several values of ϵ for 10 reweighted iterations. Figure 3.6 shows the results of the inverse problem. Algorithm 1 provides an increase in SNR on the final solution and decrease artifacts when compared to the analysis or synthesis approach alone. It also makes the solution sparser with increasing reweighting iterations, as shown in Figure 3.7. Note that all the solutions seem very close, which is different from what is displayed in the toy problem shown in Figure 3.5. Also, none of the solutions in Figure 3.6 were able to better estimate areas where large gaps between traces exist with this number of reweighted iterations. This may be caused by the different nature of both problems. Because the seismic interpolation problem uses a restriction matrix as a kernel, there is no information about the missing traces in it. Therefore, many local mimina may exist in the inverse problem. If the solution of the analysis approach is close to a local minimum of the L_0 norm objective function, then any choice of weights based on such solution may direct the reweighted solution towards that local minimum.

While for the toy example in Figure 3.5 I can obtain a reasonably estimated model using few reweighted iterations, this is not that same for the Sigsbee case in Figure 3.6. However,

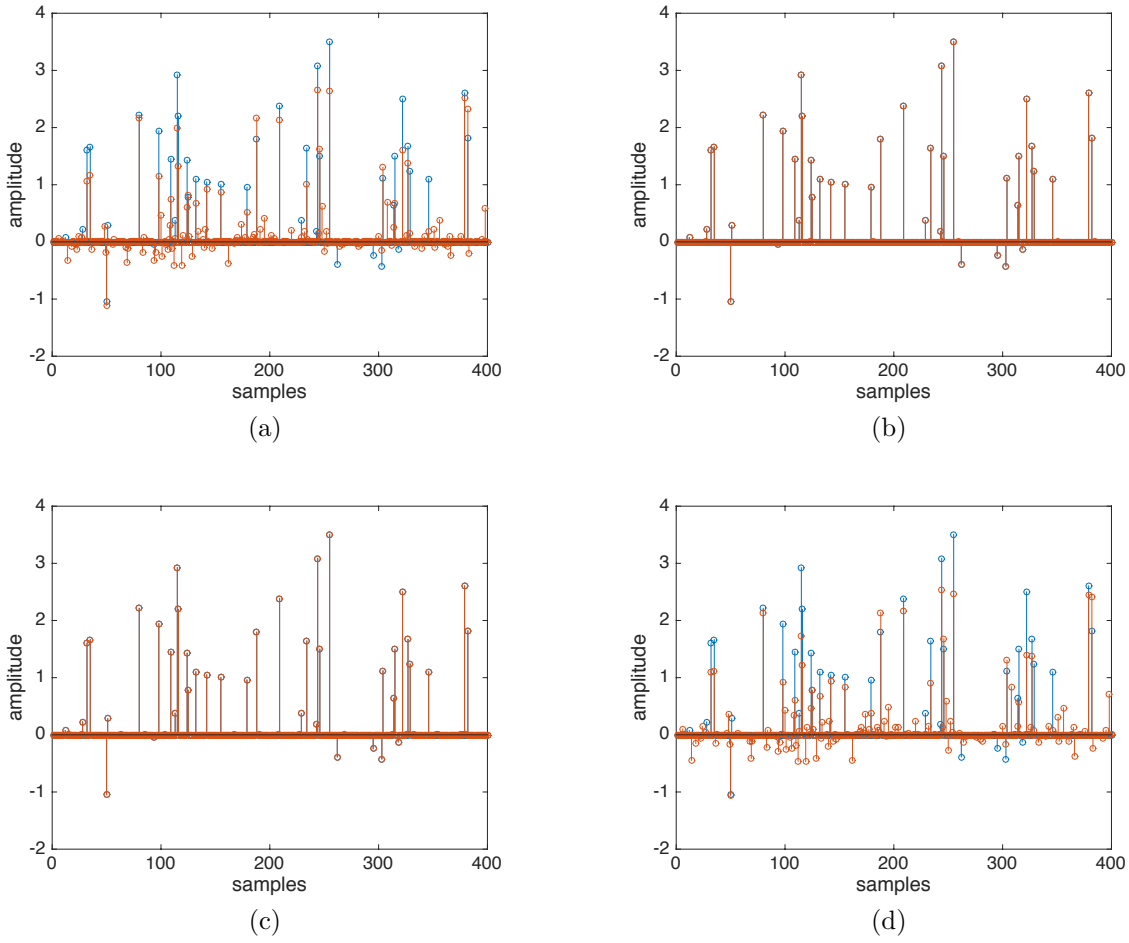


Figure 3.5: Toy example using Algorithm 1: (a) $\epsilon = 10^2$ (b) $\epsilon = 1$. (c) $\epsilon = 10^{-2}$. (d) $\epsilon = 10^{-6}$. The true model and the estimated solution are displayed in blue and orange, respectively.

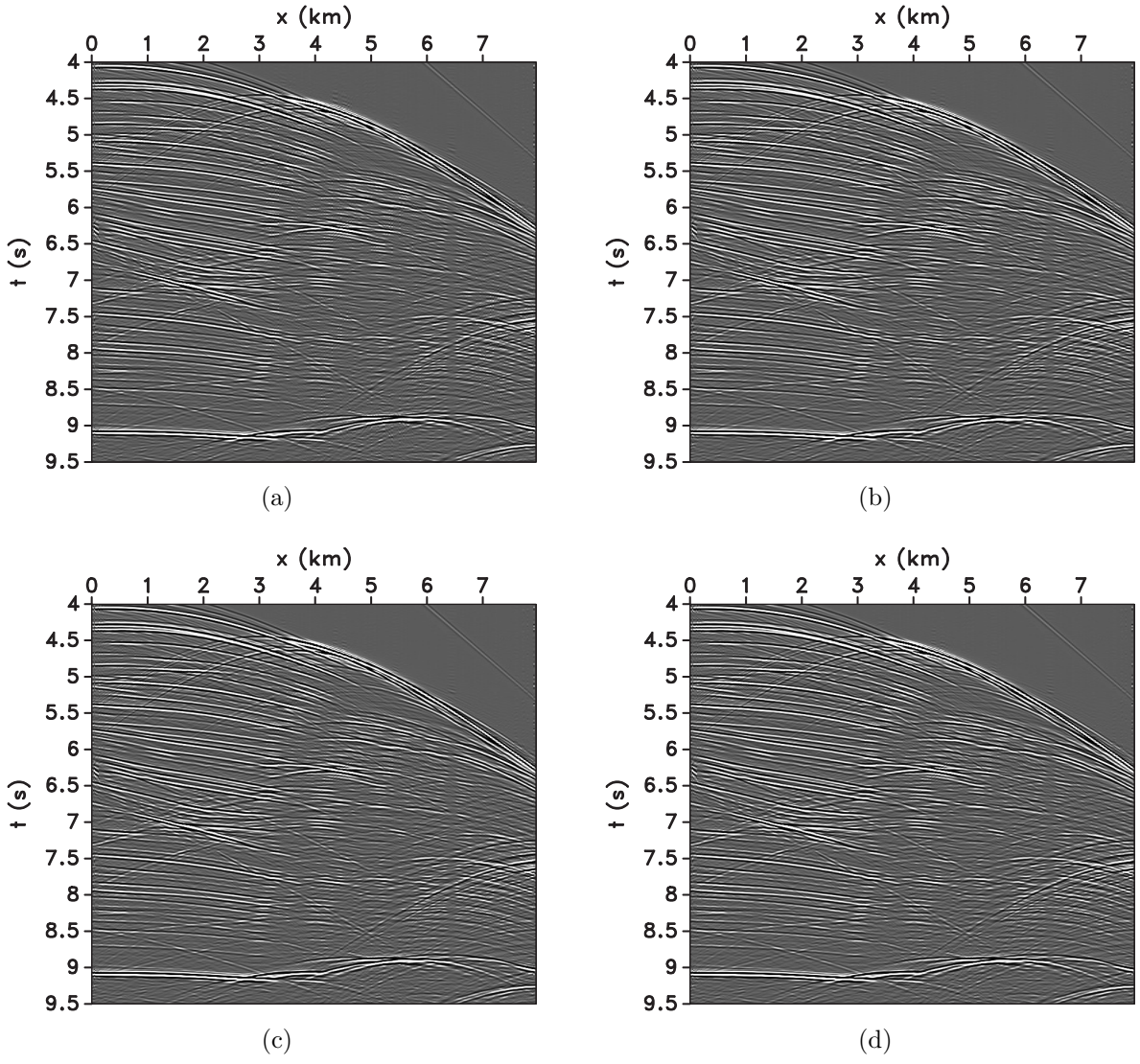
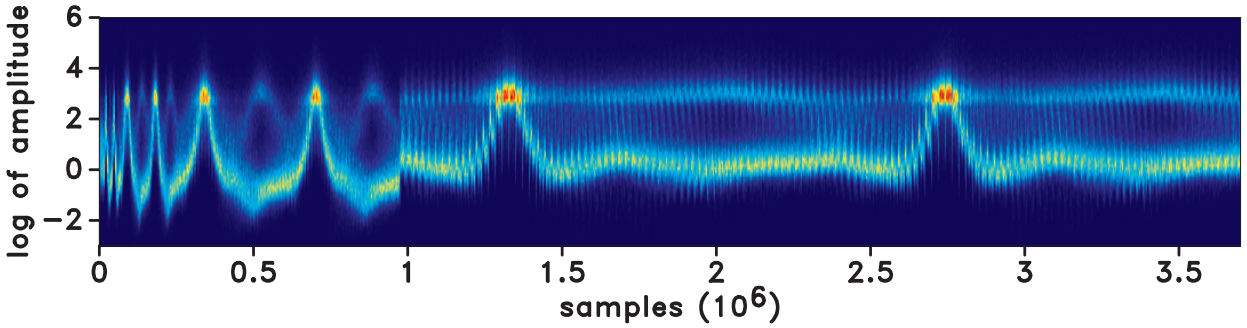
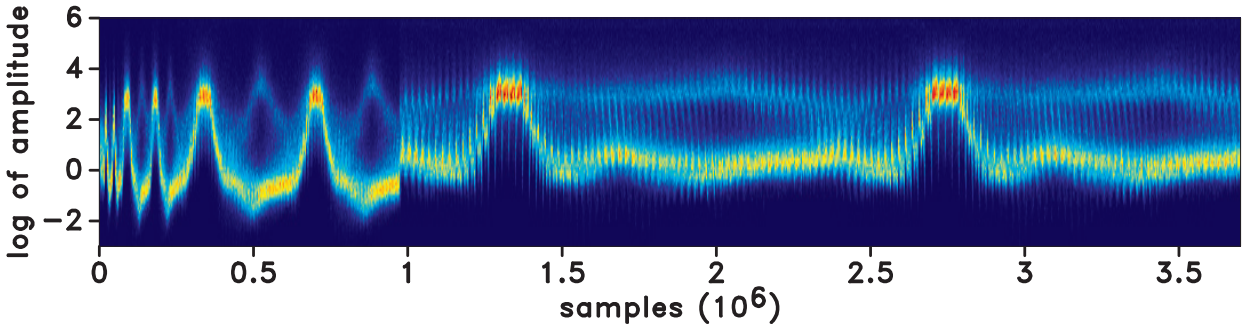


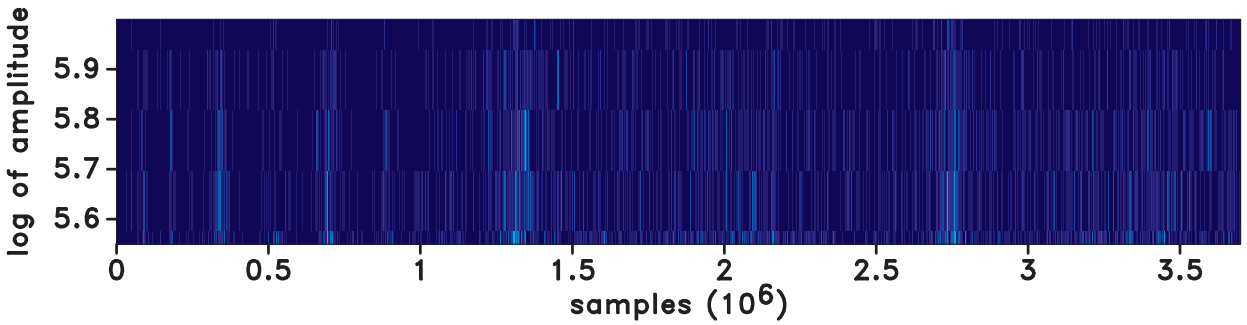
Figure 3.6: Interpolation results for the minimization problem using Algorithm 1 and equation 3.5: (a) $\epsilon = 10^2$, SNR of 17.11dB. (b) $\epsilon = 1$, SNR of 18.96. (c) $\epsilon = 10^{-2}$, SNR of 18.62 dB. (d) $\epsilon = 10^{-6}$, SNR of 18.54 dB.



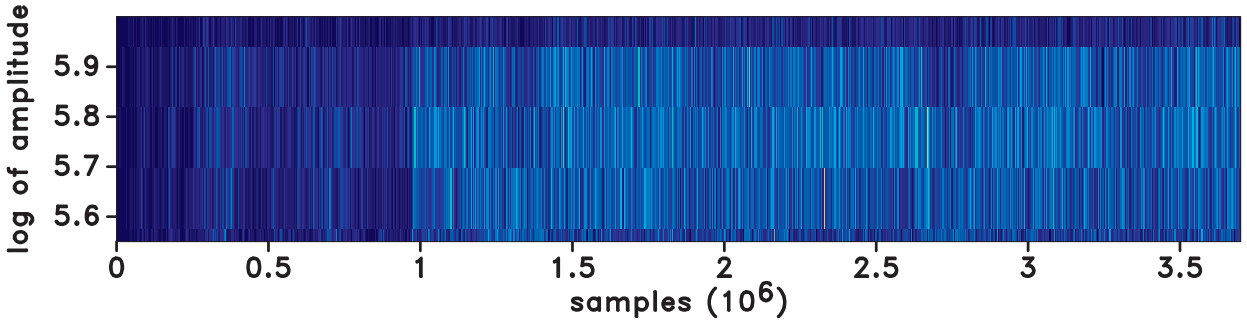
(a)



(b)



(c)

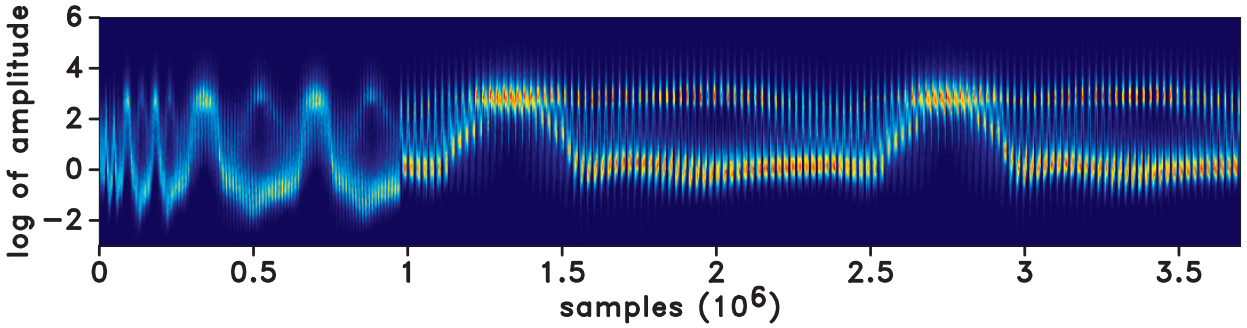


(d)

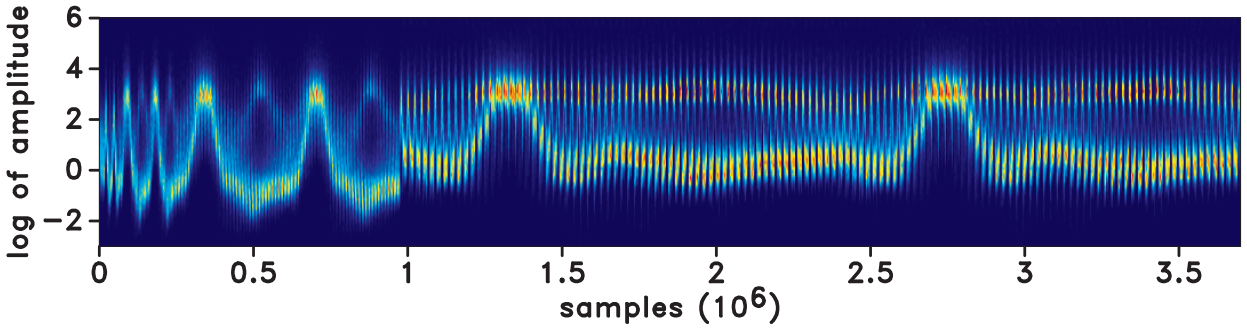
Figure 3.7: Binned weight matrix for $\epsilon = 10^{-6}$: (a) first and (b) second reweighted iterations. (c), (d) Detail of the top part of (a) and (b), respectively. Note the increase in number of high amplitude weights, which correspond to small values in the coefficients, in (d).

this is a more difficult problem, i.e., it is very high dimensional in the transform domain and has an extensive null space due to its kernel, the restriction matrix. Also, in this case, one is dealing with a model that is not strictly sparse. Due to such problems, one may ask whether the true model is, indeed, a global minimum of the objective function. In other words, is the true solution the sparsest possible sequence in the transform domain that fits the data fitting term? Figure 3.8 show weights produced by the forward curvelet transform of Figure 2.8(a) and Figure 3.6(d). Note that the true solution is less sparse, i.e., possesses fewer coefficients in the bins denoting high amplitude weights, when compared to the estimated solution. This means that, in this case, one may benefit from looking for a less sparse sequence instead of enhancing sparsity.

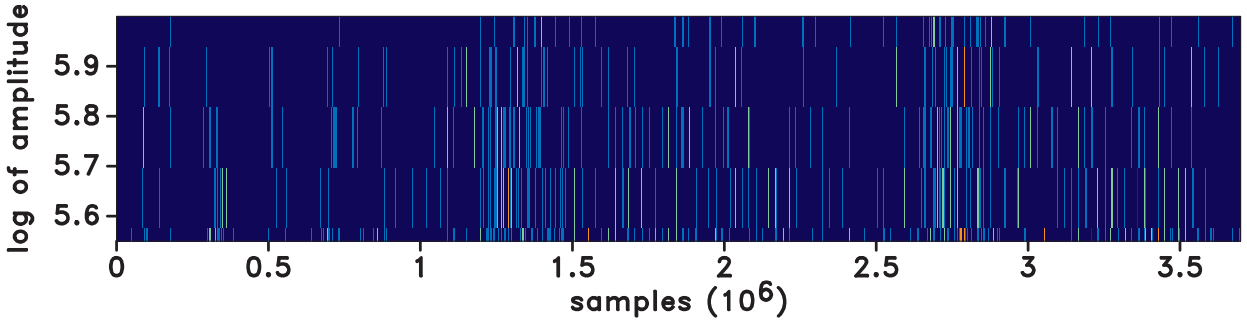
Another concern is that when using Algorithm 1, each reweighted iteration has its own global (and local) minimum, which is influenced by the weight matrix. Therefore, the procedure may not have a stable convergence rate, i.e., the rate at which the model changes in adjacent reweighted iterations may not be fixed. In fact, Candes et al. (2008) observes that there is no guarantee that the sequence of reweighted iterations converge at all. In this case, how many reweighted iterations should be run so that the weight matrix sets up an inverse problem which provides a solution comparable to the true model? Figure 3.9 shows the results of the interpolation problem for 25, 50, 75 and 100 reweighted iterations for $\epsilon = 1$. Note that their SNR are relatively close, which indicates that 25 reweighted iterations actually converge for this problem. However, a comparison of Figure 3.8(d) and Figure 3.10(b) shows that the estimated solution changes dramatically as Algorithm 1 progresses. In particular, the estimated solution in iteration 10 is sparser than that of the analysis solution, but less sparse than the estimated solution of iteration 25. This is probably due to the weight matrix design, which enforces sparser solutions even though the model is not strictly sparse. That said, such a large number of reweighted iterations may prove to be undesirable for geophysical applications, especially if the problem is very large.



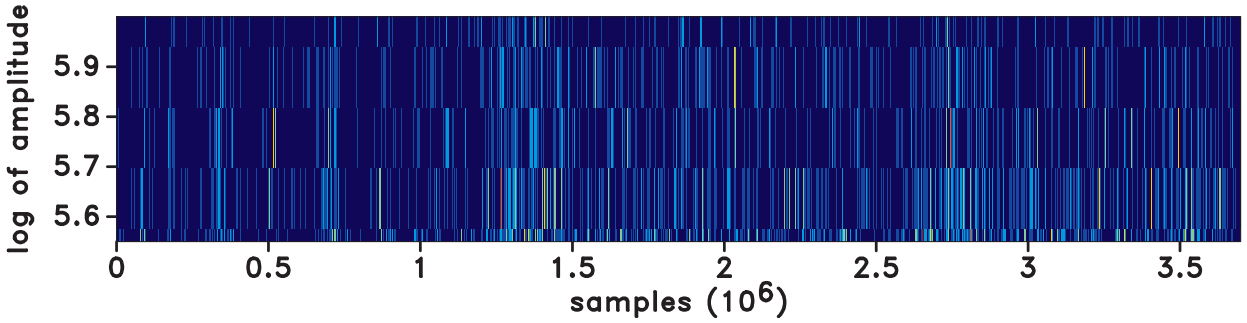
(a)



(b)



(c)



(d)

Figure 3.8: Binned weights using equation 3.5: (a) True model in Figure 2.8(a). (b) Estimated model in Figure 3.6(d). (c), (d) Detail of the top part of (a) and (b), respectively. Note the increase in number of high amplitude weights, which correspond to small values in the coefficients, in (d).

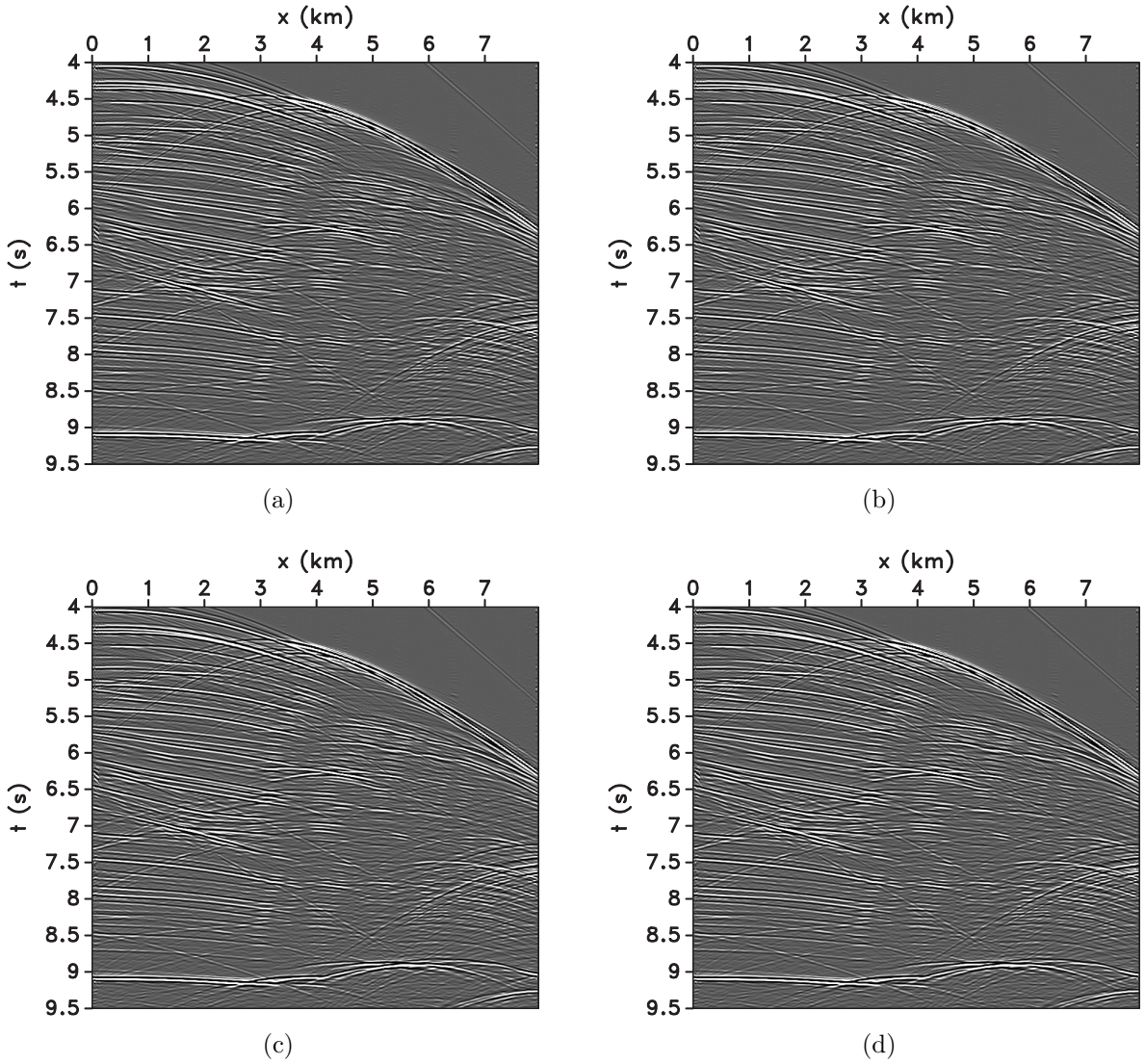
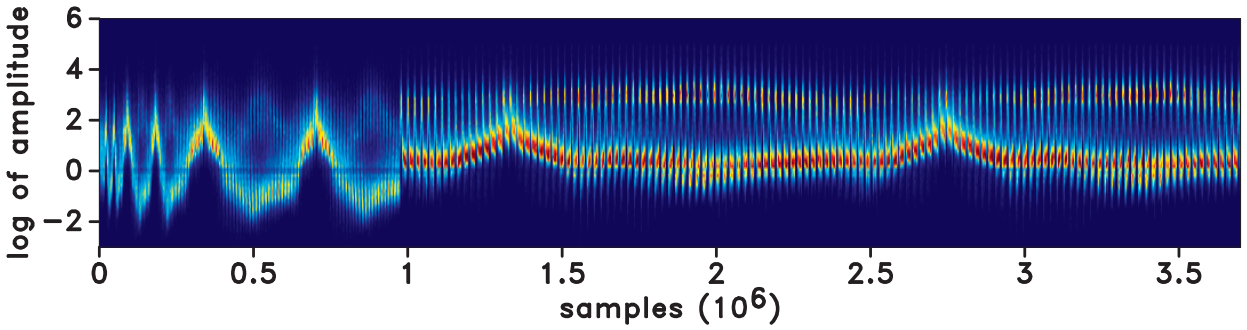
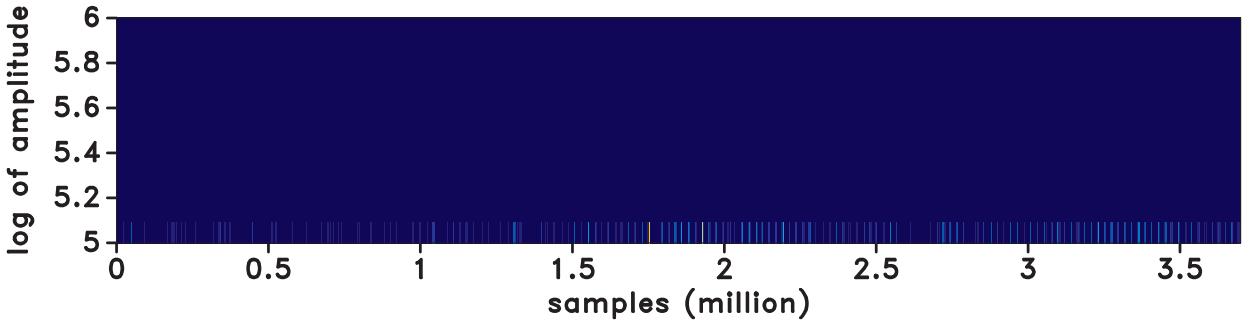


Figure 3.9: Results of Algorithm 1 for different number of iterations and $\epsilon = 1$: (a) 25 iterations, 19.76 dB. (b) 50 iterations, SNR fof 19.78 dB. (c) 75 iterations, SNR of 19.78 dB. (d) 100 iterations, SNR of 19.78 dB.



(a)



(b)

Figure 3.10: Binned weights using equation 3.5: (a) Figure 3.9(a). (b) Detail of the top part of (a). Note that the number of high amplitude weights, which correspond to the small values in the coefficients, decrease when compared to Figure 3.8(b).

3.4 An alternative solution

In the past experiments, I have established that Algorithm 1 can be used to enhance sparse solutions when compared to more common approaches, such as synthesis and analysis. In particular, the choice of weights in equation 3.5 enhances sparsity by assigning large weights to coordinates of the solution that are supposed to be small and small weights otherwise. However, the example using the Sigsbee model in Figure 3.9 shows that starting from a relatively sparse solution, such as the one provided by using the analysis approach as the unweighted iteration, directs the subsequent reweighted steps to models that are sparser, even though the true model I look for is not as sparse, as denoted in Figure 3.8(a). While the transform coefficients of the estimated solution become less sparse as the reweighted iterations progress (Figure 3.10(a)), one can imagine that using a model that is less sparse than the one originated from the analysis solution as an initial model can help better direct Algorithm 1 towards a comparable solution in less reweighted steps. In this section, I explore a way of creating such model using a modified gradient that looks for less sparse models when compared to the conventional one.

3.4.1 Modification of the L_1 gradient

The true force of the L_1 norm in inverse problems is its gradient. Recall that the gradient of such function for a certain model $\mathbf{x}_k$ at a given iteration of the algorithm k, is obtained using

$$\nabla f(\mathbf{x}_k(j)) = \begin{cases} sgn(\mathbf{x}_k(j)), & if \quad |\mathbf{x}_k(j)| > \delta \\ \frac{\mathbf{x}_k(j)}{\gamma}, & if \quad |\mathbf{x}_k(j)| \leq \delta \end{cases} \tag{3.7}$$

where δ is an arbitrarily small parameter that accounts for the non-differentiability of the L_1 norm. Note that equation 3.7 essentially denotes that the gradient, in this case, is mostly just the sign of the model coefficients. This means that this gradient treats such coefficients uniformily, i.e., they have the same influence inside the gradient regardless of their amplitude.

Note that, if I denote equation 3.7 by an operator $\mathbf{G}$, I can write

$$\nabla f(\mathbf{x}_k) = \mathbf{\Theta}^* \mathbf{G} \mathbf{\Theta} \mathbf{x}_k, \tag{3.8}$$

as the gradient of the analysis approach objective function. When incorporating the weight matrix in the inversion, the gradient becomes

$$\nabla f(\mathbf{x}_k) = \mathbf{\Theta}^* \mathbf{W}^* \mathbf{G} \mathbf{W} \mathbf{\Theta} \mathbf{x}_k, \tag{3.9}$$

which means that equation 3.9 is now a scaled version of equation 3.8, and points to a preferential direction acording to $\mathbf{W}$. In particular, this modified gradient seeks models that are small for coefficients where $\mathbf{W}$ has large values and vice-versa, in accordance with our previous experiments. One can alter the direction of the gradient by using the inverse of the weight matrix instead of the adjoint, yielding

$$\nabla f(\mathbf{x}_k) = \mathbf{\Theta}^* \mathbf{W}^{-1} \mathbf{G} \mathbf{W} \mathbf{\Theta} \mathbf{x}_k, \tag{3.10}$$

Note that Equations 3.9 and 3.10 can be interpreted as two different ways of obtaining a representation of the gradient, $\mathbf{G}\mathbf{W}\mathbf{\Theta}\mathbf{x}_k$, in the ambient domain. While the former produces the adjoint represention of the gradient, the latter obtains the simplest gradient according to the L_2 norm. In practice, Equation 3.10 returns a gradient that is the converse of Equation 3.9 in terms of weight placement, since $\mathbf{W}^{-1} = |\mathbf{\Theta}\mathbf{x}^i| + \epsilon$. This enforces the weighted inversion with this gradient to find models that are less sparse than those of equation 3.9, since small weights are used in different coefficients. To implement such change in Algorithm 1, I just need to supply I different function to calculate the gradient according to equation 3.10 instead of equation 3.9.

3.4.2 Implications and examples

The gradient in Equation 3.10 motivates finding models that are less sparse than those obtained using its counterpart in Equation 3.9, because of the assignment of the weights explained in section 3.4.1. One can argue that by using a less sparse model to create weights, the focus turns away from approximating the L_0 norm, since enhancing sparsity and better

resolving the largest coefficients in the transform domain are not the main objective. While this is true, the intention of such gradient is to create a less sparse model that fits the measurements, thus providing the reweighted steps with weights that are more suitable to generate good solutions for models that are not strictly sparse.

I demonstrate the modified gradient in a toy problem using Figure 2.1, with $\epsilon = 10^{-2}$ taken from the previous experiments. In Figure 3.11, I show the result of the reweighted steps for this model using the conventional gradient in equation 3.9. Note that after 5 reweighted iterations, the estimated model, in orange, is very close to the true model, in blue. In Figure 3.12, I show the results of the same experiment but with the gradient as in equation 3.10. In this case, the inversion converges to a reasonably estimated model in 3 reweighted iterations, as opposed to 5. Also, the intermediate models from both experiments are clearly different, which means that both problems took different paths in the objective function, as expected. In fact, Figure 3.12(b) closely resembles Figure 3.11(c), which indicates that the modified gradient equation 3.10 provides better weights for the reweighted algorithm.

I also use the modified gradient to interpolate the randomly undersampled Sigsbee model in Figure 2.12(a), as I did with Algorithm 1, also with $\epsilon = 10^{-2}$. Figure 3.13 show the solution using the modified gradient after 2 reweighted steps. A comparison of Figure 3.6(c) with this experiment show that the large gap area is better represented. Figure 3.14 shows the weights for 2 reweighted steps using the modified gradient. Again, the weights are binned on a 100×1000 grid, representing the amplitude of the weights and the number of samples, respectively. Notice that, in the second reweighted step, the weights are much closer together around lower values when compared to the weights in Figure 3.7(b), even though they maintain a structure that resembles the weights obtained with the analysis solution.

I use the solution present in Figure 3.13 as a starting model for the Algorithm 1 using the conventional gradient, i.e., equation 3.9. Figure 3.15 shows the results for several number of iterations. Note that the SNR of the solutions using this method are superior to the ones

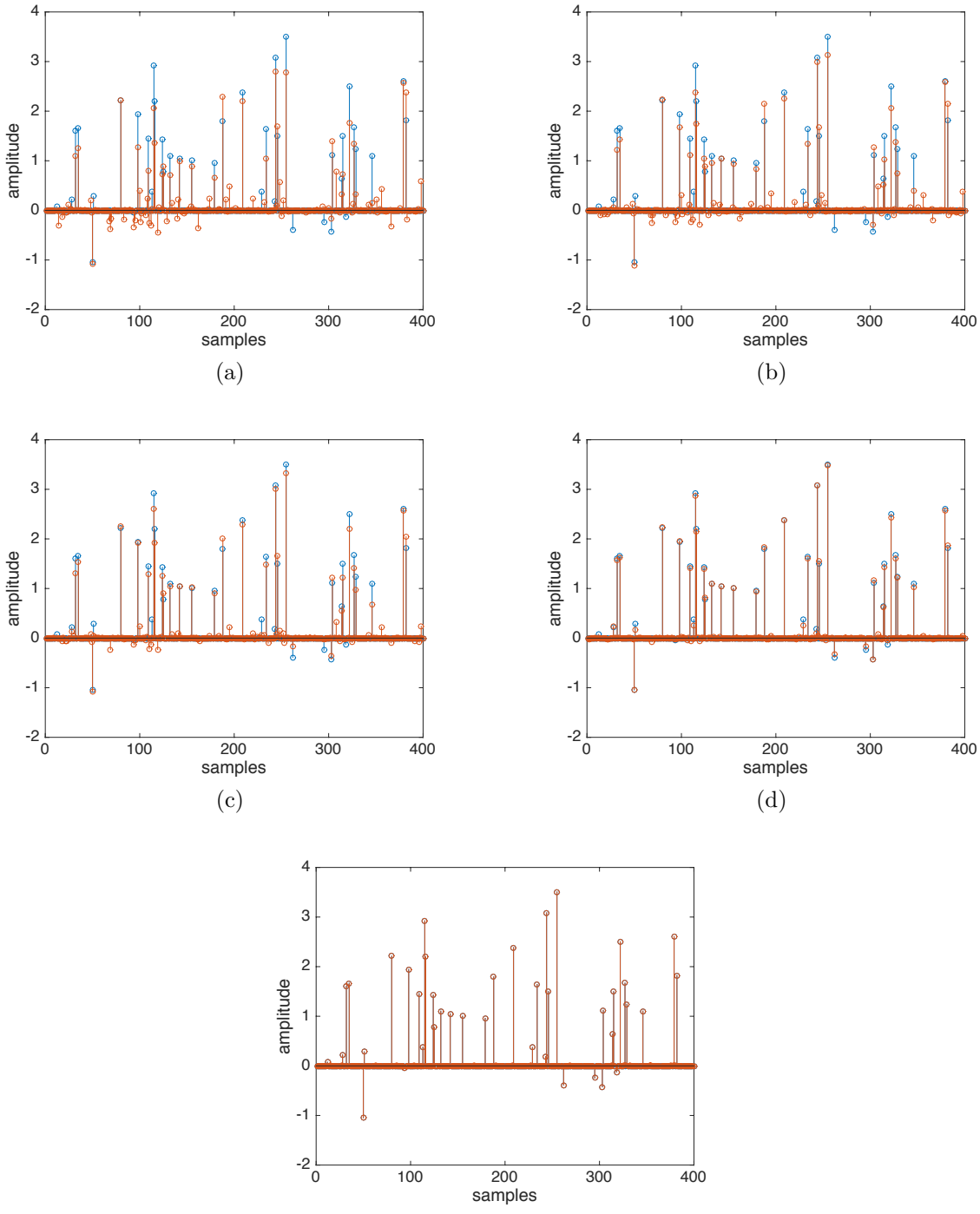


Figure 3.11: Reweighted iterations for the toy problem using equation 3.9: (a) First iteration, (b) second iteration, (c) third iteration, (d) fourth iteration, (e) fifth iteration. The true model and the estimated solution are displayed in blue and orange, respectively.

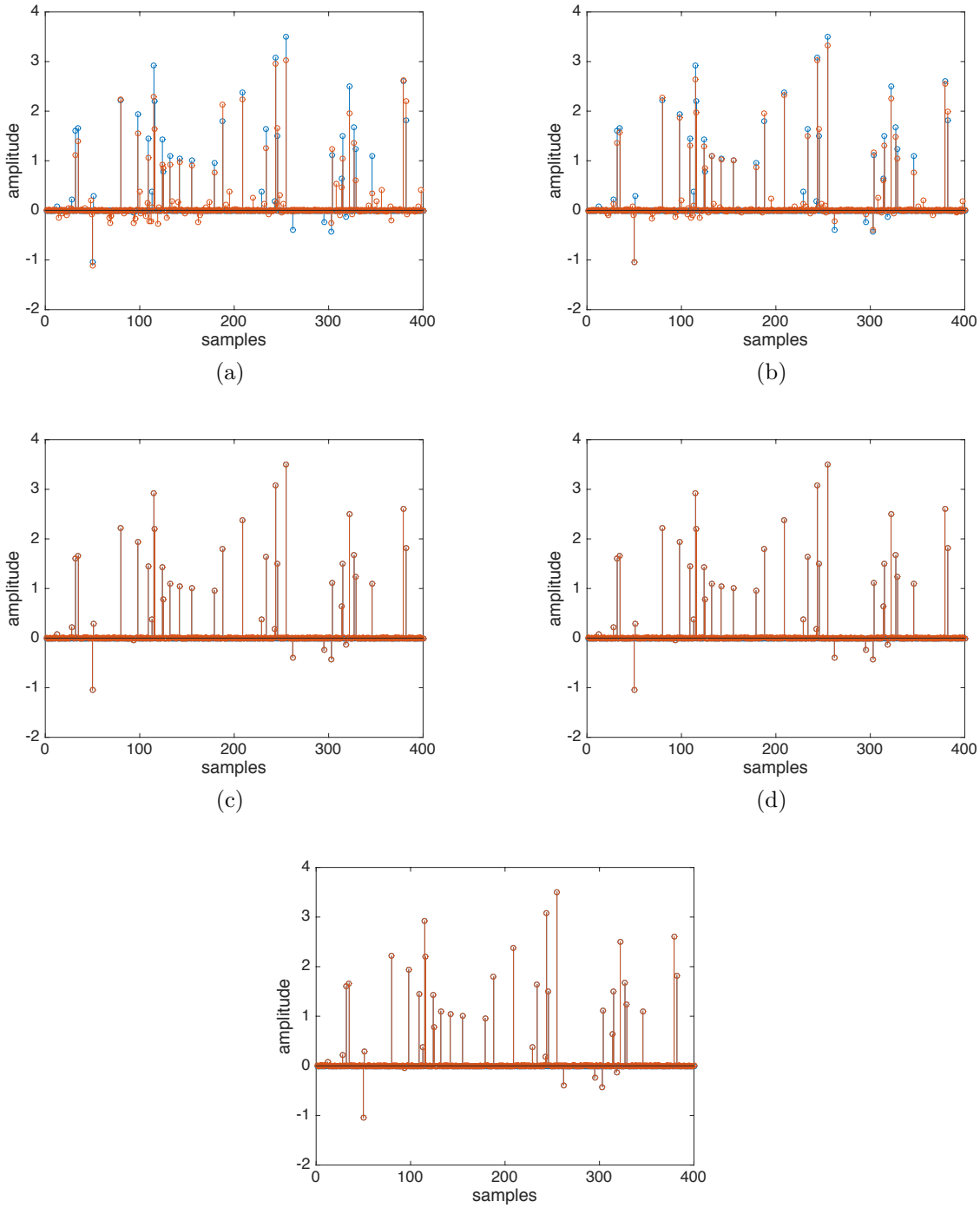
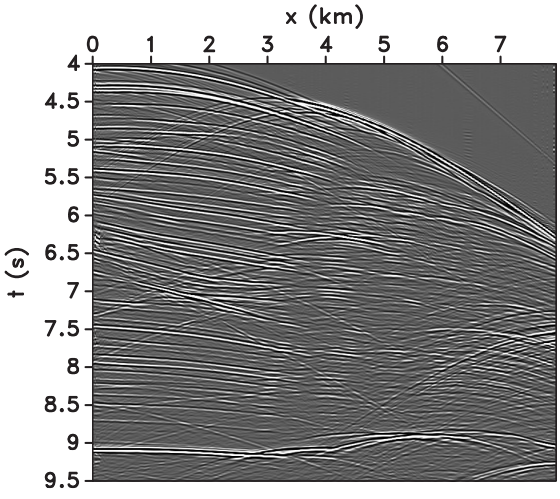
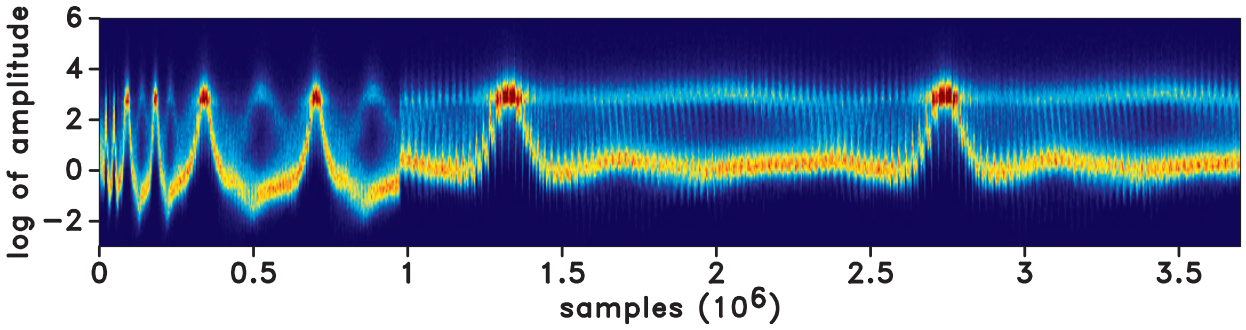


Figure 3.12: Reweighted iterations for the toy problem using equation 3.10: (a) First iteration, (b) second iteration, (c) third iteration, (d) fourth iteration, (e) fifth iteration. The true model and the estimated solution are displayed in blue and orange, respectively.

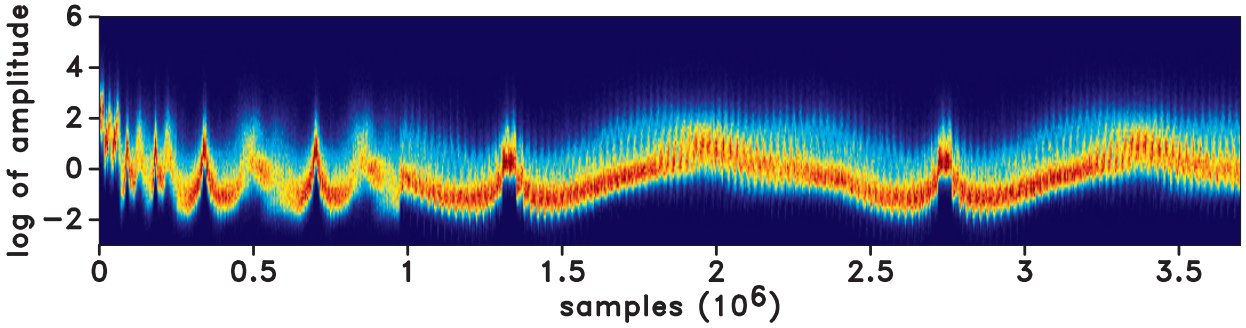


(a)

Figure 3.13: (a) Solution using the modified gradient in equation 3.10, SNR of 18.70 dB.



(a)



(b)

Figure 3.14: Binned weight vectors for minimization procedure that estimates Figure 3.13: (a) first and (b) second reweighted iterations. Notice how the weights are close to each other in the second reweighted iteration. The colors indicate the density of points inside a given bin.

found in Figure 3.9, which indicates that the solutions are numerically closer to the true model in Figure 2.8(a) for a given number of iterations. Figure 3.16 shows the convergence curves for both experiments. Note that the blue curve, which represents the experiments in Figure 3.9, obtains lower model differences faster than the red curve in the earlier iterations. In addition, while the red curve plateaus fairly quickly, the blue curve obtains a lower error in the end of the inversion, which points to a better solution. This is due to the different initial model in the experiments, which enables the inversion to converge to different local minima.

3.5 3D Examples

In this section, I consider a 3D cube composed of 100 shots from the Gulf of Mexico data, shown in Figure 3.17. Because of the dimension of the data, I will focus on a subset of the data in our next examples, which I refer to as details. I uniformly undersample the Gulf of Mexico data by 50% and obtain the data with missing traces shown in the detail Figure 3.18(a). Figure 3.18(b), Figure 3.18(c) and Figure 3.18(d) show the estimates for the proposed approach and also for the synthesis and analysis solutions, respectively. Notice that none of the approaches manage to obtain a very good SNR. This is because of the uniform undersampling scheme, which does not favor reconstruction along the crossline direction, which means that although the inline sections are well reconstructed, the crossline sections deviate more from the true data. However, notice that the proposed approach obtains a higher SNR than those of the synthesis and analysis approaches. Also, the artifacts shown in the analysis and synthesis solutions are greatly diminished in the solution of the proposed approach.

I also randomly undersample the Gulf of Mexico data by 50%, obtaining the data with missing traces shown in the detail Figure 3.19(a). Figure 3.19(b), Figure 3.19(c) and Figure 3.19(d) show the estimates for the analysis, synthesis and proposed approaches. Notice that both estimates are very similar and feature no coherent aliases. This occurs because this type of undersampling scheme heavily favors reconstruction, since crossline and inline

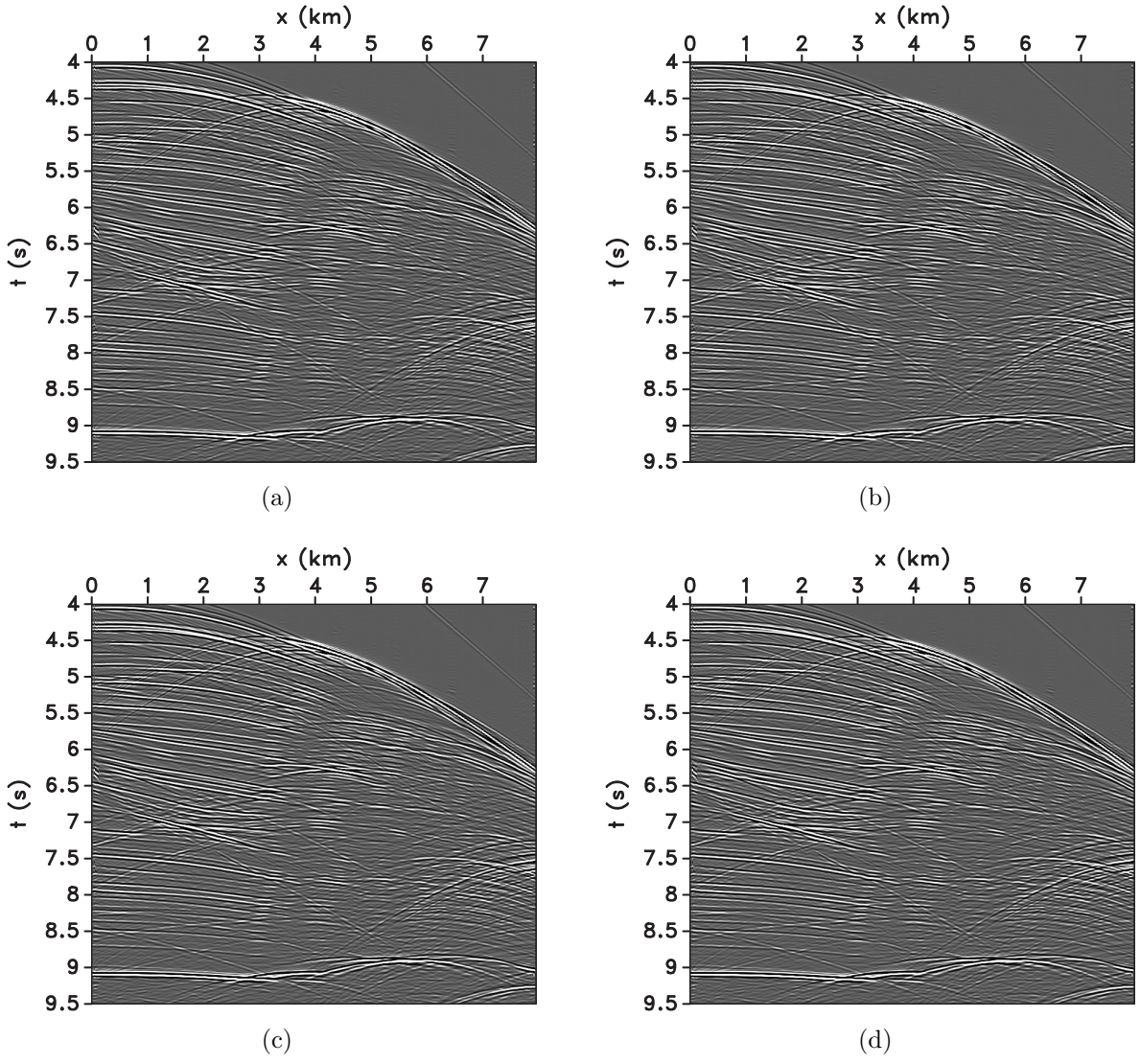
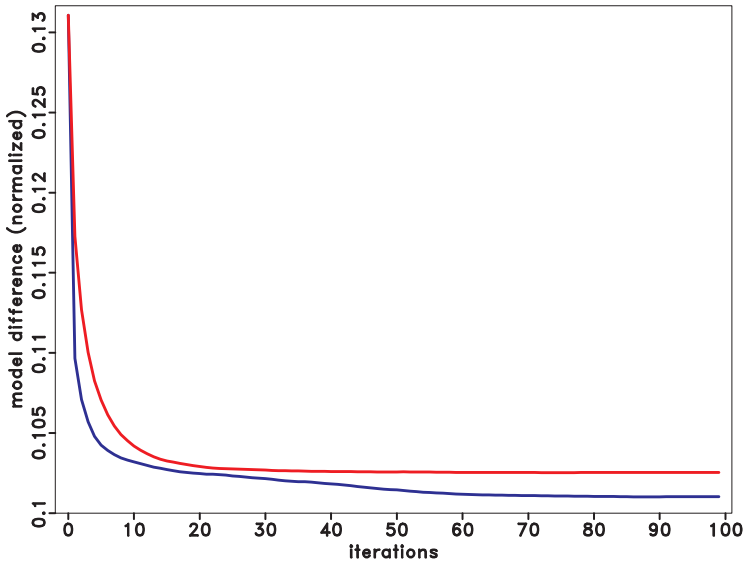


Figure 3.15: Results of Algorithm 1 using Figure 3.13 as starting model and $\epsilon = 1$: (a) 25 iterations, 19.80 dB. (b) 50 iterations, SNR fof 19.85 dB. (c) 75 iterations, SNR of 19.97 dB. (d) 100 iterations, SNR of 19.98 dB.



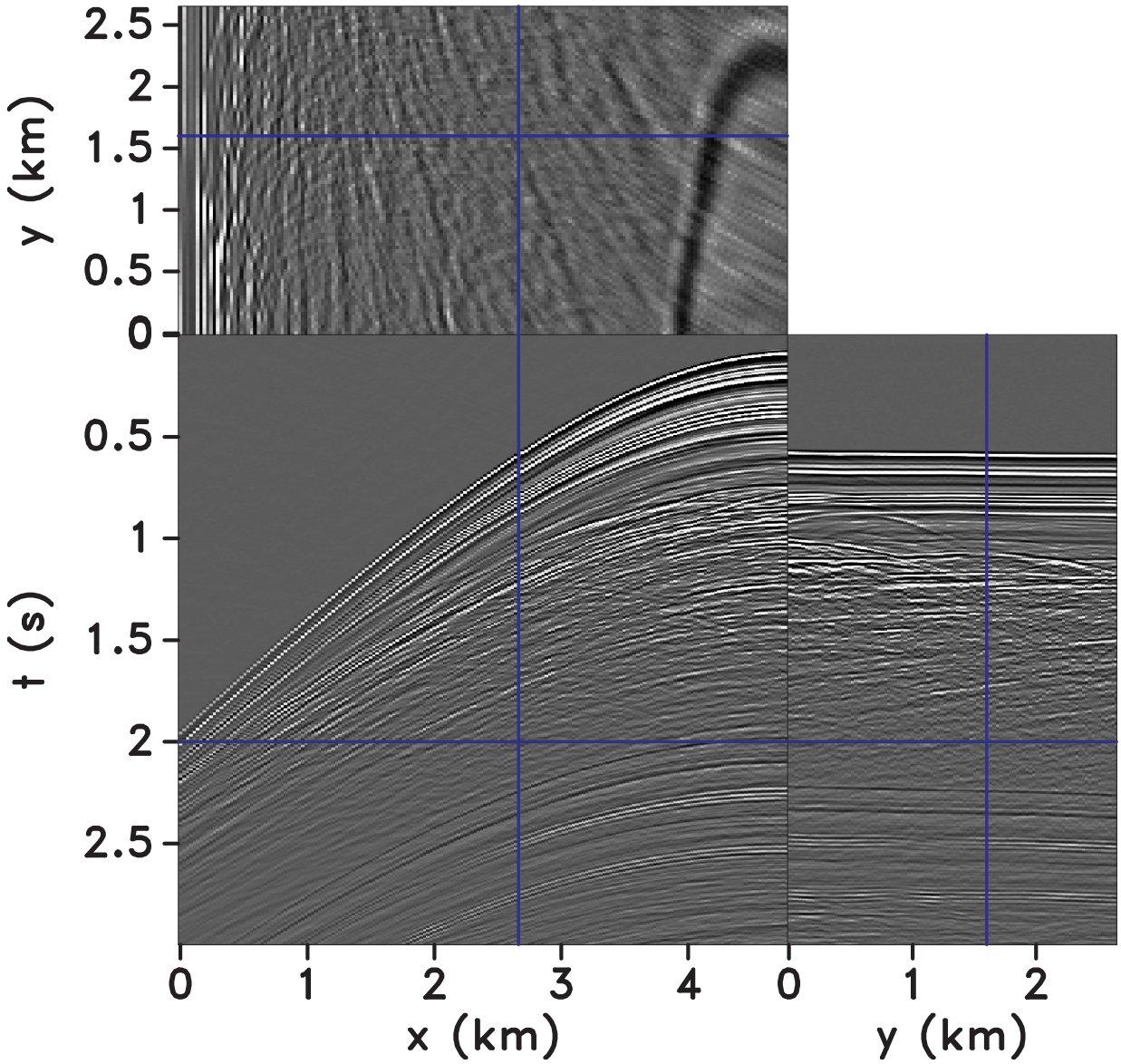
(a)

Figure 3.16: (a) Convergence curves for the experiments in Figure 3.9, in red, and Figure 3.15, in blue. The normalized difference is calculated as $\|\mathbf{x} - \tilde{\mathbf{x}}\|_2 / \|\mathbf{x}\|_2$, where $\mathbf{x}$ and $\tilde{\mathbf{x}}$ are the true model and the estimated solution, respectively.

information can be used. Also, because the data are only undersampled by 50%, there are no regions featuring large gaps both in the inline and crossline directions at the same time inside the 3D cube, which also favors reconstruction. However, the solution of the proposed approach shows higher SNR than those of the synthesis and analysis approaches, thus indicating that it better matches the original data.

3.6 Discussion

In this chapter, I evaluate the performance of a reweighted L_1 norm algorithm as an approximation for the L_0 norm objective function in the context of seismic interpolation. Because the L_0 norm does not depend on the magnitude of the coefficients of the solution in the transform domain, ideally such objective function would enhance sparsity and help obtain more accurate results for the problem. However, due to the non-convexity of the L_0 norm and the multiple local minima intrinsic to the seismic interpolation problem due to the restriction matrix, obtaining a good solution in this setting is heavily dependent of



(a)

Figure 3.17: 100 shots from the 2D Gulf of Mexico dataset.

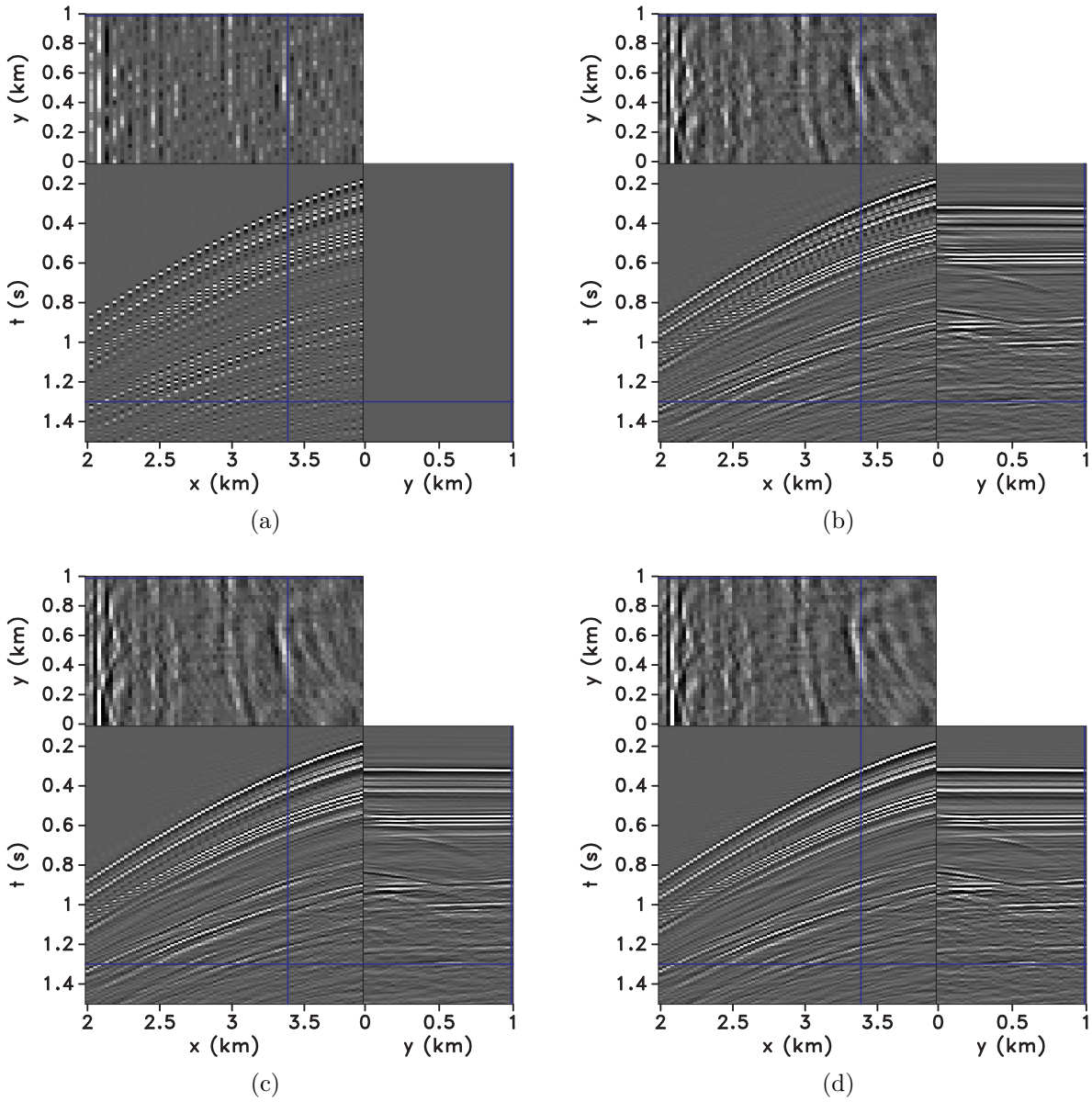


Figure 3.18: (a) Detail illustrating uniformly undersampled data. (b), (c) Synthesis and analysis solutions, SNR of 6.23 dB and 6.27 dB, respectively. (d) Reweighted solution with the inverse, SNR of 7.15 dB.

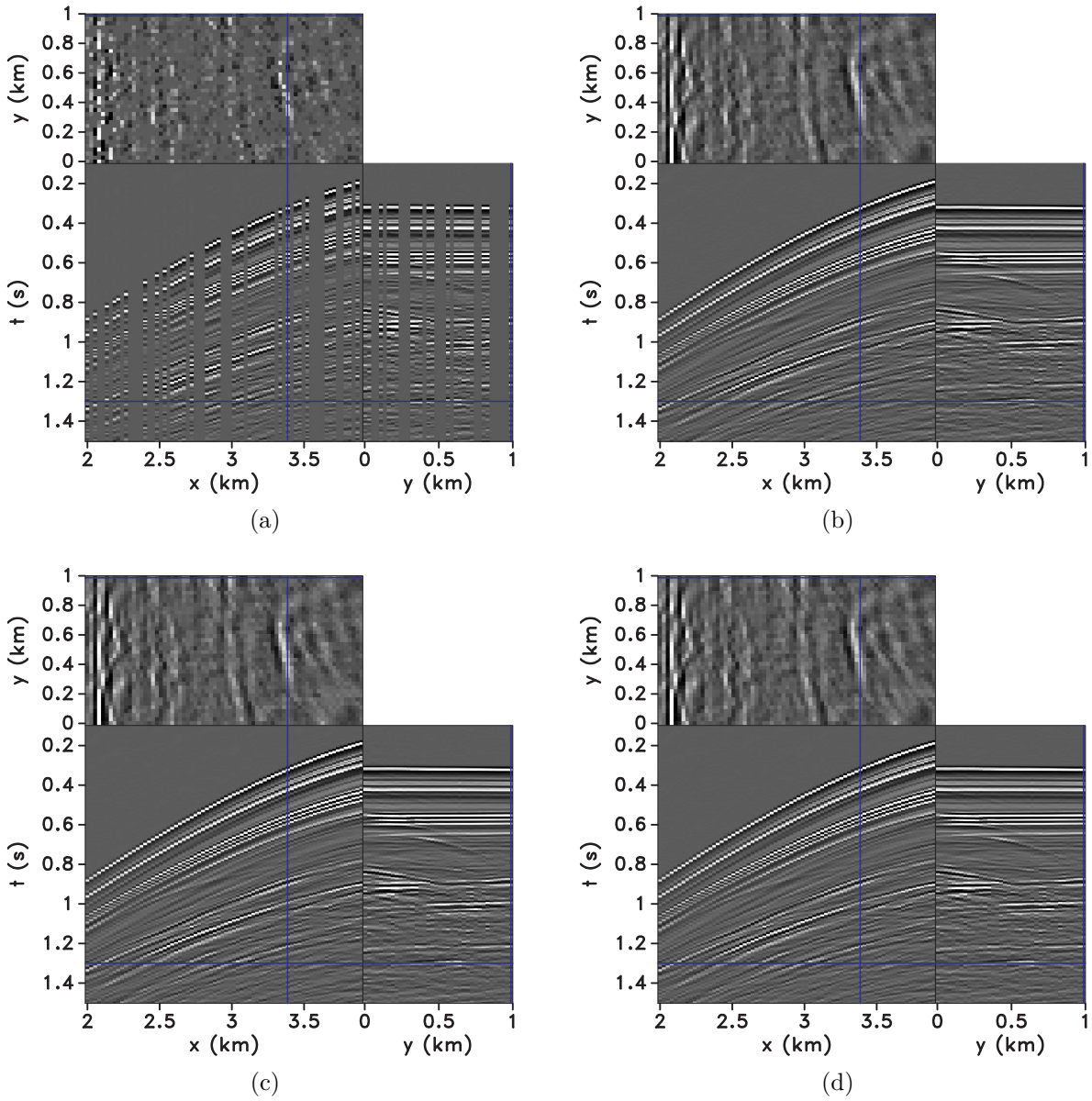


Figure 3.19: (a) Detail illustrating randomly undersampled data. (b), (c) Synthesis and analysis solutions, SNR of 17.27 dB and 17.21 dB, respectively. (d) Reweighted solution with the inverse, SNR of 18.94 dB.

the initial weights of the reweighted algorithm, which is commonly taken as the analysis approach solution. The results show that this choice of initial weights takes a large number of reweighted iterations to produce a result that better explains the big gaps when compared to the conventional analysis solution.

To solve this problem, I propose a modified gradient approach to obtain an initial model with different weights then those of the analysis approach, effectively exploring a different local minima of the L_0 objective function. Although this approach is not theoretically proven, i.e., there is no guarantee that it will yield a solution that converges to a better local minima for a given problem through Algorithm 1, the empirical results show that this approach returns better models than the conventional (synthesis and analysis) approaches for the case of seismic interpolation in a reasonable number of reweighted iterations. In particular, it helps mitigate artifacts due to aliasing when the uniform undersampling scheme is used and produces solutions with higher SNR in the case of random undersampling. Further research needs to be conducted to evaluate the theoretical performance of such approach and whether it would be directly applicable to other types of problems.

CHAPTER 4

CONCLUSIONS

In this thesis, I evaluate sparsity constrained methods for inversion in the context of seismic interpolation. This is a challenging problem because of its kernel, the restriction matrix, which provides no information about the missing samples. Therefore, this problem is plagued with local minima and requires regularization in order to estimate feasible models.

I consider two approaches for sparsity constrained inversion: the analysis and synthesis approaches. These formulations involve the minimization of an objective function containing the L_1 norm, which is known to promote sparsity. However, while the synthesis approach concentrates on finding the sparsest solution that fits the data, the analysis approach uses model shaping to determine the best solution. The results show that although it is possible to obtain reasonable estimates of the missing traces using either approach, the analysis solution attains fewer artifacts than that of the synthesis approach due to the model shaping nature of its objective function. In general, the quality of the solution for both approaches depend on the undersampling scheme (related to the kernel) and the size of gaps between adjacent traces, since large gaps produce local minima in the inversion scheme. This occurs because, in general, the larger the gap size, the more it contributes in magnitude to the solution. Therefore, such approaches should be used when the undersampling scheme is favorable (random or close to it) and when there are no large gaps between adjacent traces.

Because the L_1 norm scheme is dependent on the magnitude of the coefficients of the solution and thus is just a proxy for sparsity, I evaluate a reweighted scheme for sparse constrained inversion that approximates the L_0 norm, which is independent of magnitudes. Although the L_0 norm objective function better enforces sparsity when compared to the L_1 norm, it is also non-convex and thus, possesses many local minima. Therefore, it is dependent on the choice of weights and initial model. The examples and results shown in

this thesis demonstrate that although using the analysis approach to create the initial model for the reweighted scheme may work well for some problems, this works slowly for the missing trace reconstruction problems treated in this thesis. This is due to the fact that the models we commonly seek in geophysical applications are not strictly sparse. Therefore, using the analysis solution as an initial model for the reweighted scheme may not be optimal in such situations, since it is obtained using a conventional L_1 norm inversion, and thus is commonly sparser than the model we seek.

To solve this problem, I propose to create a less sparse initial model using a modification of the objective function's gradient by using the inverse of the weight matrix instead of its adjoint, in order to obtain weights that differ from the ones tied to the analysis approach solution. Such gradient can be interpreted as to lead to a less sparse solution when compared to the conventional gradient, by allowing more coefficients to participate in the inversion. The results show that this approach can lead to faster convergence and the same benefits of the conventional reweighted approach, such as better resolved large gaps and higher SNR when compared to the synthesis and analysis approaches.

REFERENCES CITED

Aharon, M., M. Elad, and A. Bruckstein, 2006, *rmk*-svd: An algorithm for designing overcomplete dictionaries for sparse representation: IEEE Transactions on signal processing, **54**, 4311–4322.

Becker, S., J. Bobin, and E. J. Candès, 2011, Nesta: a fast and accurate first-order method for sparse recovery: SIAM Journal on Imaging Sciences, **4**, 1–39.

Boyd, S., and L. Vandenberghe, 2004, Convex optimization: Cambridge university press.

Candes, E., L. Demanet, D. Donoho, and L. Ying, 2006, Fast discrete curvelet transforms: Multiscale Modeling & Simulation, **5**, 861–899.

Candes, E., and J. Romberg, 2007, Sparsity and incoherence in compressive sampling: Inverse problems, **23**, 969.

Candes, E. J., 2008, The restricted isometry property and its implications for compressed sensing: Comptes Rendus Mathematique, **346**, 589–592.

Candès, E. J., and D. L. Donoho, 2004, New tight frames of curvelets and optimal representations of objects with piecewise c2 singularities: Communications on pure and applied mathematics, **57**, 219–266.

Candes, E. J., Y. C. Eldar, D. Needell, and P. Randall, 2011, Compressed sensing with coherent and redundant dictionaries: Applied and Computational Harmonic Analysis, **31**, 59–73.

Candes, E. J., and Y. Plan, 2011, A probabilistic and ripless theory of compressed sensing: IEEE Transactions on Information Theory, **57**, 7235–7254.

Candès, E. J., J. Romberg, and T. Tao, 2006, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information: Information Theory, IEEE Transactions on, **52**, 489–509.

Candès, E. J., and M. B. Wakin, 2008, An introduction to compressive sampling: IEEE signal processing magazine, **25**, 21–30.

Candes, E. J., M. B. Wakin, and S. P. Boyd, 2008, Enhancing sparsity by reweighted 1 minimization: Journal of Fourier analysis and applications, **14**, 877–905.

Chen, Y., J. Ma, and S. Fomel, 2016, Double sparsity dictionary for seismic noise attenuation: Geophysics, **81**, V103–V116.

Claerbout, J., 2014, Geophysical image estimation by example: Stanford University Press.

Clapp, R. G., et al., 2006, Amo inversion to a common azimuth dataset: 2006 SEG Annual Meeting, Society of Exploration Geophysicists, 2097–2101.

Daubechies, I., et al., 1992, Ten lectures on wavelets: SIAM, **61**.

De Almeida, L. A., M. Wakin, and P. Sava, 2016, Data denoising and interpolation using synthesis and analysis sparse regularization: SEG Technical Program Expanded Abstracts 2016, Society of Exploration Geophysicists, 4112–4117.

Do, T. T., T. D. Tran, and L. Gan, 2008, Fast compressive sampling with structurally random matrices: Acoustics, Speech and Signal Processing, 2008. ICASSP 2008. IEEE International Conference on, IEEE, 3369–3372.

Donoho, D. L., 2006, Compressed sensing: Information Theory, IEEE Transactions on, **52**, 1289–1306.

Duarte, M. F., and Y. C. Eldar, 2011, Structured compressed sensing: From theory to applications: IEEE Transactions on Signal Processing, **59**, 4053–4085.

Elad, M., P. Milanfar, and R. Rubinstein, 2007, Analysis versus synthesis in signal priors: Inverse problems, **23**, 947.

Fomel, S., 2002, Applications of plane-wave destruction filters: Geophysics, **67**, 1946–1960.

———, 2007, Shaping regularization in geophysical-estimation problems: Geophysics, **72**, R29–R36.

Fomel, S., and Y. Liu, 2010, Seislet transform and seislet frame: Geophysics, **75**, V25–V38.

Hennenfent, G., and F. J. Herrmann, 2008, Simply denoise: wavefield reconstruction via jittered undersampling: Geophysics, **73**, V19–V28.

Li, X., A. Y. Aravkin, T. van Leeuwen, and F. J. Herrmann, 2012, Fast randomized full-waveform inversion with compressive sensing: Geophysics, **77**, A13–A17.

Lin, T. T., and F. J. Herrmann, 2013, Cosparse seismic data interpolation: Presented at the 75th EAGE Conference & Exhibition incorporating SPE EUROPEC 2013.

Liu, B., and M. D. Sacchi, 2004, Minimum weighted norm interpolation of seismic records: Geophysics, **69**, 1560–1568.

Mallat, S., 2008, A wavelet tour of signal processing: the sparse way: Academic press.

Mansour, H., F. J. Herrmann, and Ö. Yılmaz, 2013, Improved wavefield reconstruction from randomized sampling via weighted one-norm minimization: Geophysics, **78**, V193–V206.

Naghizadeh, M., and M. D. Sacchi, 2010, Beyond alias hierarchical scale curvelet interpolation of regularly and irregularly sampled seismic data: Geophysics, **75**, WB189–WB202.

Nam, S., M. E. Davies, M. Elad, and R. Gribonval, 2013, The cosparse analysis model and algorithms: Applied and Computational Harmonic Analysis, **34**, 30–56.

Nesterov, Y., 2005, Smooth minimization of non-smooth functions: Mathematical programming, **103**, 127–152.

Porsani, M. J., 1999, Seismic trace interpolation using half-step prediction filters: Geophysics, **64**, 1461–1467.

Ramirez, C., G. Larrazabal, and G. Gonzalez, 2016, Salt body detection from seismic data via sparse representation: Geophysical Prospecting, **64**, 335–347.

Rodriguez, I. V., and M. D. Sacchi, 2014, Microseismic source imaging in a compressed domain: Geophysical Journal International, **198**, 1186–1198.

Ronen, J., 1987, Wave-equation trace interpolation: Geophysics, **52**, 973–984.

Selesnick, I. W., and M. A. Figueiredo, 2009, Signal restoration with overcomplete wavelet transforms: comparison of analysis and synthesis priors: SPIE Optical Engineering+ Applications, International Society for Optics and Photonics, 74460D–74460D.

Spitz, S., 1991, Seismic trace interpolation in the fx domain: Geophysics, **56**, 785–794.

Stolt, R. H., 2002, Seismic data mapping and reconstruction: Geophysics, **67**, 890–908.

Trad, D., 2003, Interpolation and multiple attenuation with migration operators: Geophysics, **68**, 2043–2054.

———, 2009, Five-dimensional interpolation: Recovering from acquisition constraints: Geophysics, **74**, V123–V132.

Wason, H., F. J. Herrmann, T. T. Lin, et al., 2011, Sparsity-promoting recovery from simultaneous data: A compressive sensing approach: Presented at the 2011 SEG Annual Meeting, Society of Exploration Geophysicists.

Xu, S., Y. Zhang, D. Pham, and G. Lambaré, 2005, Antileakage fourier transform for seismic data regularization: Geophysics, **70**, V87–V95.

Yang, P., and S. Fomel, 2015, Seislet-based morphological component analysis using scale-dependent exponential shrinkage: Journal of Applied Geophysics, **118**, 66–74.

Yu, S., J. Ma, X. Zhang, and M. D. Sacchi, 2015, Interpolation and denoising of high-dimensional seismic data by learning a tight frame: Geophysics, **80**, V119–V132.

Zhu, L., E. Liu, and J. H. McClellan, 2015, Seismic data denoising through multiscale and sparsity-promoting dictionary learning: Geophysics, **80**, WD45–WD57.

Zhu, Z., and M. B. Wakin, 2015, Wall clutter mitigation and target detection using discrete prolate spheroidal sequences: Compressed Sensing Theory and its Applications to Radar, Sonar and Remote Sensing (CoSeRa), 2015 3rd International Workshop on, IEEE, 41–45.

CHAPTER 5

APPENDIX - NESTA

In this thesis, I have used my own implementation of the NESTA solver (Becker et al., 2011) to perform the analysis and synthesis minimization problems present in equation 2.10 and equation 2.11. For completeness, I include here the main steps of this solver. For benchmark comparisons with other state-of-the-art solvers and theoretical background, I refer the reader to Becker et al. (2011) and Nesterov (2005), respectively.

The NESTA solver proposes to minimize the following function

$$f(\mathbf{x}) = \|\mathbf{x}\|_1 = \max_{\mathbf{u}\in Q_d} \langle \mathbf{u}, \mathbf{x} \rangle, \tag{.1}$$

on the feasible set given by

$$Q_d = \{\mathbf{u} : \|\mathbf{u}\|_\infty \leq 1\}. \tag{.2}$$

A smooth approximation to equation .1 is given by

$$f(\mathbf{x})_\mu = \|\mathbf{x}\|_1 = \max_{\mathbf{u}\in Q_d} \langle \mathbf{u}, \mathbf{x} \rangle - \mu p_d(\mathbf{u}), \tag{.3}$$

where p_d is a prox-function. Note that, as $\mu \longrightarrow 0$, the better the minimization of the L_1 norm of the function. The function present in equation .3 can be minimized using the following iterative procedure (Nesterov, 2005):

for each iteration i **do**
  $\alpha_i = 1/2(i+1)$, $\tau_i = 2/(i+3)$
  Calculate $\nabla f(\mathbf{x}_i)$
  $\mathbf{y}_i = \min_{x\in Q_p} (L/2)\|\mathbf{x} - \mathbf{x}_i\| + \langle \nabla f(\mathbf{x}_i), \mathbf{x} - \mathbf{x}_i \rangle$
  $\mathbf{z}_i = \min_{x\in Q_p} (L/\sigma_p) p_d(\mathbf{x}) + \sum_{j=0}^{i} \alpha_j \langle \nabla f(\mathbf{x}_j), \mathbf{x} - \mathbf{x}_j \rangle$
  $\mathbf{x}_{i+1} = \tau_i \mathbf{z}_i + (1 - \tau_i)\mathbf{y}_i$
end for
return $\mathbf{x}_i$

Algorithm 2: General NESTA Solver

In the above algorithm, σ_p is the convexity parameter of p_p and L is the Lipschitz constant of the gradient. The NESTA solver considers $p_d(\mathbf{u}) = 1/2\|\mathbf{u}\|_2^2$, which makes equation .3 the Huber function. This function has convexity parameter equal to one and a gradient that is Lipschitz with constant $1/\mu$. The gradient for this function can be calculated as

$$\nabla f_\mu(\mathbf{x})[k] = \begin{cases} x[k]/\mu & \text{if } |x[k]| < \mu \\ sgn(x[k]) & \text{otherwise,} \end{cases} \tag{.4}$$

where $x[k]$ is an element of the vector $\mathbf{x}$. We want to minimize equation .3 in the feasible set $Q_d = \{\mathbf{x} : \|\mathbf{b} - \mathbf{A}\mathbf{x}\|_2 \leq \epsilon\}$, which is an $L2$ ball enclosing the portion of the ambient domain where the data misfit is up to ϵ. In this appendix, $\mathbf{b}$ refers to the measured data. Assuming that $\mathbf{A}$ is an orthogonal projector, i.e., $\mathbf{A}^*\mathbf{A} = \mathbf{I}$ or $\mathbf{A}\mathbf{A}^* = \mathbf{I}$, the iterative process described above can be efficiently carried out using Algorithm 2.

Set $\mathbf{x}_0$, $\mathbf{b}$, μ, $L = 1/\mu$, ϵ
for each iteration i **do**
  $\alpha_i = 1/2(i+1)$, $\tau_i = 2/(i+3)$
  Calculate $\nabla f(\mathbf{x}_i)$
  $\mathbf{q}_{y,i} = \mathbf{x}_i - (1/L)\nabla f(\mathbf{x}_i)$
  $\lambda_{y,i} = \max(0, \epsilon^{-1}\|\mathbf{b} - \mathbf{A}\mathbf{q}_{y,i}\|_2 - L)$
  $\mathbf{y}_i = (I - \frac{\lambda_{y,i}}{(\lambda_{y,i}+L)}\mathbf{A}^*\mathbf{A})((\lambda_{y,i}/L)\mathbf{A}^*\mathbf{b} + \mathbf{q}_{y,i})$
  $\mathbf{q}_{z,i} = \mathbf{x}_0 - (1/L)\sum_{j=0}^{i}\alpha_j\nabla f(\mathbf{x}_j)$
  $\lambda_{z,i} = \max(0, \epsilon^{-1}\|\mathbf{b} - \mathbf{A}\mathbf{q}_{z,i}\|_2 - L)$
  $\mathbf{z}_i = (I - \frac{\lambda_{z,i}}{(\lambda_{z,i}+L)}\mathbf{A}^*\mathbf{A})((\lambda_{z,i}/L)\mathbf{A}^*\mathbf{b} + \mathbf{q}_{z,i})$
  $\mathbf{x}_{i+1} = \tau_i\mathbf{z}_i + (1-\tau_i)\mathbf{y}_i$
end for
return $\mathbf{x}_i$

Algorithm 3: NESTA Solver for Orthogonal Projections

Although the above algorithm works only for orthogonal projections, it is importante to notice that this assumption is widely present in compressive sensing approaches. Notably, Algorithm 2 does not require any matrix inversion, which allows for a fast solver. The aforementioned algorithm solves both the analysis and synthesis minimization problems by making the following change of variables: suppose $\mathbf{A} = \mathbf{R}\mathbf{U}$, where $\mathbf{R}$ is a restriction matrix and $\mathbf{U}$ is a frame. Substituting $\mathbf{x} \leftarrow \mathbf{U}\mathbf{x}$, one can apply the algorithm to solve the analysis

problem (Becker et al., 2011).

Regarding parameters, I use the ones recommended in (Becker et al., 2011). I set $\mu = 0.02$ for every experiment. For the noise level ϵ, I set $\epsilon = (\sqrt{m + 2\sqrt{2m}})\sigma$, where m is the number of elements in $\mathbf{b}$ and σ is the standard deviation of the noise.